# Why Jacket Matrices?

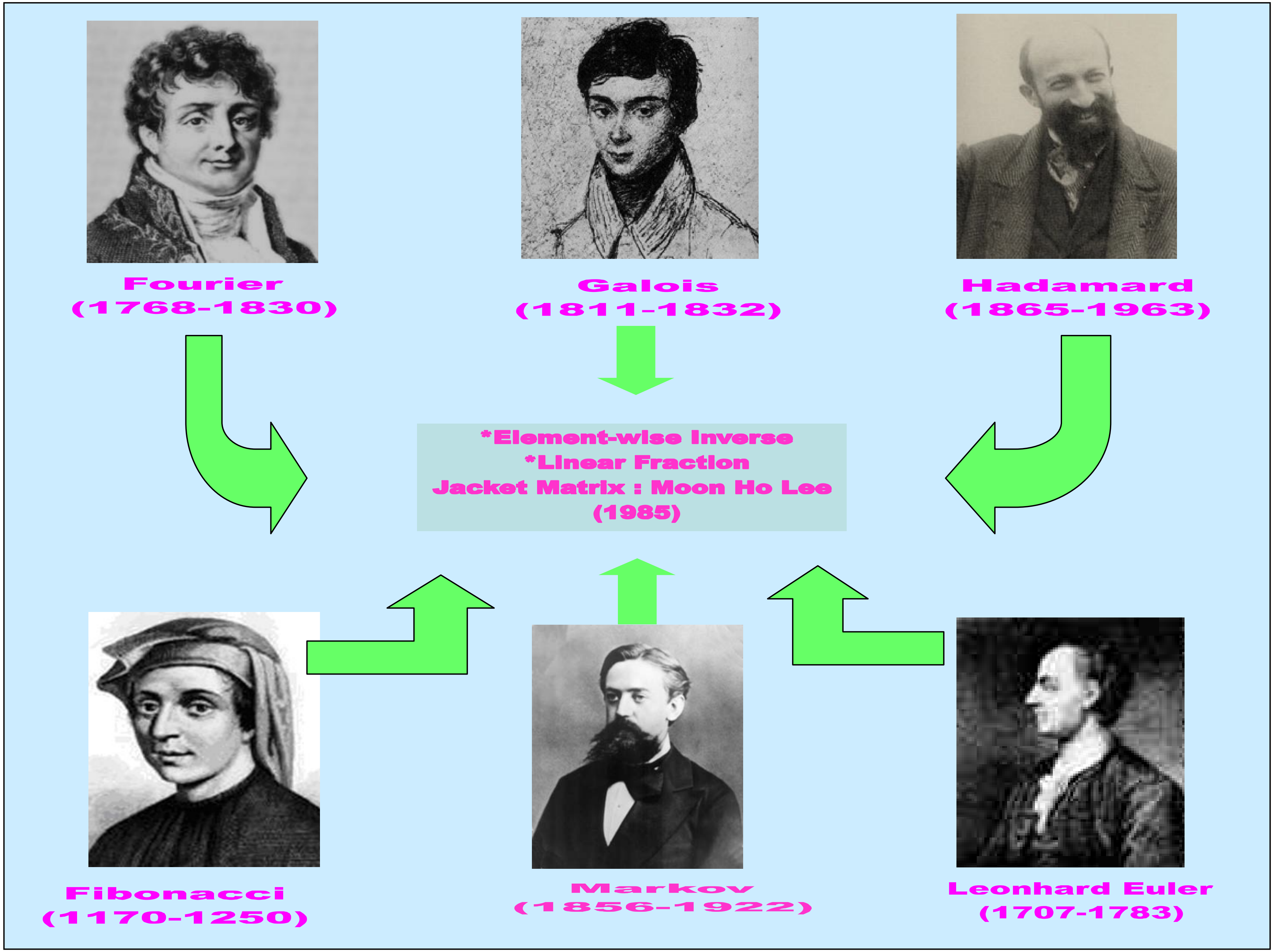
$$[A]_N^{-1} = [a_{ij}^{-1}]^T$$

**Moon Ho Lee (E-mail:moonho@chonbuk.ac.kr)**

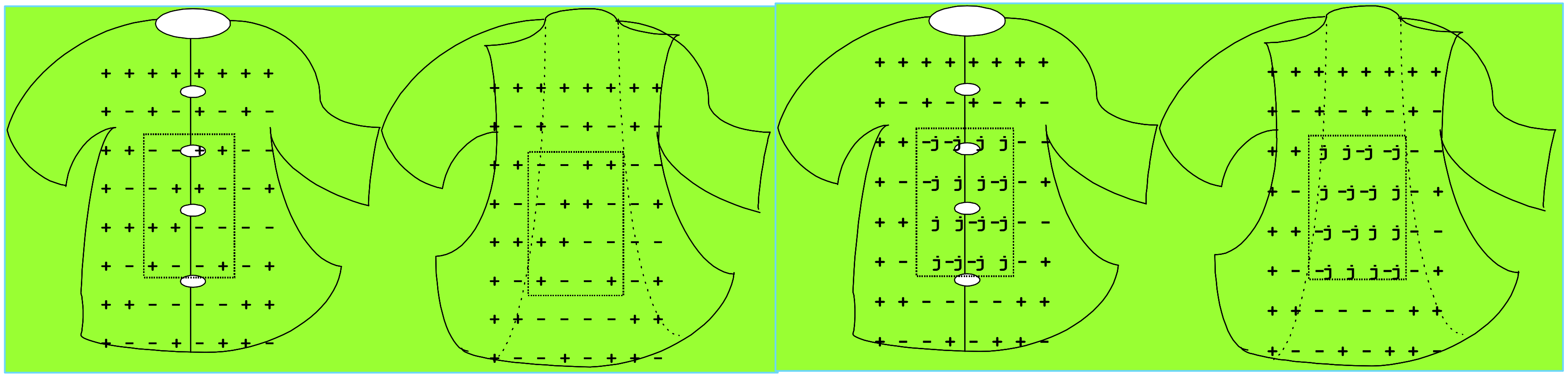
**wcu.chonbuk.ac.kr, mdmc.chonbuk.ac.kr**

**Institute of Information & Communication**
**Chonbuk National University**
**Jeonju, 561-756, Korea**
**Tel: +82632702463**
**Fax: +82632704166**

**htttp://en.wikipedia.org/wiki/Category:Matrices**
**htttp://en.wikipedia.org/wiki/Jacket:Matrix**
**http://en.wikipedia.org/wiki/user:leejacket**


Fourier
(1768-1830)
Galois
(1811-1832)
Hadamard
(1865-1963)
*Element-wise Inverse
*Linear Fraction
Jacket Matrix : Moon Ho Lee
(1985)
Fibonacci
(1170-1250)
Markov
(1856-1922)
Leonhard Euler
(1707-1783)

# Definition of Jacket matrix



Real Domain

Complex Domain

*The basic idea was motivated by the cloths of Jacket. As our two sided Jacket is inside and outside compatible, at least two positions of a Jacket matrix are replaced by their inverse; these elements are changed in their position and are moved, for example, from inside of the middle circle to outside or from to inside without loss of sign.*

# General Definition of Jacket matrix

In mathematics a **Jacket matrix** is a square matrix $A = aij$ of order $n$ whose entries are from a field (including real field, complex field, finite field ), if

$$AA^* = A^*A = nI_n$$

Where : $A^*$ is the transpose of the matrix of inverse entries of $A$ , i.e.

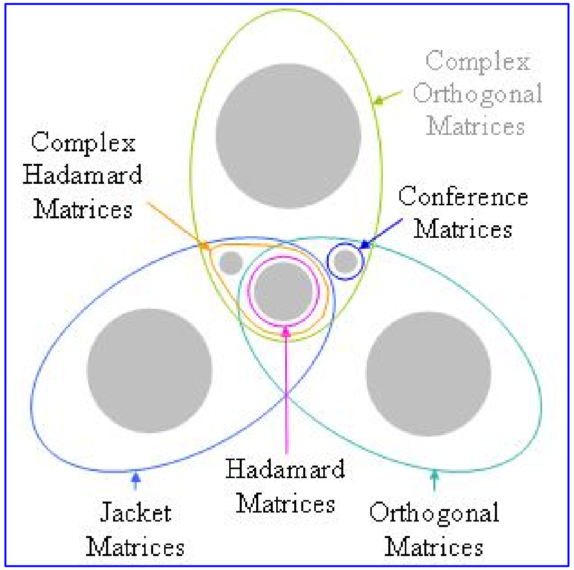
Written in different form is

$$u,v \in \{1,2,...,n\}, u \neq v : \sum \frac{a_{u,i}}{a_{v,i}} = 0$$

The inverse form which is only from the entrywise *inverse* and *transpose* : Jacket Matrices

$$[J]_{m\times n} = \begin{bmatrix} j_{0,0} & j_{0,1} & \cdots & j_{0,n-1} \\ j_{1,0} & j_{1,1} & \cdots & j_{1,n-1} \\ \vdots & \vdots & & \vdots \\ j_{m-1,0} & j_{m-1,1} & \cdots & j_{m-1,n-1} \end{bmatrix} \quad [J]^{-1}_{m\times n} = \frac{1}{C}\begin{bmatrix} 1/j_{0,0} & 1/j_{0,1} & \cdots & 1/j_{0,n-1} \\ 1/j_{1,0} & 1/j_{1,1} & \cdots & 1/j_{1,n-1} \\ \vdots & \vdots & & \vdots \\ 1/j_{m-1,0} & 1/j_{m-1,1} & \cdots & 1/j_{m-1,n-1} \end{bmatrix}^T$$

$$Orthogonal : u,v \in \{1,2,...,n\}, u \neq v : \sum_i a_{u,i}.a_{v,i} = 0; \sum_i a^2_{u,i} = const.$$



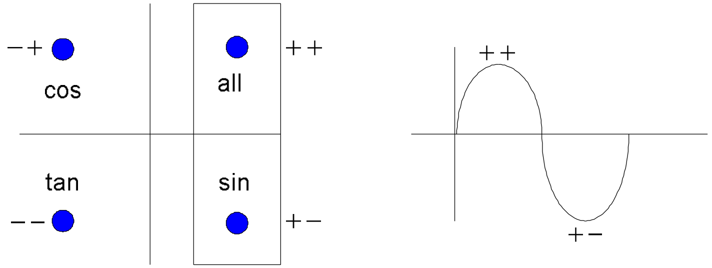
# Key Idea

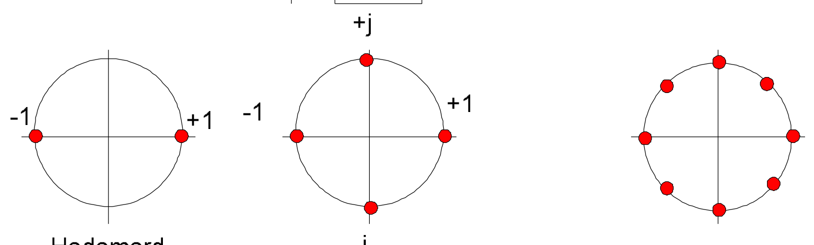
$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}_{-1=w} = \begin{bmatrix} a & b \\ b^T & c \end{bmatrix}$$

$$\left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{array}\right] \Rightarrow \begin{bmatrix} a & b \\ b^T & c \end{bmatrix} \text{ Orthogonal;}$$

$$\left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -w & w & -1 \\ \hline 1 & w & -w & -1 \\ 1 & -1 & -1 & 1 \end{array}\right] \quad w = 2^0, 2^1, j$$

**Descartes (1596~1650 France) Coordinates**

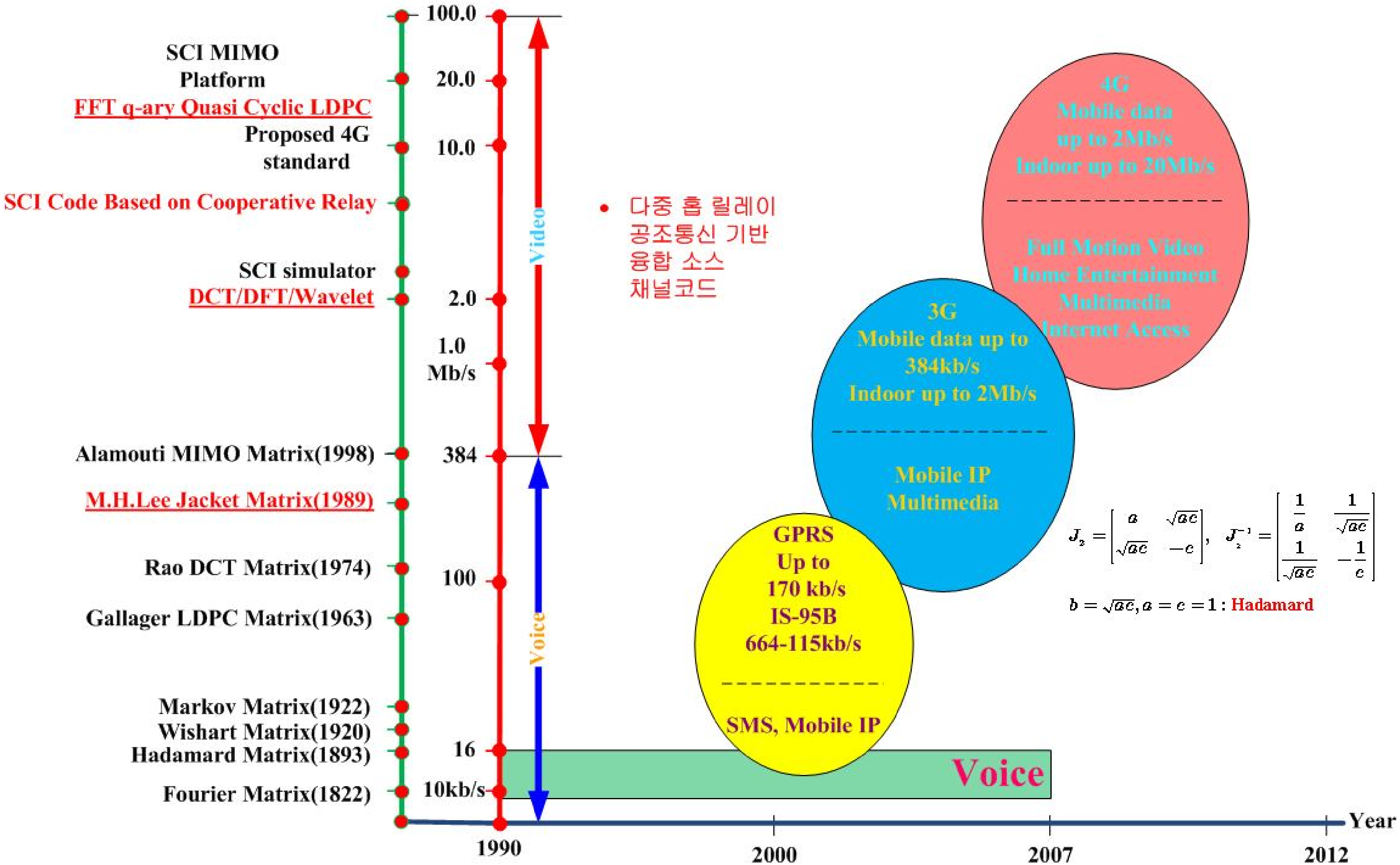


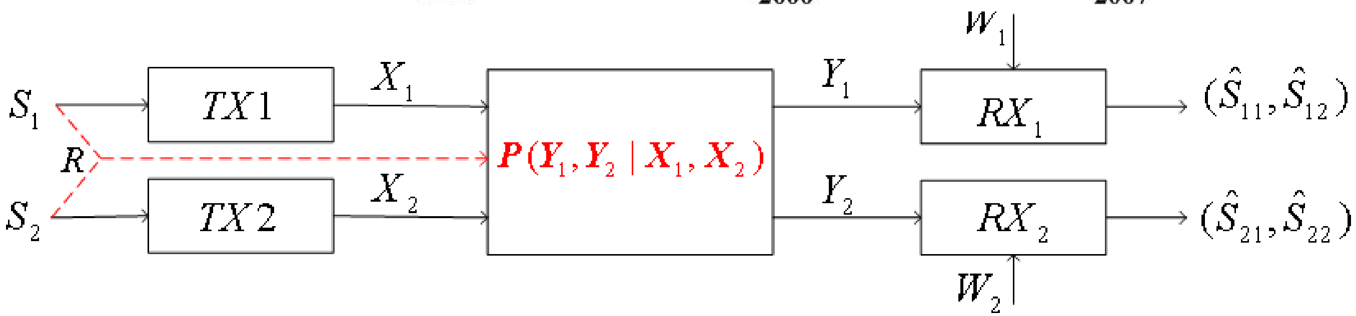

Hadamard

Jacket $\{\pm 1, \pm j\}$

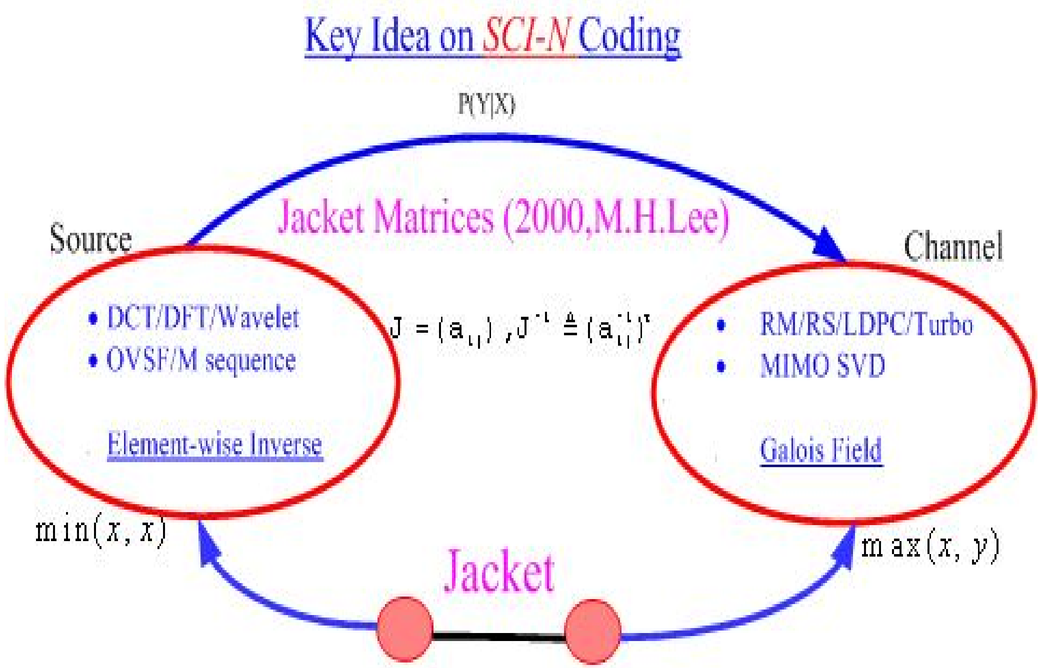
Fourier $\{\pm 1, e^{j\frac{2\pi}{N}}\}$



100.0
20.0
10.0
2.0
1.0 Mb/s
384
100
16
10kb/s
SCI MIMO Platform
FFT q-ary Quasi Cyclic LDPC
Proposed 4G standard
SCI Code Based on Cooperative Relay
SCI simulator
DCT/DFT/Wavelet
Alamouti MIMO Matrix(1998)
M.H.Lee Jacket Matrix(1989)
Rao DCT Matrix(1974)
Gallager LDPC Matrix(1963)
Markov Matrix(1922)
Wishart Matrix(1920)
Hadamard Matrix(1893)
Fourier Matrix(1822)
Video
Voice
• 다중 홉 릴레이 공조통신 기반 융합 소스 채널코드
4G Mobile data up to 2Mb/s Indoor up to 20Mb/s
Full Motion Video Home Entertainment Multimedia Internet Access
3G Mobile data up to 384kb/s Indoor up to 2Mb/s
Mobile IP Multimedia
GPRS Up to 170 kb/s IS-95B 664-115kb/s
SMS, Mobile IP
Voice
$J_2 = \begin{bmatrix} a & \sqrt{ac} \\ \sqrt{ac} & -c \end{bmatrix}, \quad J_2^{-1} = \begin{bmatrix} \frac{1}{a} & \frac{1}{\sqrt{ac}} \\ \frac{1}{\sqrt{ac}} & -\frac{1}{c} \end{bmatrix}$
$b = \sqrt{ac}, a = c = 1$ : Hadamard
1990
2000
2007
2012
Year

$S_1$
$S_2$
$R$
$TX1$
$TX2$
$X_1$
$X_2$
$P(Y_1, Y_2 \mid X_1, X_2)$
$Y_1$
$Y_2$
$W_1$
$W_2$
$RX_1$
$RX_2$
$(\hat{S}_{11}, \hat{S}_{12})$
$(\hat{S}_{21}, \hat{S}_{22})$




**Category : Matrices (from Http://en.wikipedia.org/wiki/Category:Matrices)**

**H**
- Hadamard matrix
- Hamiltonian matrix
- Hankel matrix
- Hasse-Witt matrix
- Hat matrix
- Hermitian matrix
- Hessenberg matrix

**I**
- Identity matrix
- Incidence matrix
- Integer matrix
- Invertible matrix
- Involutory matrix
- Irregular matrix

**J**
- **Jacket matrix**
- Jacobian matrix and determinant
- Jones calculus

**K**
- Kernel (matrix)
- Krawtchouk matrices

**L**
- Laplacian matrix
- Lehmer matrix
- Leslie matrix
- Levinson recursion
- List of matrices

# Center Weighted Hadamard Transform

Jacket Basic Concept from Center Weighted Hadamard

$$[WH]_4 \overset{\Delta}{=} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -2 & 2 & -1 \\ 1 & 2 & -2 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}, [WH]_4^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1/2 & 1/2 & -1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$[WH]_N \overset{\Delta}{=} [WH]_{N/2} \otimes [H]_2 \quad \text{where} \quad [H]_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Sparse matrix and its relation to construction of center weighted Hadamard

$$[WC]_N \overset{\Delta}{=} [H]_N [WH]_N = [WC]_{N/2} \otimes 2[I]_2 \qquad [WH]_N = \frac{1}{N}[H]_N [WC]_N$$

$$[WC]_N^{-1} = [WC]_{N/2}^{-1} \otimes \frac{1}{2}[I]_2 \qquad [WH]_N = N[WC]_N^{-1}[H]_N^{-1}$$

** Moon Ho Lee, "Center Weighted Hadamard Transform" IEEE Trans. on CAS, vol.26, no.9, Sept. 1989*

** Moon Ho Lee, and Xiao-Dong Zhang, "Fast Block Center Weighted Hadamard Transform" IEEE Trans. On CAS-I, vol.54, no.12, Dec. 2007.*

IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, VOL. 36, NO. 9, SEPTEMBER 1989

# The Center Weighted Hadamard Transform

MOON HO LEE

*Abstract* —The center weighted Hadamard transform (CWHT) is defined. This transform is similar to the Hadamard transform (HT) in that it requires only real operations. The CWHT, however, weights the region of mid-spatial frequencies of the signal more than the HT. A simple factorization of the weighted Hadamard matrix is used to develop a fast algorithm for the CWHT. The matrix decomposition is of the form of the Kronecker products of fundamental Hadamard matrices and successively lower order weighted Hadamard matrices.

## I. Introduction

The application of discrete orthogonal transforms for signal and image representation and compression is well known [1]–[4]. A much investigated method, due to the ease and efficiency of its

Manuscript received November 10, 1988. This work was supported in part by the Korean Science and Engineering Foundation. This letter was recommended by Associate Editor T. R. Viswanathan.

The author is with the Department of Information and Telecommunication Engineering, Chonbuk National University, Chonju 560-756, Korea.

IEEE Log Number 8926956.

implementation, is based on the Hadamard transform (HT) [8]. In this paper, we present a modification to the HT, which we call the center weighted Hadamard transform (CWHT). This method retains much of the simplicity of HT, but offers better quality of representation over the central region of the image [5]–[7]. The scheme was motivated by the fact that the human visual system is most sensitive to the mid-spatial frequencies [9]. The paper is organized as follows. In the next section the CWHT is introduced and recursive relations for the generation of the transform matrix is presented. Next, a fast CWHT algorithm which is similar to a fast HT method is derived for both the forward and inverse transforms that we have proved for examples.

## II. The Center Weighted Hadamard Transform (CWHT) and Its Fast Implementation

Let the Hadamard and the center weighted Hadamard matrices of order $N = 2^k$ be denoted by $[H]_N$ and $[WH]_N$, respectively. The CWHT of an $N \times 1$ vector $[f]$ and an $N \times N$ (image) matrix $[g]$ are given by

$$[F] = [\mathrm{WH}]_N [f] \tag{1}$$

$$[G] = [\mathrm{WH}]_N [g][\mathrm{WH}]_N. \tag{2}$$

The lowest order WH matrix is of size $(4 \times 4)$ and is defined as follows:

$$[\mathrm{WH}]_4 \triangleq \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -2 & 2 & -1 \\ 1 & 2 & -2 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}. \tag{3}$$

The inverse of (3) is

$$[\mathrm{WH}]_4^{-1} = \frac{1}{8}\begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & -1 & 1 & -2 \\ 2 & 1 & -1 & -2 \\ 2 & -2 & -2 & 2 \end{bmatrix}. \tag{4}$$

This choice of weighting was dictated, to a large extent, by the requirement of digital hardware simplicity [6]. As with the Hadamard matrix, a recursive relation governs the generation of higher order WH matrixes, i.e.,

$$[\mathrm{WH}]_N \triangleq [\mathrm{WH}]_{N/2} \otimes [H]_2, \tag{5}$$

where $\otimes$ is the kronecker product and $[H]_2$ is the lowest order Hadamard matrix given by [1]–[4], [8]:

$$[H]_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{6}$$

We now present a fast algorithm for the CWHT which is related to the fast HT (FHT) algorithm [2]–[4], [8]. The FHT can be derived by decomposing $[H]_N$ into a product of $k$ sparse matrices, each having rows/columns with only two nonzero elements. In order to develop a similar algorithm for the CWHT, define a coefficient matrix $[WC]_N$ by

$$[\mathrm{WC}]_N \triangleq [H]_N[\mathrm{WH}]_N. \tag{7}$$

Since $[H]_N^{-1} = 1/N[H]_N$, we have from (7) that

$$[\mathrm{WH}]_N = 1/N[H]_N[\mathrm{WC}]_N. \tag{8}$$

It is shown that $[\mathrm{WC}]_N$ is a sparse matrix with at most two nonzero elements per row and column. Therefore, the fast CWHT (FCWHT) is simply the FHT followed by a sparse matrix $1/N[\mathrm{WC}]_N$.

To show the sparseness of $[\mathrm{WC}]_N$ we start by computing the lowest order [WC], i.e. $[\mathrm{WC}]_4$.

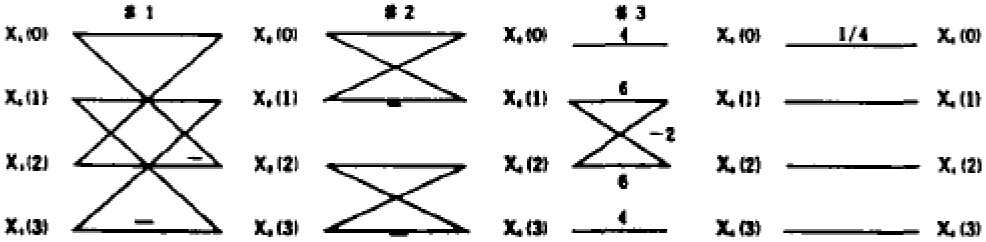



Fig. 1. Fast CWHT flow graph, $N = 4$.

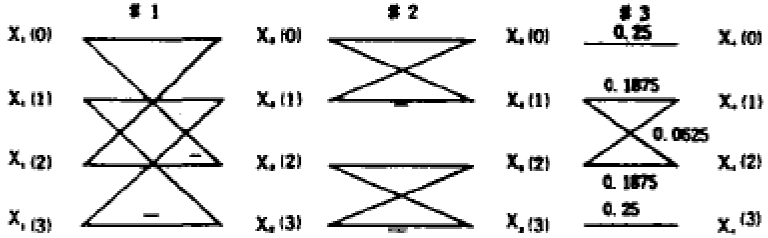



Fig. 2. Fast inverse CWHT flow graph, $N = 4$.

From (7), we have

$$[\mathrm{WC}]_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -2 & 2 & -1 \\ 1 & 2 & -2 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 6 & -2 & 0 \\ 0 & -2 & 6 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} \qquad (9)$$

Clearly $[\mathrm{WC}]_4$ is sparse. Using the expansion properties of the Hadamard and weighted Hadamard matrices, (7) can be written

Clearly $[\mathrm{WC}]_4$ is sparse. Using the expansion properties of the Hadamard and weighted Hadamard matrices, (7) can be written as

$$\begin{aligned}[\mathrm{WC}]_N &= ([H]_{N/2} \otimes [H]_2)([\mathrm{WH}]_{N/2} \otimes [H]_2) \\ &= ([H]_{N/2}[\mathrm{WH}]_{N/2}) \otimes ([H]_2[H]_2) \\ &= [\mathrm{WC}]_{N/2} \otimes (2[I]_2) \end{aligned} \tag{10}$$

where $[I]_2$ is the $2 \times 2$ identity matrix. Since $[\mathrm{WC}]_4$ is symmetric and has at most two non-zero elements in each row, it clearly follows from (10) that the same is true for $[\mathrm{WC}]_8$, and hence, for any $[\mathrm{WC}]_N$, $N = 2^k$, $k = 2,3,4,\cdots$. Fig. 1 shows a flow graph of the 4-point FCWHT algorithm. From this figure it is clear that the first three iterations of the algorithm are those of the FHT. These are followed by the operation of the $[\mathrm{WC}]_4$. The $N = 2^k$ point FCWHT algorithm requires $kN + N/2$ real additions and $1.5N$ real multiplications in contrast with the $N$-point FHT which requires $kN$ real additions.

The inverse FCWHT may be formulated in a similar fashion as the FCWHT. First we note that

$$[\mathrm{WC}]_N^{-1} = [\mathrm{WC}]_{N/2}^{-1} \otimes 1/2[I]_2.$$

Obviously,

$$\begin{aligned}[\mathrm{WC}]_N[\mathrm{WC}]_N^{-1} &= ([\mathrm{WC}]_{N/2} \otimes 2[I]_2)\left([\mathrm{WC}]_{N/2}^{-1} \otimes 1/2[I]_2\right) \\ &= \left([\mathrm{WC}]_{N/2}[\mathrm{WC}]_{N/2}^{-1}\right) \otimes ([I]_2[I]_2) = [I]_N. \end{aligned} \tag{11}$$

Equation (11) can be shown to be true by multiplying $[\mathrm{WC}]_N^{-1}$ in (11) by the expression for $[\mathrm{WC}]_N$ given in (10). From (11) and the sparseness and symmetry of $[\mathrm{WC}]_N^{-1}$ it follows that $[\mathrm{WC}]_N^{-1}$ is also symmetric and sparse. Furthermore, using (8), we have

$$[\mathrm{WH}]_N^{-1} = N[\mathrm{WC}]_N^{-1}[H]_N^{-1}. \tag{12}$$

But $[\mathrm{WC}]_N$ and $[\mathrm{H}]_N^{-1} = 1/N[H]_N$ are both symmetric with a symmetric product. Thus

$$[\mathrm{WH}]_N^{-1} = [H]_N[\mathrm{WC}]_N^{-1}. \tag{13}$$

Equation (13), with the exception of the scale factor $1/N$, is of the same form as (8). Consequently, it signifies a fast algorithm for the inverse of $[\mathrm{WH}]_N$ composed of FHT followed by $[\mathrm{WC}]_N$. Figs. 2 and 3 show a flow graph of the inverse FCWHT for $N=4$ and a signal flowchart of the FCWHT, respectively.

## III. Examples

The simple recursive relationship in (8) and (13) can now be used to formulate a sparse matrix decomposition of $[\mathrm{WH}]_N$ and $[\mathrm{WH}]_N^{-1}$. As an example for $N=8$, $[\mathrm{WH}]_8$ can be represented as

$$\begin{aligned}[\mathrm{WH}]_8 &= \frac{1}{8}([H]_4 \otimes [H]_2)([\mathrm{WC}]_4 \otimes 2[I]_2)\\ &= \frac{1}{8}([H]_4[\mathrm{WC}]_4) \otimes 2[H]_2\\ &= \frac{1}{8}([\mathrm{WC}]_4[H]_4) \otimes 2[H]_2 \qquad (14)\end{aligned}$$

i.e., $[\mathrm{WH}]_8$,

$$\frac{1}{8}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}4 & 0 & 0 & 0\\ 0 & 6 & -2 & 0\\ 0 & -2 & 6 & 0\\ 0 & 0 & 0 & 4\end{bmatrix} \otimes 2\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$

$$= \begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1\\ 1 & 1 & -2 & -2 & 2 & 2 & -1 & -1\\ 1 & -1 & -2 & 2 & 2 & -2 & -1 & 1\\ 1 & 1 & 2 & 2 & -2 & -2 & -1 & -1\\ 1 & -1 & 2 & -2 & -2 & 2 & -1 & 1\\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1\\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1\end{bmatrix}. \qquad (15)$$

In a similar manner as the FCWHT, we note that

$$\begin{aligned}[\mathrm{WH}]_8^{-1} &= ([H]_4 \otimes [H]_2)\left([\mathrm{WC}]_4^{-1} \otimes 1/2[I]_2\right)\\ &= \left([H]_4[\mathrm{WC}]_4^{-1}\right) \otimes 1/2[H]_2. \qquad (16)\end{aligned}$$

Therefore, (16) becomes

$$\begin{bmatrix}0.25 & 0 & 0 & 0\\ 0 & 0.1875 & 0.0625 & 0\\ 0 & 0.0625 & 0.1875 & 0\\ 0 & 0 & 0 & 0.25\end{bmatrix}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}$$

$$\otimes \frac{1}{2}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$$

$$= \frac{1}{16}\begin{bmatrix}2 & 2 & 2 & 2 & 2 & 2 & 2 & 2\\ 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2\\ 2 & 2 & -1 & -1 & 1 & 1 & -2 & -2\\ 2 & -2 & -1 & 1 & 1 & -1 & -2 & 2\\ 2 & 2 & 1 & 1 & -1 & -1 & -2 & -2\\ 2 & -2 & 1 & -1 & -1 & 1 & -2 & 2\\ 2 & 2 & -2 & -2 & -2 & -2 & 2 & 2\\ 2 & -2 & -2 & 2 & -2 & 2 & 2 & -2\end{bmatrix}. \qquad (17)$$

The symmetrical matrix decomposition of the $[\mathrm{WH}]_N$ and $[\mathrm{WH}]_N^{-1}$ are of the form of the kronecker products of lowest order Hadamard matrices and successively lower order weighted Hadamard matrices. Using the algebra of kronecker products, (8) and (10), it can be shown that

$$\begin{aligned}[\mathrm{WH}]_N[\mathrm{WH}]_N^{-1} &= 1/N\left(\{[H]_{N/2}[WC]_{N/2} \otimes 2[H]_2\}\right.\\ &\quad \left.\cdot\{[H]_{N/2}[WC]_{N/2}^{-1} \otimes 1/2[H]_2\}\right)\\ &= \left([\mathrm{WH}]_{N/2} \otimes [H]_2\right)\left([\mathrm{WH}]_{N/2}^{-1} \otimes 1/2[H]_2\right)\\ &= \left([\mathrm{WH}]_{N/2}[\mathrm{WH}]_{N/2}^{-1}\right) \otimes 1/2([H]_2[H]_2)\\ &= [I]_N. \qquad (18)\end{aligned}$$

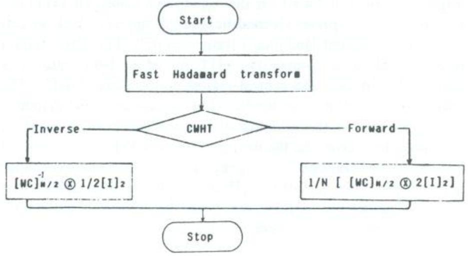



Fig. 3. Fast CWHT flow chart.

Below, we illustrate this relationship for the case of $N=8$.

$$[\mathrm{WH}]_8[\mathrm{WH}]_8^{-1} = \frac{1}{8}\left[\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -2 & 2 & -1\\ 1 & 2 & -2 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\right.$$

$$\left.\cdot\begin{bmatrix}2 & 2 & 2 & 2\\ 2 & -1 & 1 & -2\\ 2 & 1 & -1 & -2\\ 2 & -2 & -2 & 2\end{bmatrix}\right]$$

$$\otimes \frac{1}{2}\left[\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\right]$$

$$= [I]_8. \qquad (19)$$

Clearly (19) is the $8 \times 8$ identity matrix.

## IV. Conclusions

A new non-orthogonal transform, the CWHT was introduced in this paper. As in the case of the HT, the CWHT involves only real operations. Fast algorithm for the CWHT and its inverse were derived and shown to be of slightly higher complexity than similar HT algorithms. This method is presented for its simplicity and the clarity with which it decomposes a weighted Hadamard matrix in terms of the sparse matrices.

## References

[1] W. K. Pratt, J. Kane, and H. C. Andrews, "Hadamard transform image coding," *Proc. IEEE*, vol. 57, pp. 58–68, 1969.

[2] N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing*. Berlin, Germany: Springer-Verlag, 1975.

[3] D. F. Elliolt and K. R. Rao, *Fast Transforms Algorithms, Analyses and Applications*. Orlando, Fl: Academic, 1982.

[4] K. R. Rao, *Discrete Transforms and Their Applications*. Reading, MA: Addison Wesley, 1985.

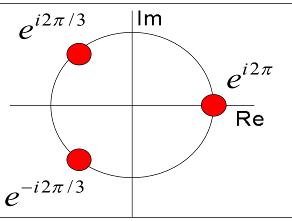
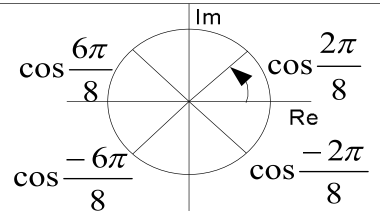
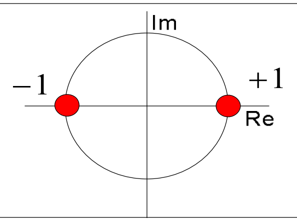
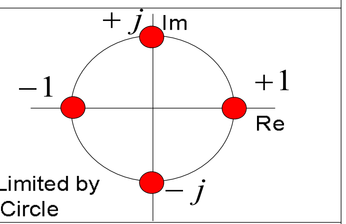
[5] M. H. Lee and D. Y. Kim, "Weighted Hadamard transformation for S/N ratio enhancement in image transformation," in *IEEE ISCAS Proc.*, pp. 65–68, May 1984.

[6] M. H. Lee, "Image data processing by weighted Hadamard transform," Project Rep., Korean Science and Engineering Foundation, Oct. 1985.

[7] M. H. Lee and Y. S. Lim, "On fast weighted Hadamard transform," in *IEEE Tencon '87*, Seoul, Korea, pp. 935–938, 1987.

[8] M. H. Lee, M. Kaveh, "Fast Hadamard transform based on a simple matrix factorization," *IEEE Trans., Acoust. Speech, Signal Processing*, vol. ASSP-34, pp. 1666–1667, Dec. 1986.

[9] M. H. Lee and J. J. Hwang, "A visual model weighted Hadamard transform," presented at Picture Coding Symp., Torino, Italy, Sept. 12–14, 1988.

| | DFT (1822)<br>J. Fourier | DCT(1974)<br>N. Ahmed, K.R. Rao,et. | Hadamard (1893)<br>J. Hadamard | Jacket(1989)*<br>Moon Ho Lee |
|---|---|---|---|---|
| Formula | $X(n)=\sum_{k=0}^{N-1} x(k)w^{nk}$<br>$n=0,1\ldots N-1, w=e^{-j2\pi/N}$ | $[C_N]_{m,n}=\sqrt{\frac{2}{N}}k_m\cos\frac{m(n+\frac{1}{2})\pi}{N}$<br>$m,n=0,1,\ldots,N-1$ | $[H]_2=\begin{bmatrix}1&1\\1&-1\end{bmatrix}$<br>$[H]_n=[H]_{n/2}\otimes[H]_2$ | $[J]_4=\begin{bmatrix}1&1&1&1\\1&-j&j&-1\\1&j&-j&1\\1&-1&-1&1\end{bmatrix}$<br>$[J]_n=[J]_{n/2}\otimes[H]_2 \qquad n>4$ |
| Forward | $N=3$<br>$F_3=\begin{bmatrix}1&1&1\\1&w&w^2\\1&w^2&w\end{bmatrix}$<br>$w=e^{-i2\pi/3}$ | $N=4$<br>$[C]_4=\begin{bmatrix}\frac{1}{\sqrt{2}}&\frac{1}{\sqrt{2}}&\frac{1}{\sqrt{2}}&\frac{1}{\sqrt{2}}\\C_8^1&C_8^3&C_8^5&C_8^7\\C_8^2&C_8^6&C_8^6&C_8^2\\C_8^3&C_8^7&C_8^1&C_8^5\end{bmatrix}$<br>$C_8^i=\cos\frac{i\pi}{8}$ | $(-1)^{\oplus_{k=0}^{n-1} i_k j_k}$<br>$[H]_4=\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}$ | $(-1)^{\oplus_{k=0}^{n-1} i_k j_k}\, w^{(i_{n-2}\oplus i_{n-1})(j_{n-2}\oplus j_{n-1})}$<br>$[J]_4=\begin{bmatrix}1&1&1&1\\1&-w&w&-1\\1&w&-w&-1\\1&-1&-1&1\end{bmatrix}$<br>$w$=1: Hadamard<br>$w$=2: Center Weighted Hadamard |
| Inverse | Element-Wise Inverse<br>$F_3^{-1}=\frac{1}{3}\begin{bmatrix}1&1&1\\1&w^{-1}&w^{-2}\\1&w^{-2}&w^{-1}\end{bmatrix}$ | Block-Wise Inverse<br>$[C]_4^{-1}=\frac{1}{2}\begin{bmatrix}\frac{1}{\sqrt{2}}&C_8^1&C_8^2&C_8^3\\\frac{1}{\sqrt{2}}&C_8^3&C_8^6&C_8^7\\\frac{1}{\sqrt{2}}&C_8^5&C_8^6&C_8^1\\\frac{1}{\sqrt{2}}&C_8^7&C_8^2&C_8^5\end{bmatrix}$ | Element-Wise Inverse<br>$[H]_4^{-1}=\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}$ | Element-Wise Inverse<br>or<br>Block-Wise Inverse<br>$[J]_4^{-1}=\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-1/w&1/w&-1\\1&1/w&-1/w&-1\\1&-1&-1&1\end{bmatrix}$ |
| Circle |  |  |  | <br>No Limited by Circle |
| Kronecker | $[DFT]_N\otimes[DFT]_N\neq[DFT]_{2N}$ | $[DCT]_N\otimes[DCT]_N\neq[DCT]_{2N}$ | $[H]_N\otimes[H]_N=[H]_{2N}$ | $[J]_N\otimes[J]_N=[J]_{2N}$ |
| Size | $2^n$ or $p$ :p is prime | $2^n$ | $2^n, 4n$ | *Arbitrary* |

# Why use Jacket Matrices?

Jacket Definition: element inverse and transpose

$$[J]_{m\times n} = \lfloor L_{ij} \rfloor_{m\times n} \quad \text{and} \quad \left([J]_{m\times n}\right)^{-1} = \left([1/L_{ij}]_{m\times n}\right)^{T} \quad \text{Simple Inverse}$$

Examples:

$$[J]_2 = \begin{bmatrix} 1 & 1 \\ 1 & \alpha \end{bmatrix} \qquad \left([J]_2\right)^{-1} = \begin{bmatrix} 1 & 1 \\ 1 & \alpha \end{bmatrix}$$

where
$1+\alpha = 0, \alpha^2 = 1$

$$[J]_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 \\ 1 & \alpha^2 & \alpha \end{bmatrix} \qquad \left([J]_3\right)^{-1} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \alpha^2 & \alpha \\ 1 & \alpha & \alpha^2 \end{bmatrix}$$

where
$1+\alpha+\alpha^2 = 0, \alpha^3 = 1$

$$[J]_4 \stackrel{\Delta}{=} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -i & -1 \\ 1 & -i & i & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \qquad \left([J]_4\right)^{-1} \stackrel{\Delta}{=} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & i & -1 \\ 1 & i & -i & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$[H]_4 \stackrel{\Delta}{=} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

# A New Reverse Jacket Transform and Its Fast Algorithm

Moon Ho Lee, *Senior Member, IEEE*

***Abstract*—This paper presents the Reverse Jacket transform [RJT] and a simple decomposition of its matrix, which is used to develop a fast algorithm for the RJT. The matrix decomposition is of the form of the matrix products of Hadamard matrices and successively lower order coefficient matrices. This decomposition very clearly leads to a block circular sparse matrix factorization of the Reverse Jacket $[RJ]_N$ matrix. The main property of $[RJ]_N$ is that the inverse matrices of its elements can be obtained very easily and have a special structure. $[RJ]_N$ is derived using the Weighted Hadamard transform corresponding to the Hadamard matrix $[H]_N$ and a basic symmetric matrix A. Each element of $[RJ]_N$ is a generalized for polygonal subsampling and canonical Smith form. In this paper we represent in particular the systematical block-wise sparse matrix of extending-method for $[RJ]_N$.**

## I. Introduction

THE HADAMARD transform is an orthogonal matrix with highly practical value for representing signals and images especially for the purposes of data compression [1]–[4], [11]. The reason for the practicality of this transform is the fact that the elements of the Hadamard matrix are either $+2^0(=1)$ or $-2^0(=-1)$ Thus, the computation of the transform of a signal consists of additions and subtractions of the signal samples.

Recently, the Hadamard matrix has been presented in that the Walsh-Hadamard transform is the most known of the nonsinusoidal orthogonal transforms. The Walsh-Hadamard matrix is used for the Walsh representation of the data sequences in image coding and for Hadamard–Walsh orthogonal- sequence generator in code-division multiple access (CDMA) spread-spectrum communication. Their basic functions are sampled Walsh functions which can be expressed in terms of the Hadamard $[H]_N$ matrices. Using the orthogonality of Hadamard matrices, we construct a generalized Weighted Hadamard (WH) matrice [5], [6], [8] called an $[RJ]_N$ matrix with a reverse geometric structure. In this paper, $[RJ]_N$ and its five cases of matrix examples are described. $[RJ]_N$ is nonorthogonal but its Hadamard matrix, which is a subset of $[RJ]_N$ [6], [8], is orthogonal.

Just as in the case of the fast Fourier transform (FFT), several algorithms have been developed for computing $N$-length Hadamard transforms in $N \log_2 N$, rather than $N^2$ operations. These are usually generalizations of the Cooley-Tukey FFT algorithm [4]. In this paper, we propose a simple recursive factorization for the [RJ] in terms of the Kronecker products of 2 * 2 [RJ] and Hadamard matrices of consecutively lower orders. A consequence of this factorization is a simple and clear fast Hadamard transform algorithm resulting from a block circulant sparse matrix factorization of the [RJ] matrix.

Section II summarized the Weighted Hadamard transform (WHT), and Section III described the RJ matrix. Section IV explain the case of RJ matrices and its fast implementation, and Section V deals with the conclusion.

Manuscript received April 13, 1998; revised August 1999. This paper was recommended by Associate Editor T. Fujii.

The author is with the Institute of Information & Communication, Department of Information and Communication Engineering, Chonbuk National University, Chonju 561-756, Korea.

Publisher Item Identifier S 1057-7130(00)00582-6.

## II. The Conventional WHT

Let the Hadamard and the Weighted Hadamard matrices of order $N = 2^k$ be denoted by $[H]_N$ and $[WH]_N$, respectively. The WH transform of an $N \times 1$ vector $[f]$ and an $N * N$ matrix (image) $[g]$ are given by [5], [6], [10]

$$[F] = [WH]_N [f] \tag{1}$$

$$[G] = [WH]_N [g] [WH]_N^T. \tag{2}$$

The lowest order WH matrix is of size (4 * 4) and is defined as follows:

$$[WH]_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -2 & 2 & -1 \\ 1 & 2 & -2 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 6 & -2 & 0 \\ 0 & -2 & 6 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}. \tag{3}$$

The inverse of (3) is

$$[WH]_4^{-1} = \frac{1}{8} \begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & -1 & 1 & -2 \\ 2 & 1 & -1 & -2 \\ 2 & -2 & -2 & 2 \end{bmatrix} = \frac{1}{16} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}. \tag{4}$$

This choice of weighting was dictated, to a large extent, by the requirement of digital hardware simplicity [5], [6]. As with the Hadamard matrix, a recursive relation governs the generation of higher order WH matrices, i.e.,

$$[WH]_N \triangleq [WH]_{N/2} \otimes [H]_2 \tag{5}$$

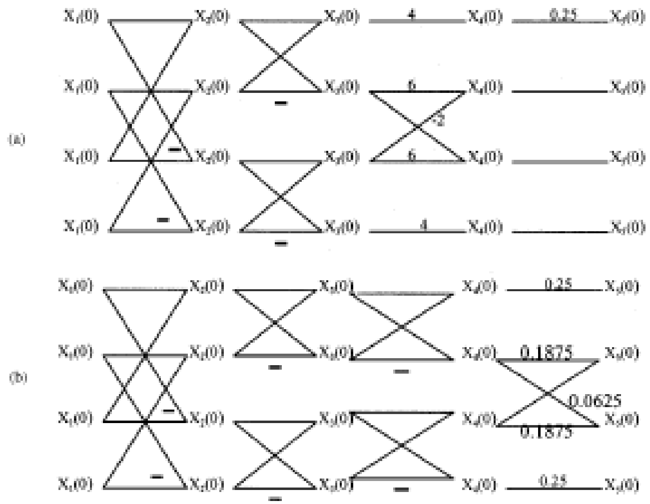



Fig. 1. The fast algorithm for Case II of $[\mathrm{RJ}]_4$ and $[\mathrm{RJ}]_4^{-1}$ flow graph. (a) Forward. (b) Inverse.

where $\otimes$ is the Kronecker product and $[H]_2$ is the lowest order Hadamard matrix given by [1]–[3], [13], [14]

$$[H]_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (6)$$

For example, for $N = 8$

$$[\mathrm{WH}]_8 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -2 & 2 & -1 \\ 1 & 2 & -2 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -2 & -2 & 2 & 2 & -1 & -1 \\ 1 & -1 & -2 & 2 & 2 & -2 & -1 & 1 \\ 1 & 1 & 2 & 2 & -2 & -2 & -1 & -1 \\ 1 & -1 & 2 & -2 & -2 & 2 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}. \quad (7)$$

In order to develop a similar general algorithm for the WHT, define a weighed coefficient matrix $[\mathrm{RC}]_N$ by

$$[\mathrm{RC}]_N \triangleq [H]_N [\mathrm{WH}]_N. \quad (8)$$

Clearly, $[\mathrm{RC}]_4$ is sparse matrix. Using the expansion properties of Hadamard and WH matrices, (8) can be written as

$$\begin{aligned} [\mathrm{RC}]_N &= ([H]_{N/2} \otimes [H]_2)([\mathrm{WH}]_{N/2} \otimes [H]_2) \\ &= ([H]_{N/2}[\mathrm{WH}]_{N/2}) \otimes ([H]_2[H]_2) \\ &= [\mathrm{RC}]_{N/2} \otimes 2[I]_2 \end{aligned} \quad (9)$$

where $[I]_2$ is the $2 \times 2$ identity matrix. Since $[\mathrm{RC}]_4$ is symmetric and has at most two nonzero elements in each row, it clearly follows from (9) that the same is true for $[\mathrm{RC}]_8$, and hence, for any $[\mathrm{RC}]_N$, $N = 2^k$, $k = 2, 3, 4, \ldots$. Fig. 1 shows a flow graph of the 4-point fast WHT being the same as RJ transform [RJT]. From this figure it is clear that the first two iterations of the algorithm are those of the fast Hadamard transform. These are followed by the operation of the $[\mathrm{RC}]_4$ such as $\log_2 N + 1$.

Since $[H]_N^{-1} = 1/N[H]_N$ we have from (10) that

$$[\mathrm{WH}]_N = 1/N[H]_N[\mathrm{RC}]_N. \quad (10)$$

Furthermore, using (9), we have

$$[\mathrm{WH}]_N^{-1} = N[\mathrm{RC}]_N^{-1}[H]_N^{-1}. \quad (11)$$

The symmetrical matrix decomposition of the $[\mathrm{WH}]_N$ and $[\mathrm{WH}]_N^{-1}$ are of the form of the Kronecker products of lowest order Hadamard matrices and successively lower order weighted Hadamard matrices. Using the algebra of Kronecker products (5) and (11), it can be shown that

$$\begin{aligned} [\mathrm{WH}]_N[\mathrm{WH}]_N^{-1} &= 1/N(\{[H]_{N/2}[\mathrm{RC}]_{N/2} \otimes 2[H]_2\} \\ &\quad \cdot \{[H]_{N/2}[\mathrm{RC}]_{N/2}^{-1} \otimes 1/2[H]_2\}) \\ &= ([\mathrm{WH}]_{N/2} \otimes [H]_2)([\mathrm{WH}]_{N/2}^{-1} \otimes 1/2[H]_2) \\ &= ([\mathrm{WH}]_{N/2}[\mathrm{WH}]_{N/2}^{-1} \otimes 1/2([H]_2[H]_2) \\ &= [I]_N. \end{aligned} \quad (12)$$

clearly (12) is the $N \times N$ identity matrix.

## III. The Proposed RJT

The $[\mathrm{RJ}]_N$ are a generalized conventional $[\mathrm{WH}]_N$ and $[H]_N$ [6], [7], [9]. The [RJT] having geometric structure property. The basic idea of this paper was motivated by the cloths of RJ. As our two-sided jacket is inside and outside compatible, at least two positions of a RJ matrix $[\mathrm{RJ}]_N$ are replaced by their inverse; these elements are changed in their position and are moved, for example, from inside of the middle circle to outside or from to inside without loss of signs; this is a very interesting phenomenon. This is the reason why we call it an RJ matrix.

The WHT was first introduced in [5], [6], [10], and fast algorithms in terms of Hadamard matrices have been investigated in [1], [2]. Basically, a WH matrix is a slight modification of a Hadamard matrix. The WH, however, weights the region of mid-spatial frequencies of the signal more than the Hadamard transform [5]. The basic concept of this paper is derived from the $2 \times 2$ *subsampling matrix of Weighted Hadamard matrix.*

[RJT] is straightforward and shows that the matrices $[RJ]_{2^k}$ have the following properties.

1) $[RJ]_{2^k}$ is a symmetric matrix, i.e.,

$$[RJ]_{2^k} = [RJ]_{2^k}^T \tag{13}$$

where $T$ denotes transpose.

2) $[RJ]_{2^k}$ is an orthogonal matrix, in Case of *RJ-I* (33), as basic matrix elements $a = b = c = 1$

$$[RJ]_{2^k}^T [RJ]_{2^k} = 2^k [I]_{2^k} \tag{14}$$

but is a different case from a nonorthogonal matrix with *RJ-II–V*, as (36), (40), (43), (46)

$$[RJ]_{2^k}^T [RJ]_{2^k} = [RJ]_{2^k}^2. \tag{15}$$

*Definition 3:* A $(2n \times 2n)$ matrix $A = (a_{ij})_{i,j=1}^{2n}$, $n \in \mathbb{N}$ is called *Hamiltonian* if $[AJ] = [AJ]^T$, with $J = \left[\begin{smallmatrix} 0 & I_n \\ -I_n & 0 \end{smallmatrix}\right]$, where $I_n \in \mathbb{R}^{2^k \times 2^k}$ is the unit matrix.

*Definition 3.2:* We define one more notion related to the Hadamard matrix $[H]_{2^k} \in \mathbb{R}^{2^k \times 2^k}$. Let $[RJ]_{2^k} \in \mathbb{R}^{2^k \times 2^k}$ be a $2^k \times 2^k$ matrix. A $2^k \times 2^k$ matrix $[RJ]_{2^k}$ such that

$$[RJ]_{2^k} = [H]_{2^k}^{-1} [RJ]_{2^k} [H]_{2^k} \tag{16}$$

is called the RJ matrix, where $k$ belongs to integer $\mathbb{N}$, and $\mathbb{R}$ is a real number. All its components are $\pm 2^n$, $n = 0, 1, 2$. Equation (13) is easily found to be equivalent to the conditions

$$[RJ]_{2^k}^n = [H]_{2^k}^{-1} [RJ]_{2^k}^n [H]_{2^k} \tag{17}$$

$$[RJ]_{2^k}^n = [RJ]_{2^k}^{[n]} \tag{18}$$

$$[RJ]_{2^k}^{-1} = \frac{1}{\det([RJ]_{2^k})} [RJ]_{2^k}^{[n]}, \quad k \geq 2. \tag{19}$$

We carefully consider the $4 \times 4$ [WH] matrix

$$[WH]_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -2 & 2 & -1 \\ 1 & 2 & -2 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}. \tag{20}$$

If we regard the upper-left $2 \times 2$ block subsampling matrix of [WH], then we can find some a regular recursive geometric structure.

The upper left $2 \times 2$ block subsampling given is

$$[WH]_2 = \begin{bmatrix} 1 & 1 \\ 1 & -2 \end{bmatrix}. \tag{21}$$

Therefore, we define several matrices for [RJ] as follows:

$$\begin{aligned} [Z]_2 &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \text{called } Sip \text{ matrix} \\ S_2 &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ J_2 &= \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \end{aligned} \tag{22}$$

Let $I_{2^k}$ and $0_{2^k}$, $k \in \mathbb{N}$ be the $(2^k \times 2^k)$ unit matrix and zero matrix, respectively. Then, for $k \geq 1$, we get

$$\begin{aligned} Z_{2^{k+1}} &= \begin{bmatrix} I_{2^k} & 0_{2^k} \\ 0_{2^k} & -I_{2^k} \end{bmatrix}, \quad S_{2^{k+1}} = \begin{bmatrix} 0_{2^k} & I_{2^k} \\ I_{2^k} & -0_{2^k} \end{bmatrix} \\ J_{2^{k+1}} &= \begin{bmatrix} 0_{2^k} & I_{2^k} \\ -I_{2^k} & 0_{2^k} \end{bmatrix}. \end{aligned} \tag{23}$$

Further, we define

$$\begin{aligned} \Lambda_2 &= \begin{bmatrix} a & b \\ b^T & -c \end{bmatrix} = M_1, \quad a, b, c \neq 0, \quad a \leq b \leq c \\ M_2 &= Z_2 \Lambda_2 S_2 \\ M_4 &= J_2^{-1} \Lambda_2 J_2 \\ T &= \text{transpose matrix} \end{aligned} \tag{24}$$

The determinant of $\Lambda_2$ such as scattering matrix [12], [18] is

$$\det \begin{pmatrix} a & b \\ b^T & c \end{pmatrix} = \det a \det(c - ba^{-1} b^T). \tag{25'}$$

Equation (25) is proved in the Appendix. Finally, we obtain $[RJ]_4$, defined by

$$[RJ]_4 \triangleq \begin{bmatrix} M_1 & M_2 \\ M_2^T & M_4 \end{bmatrix} = \begin{bmatrix} a & b & b & a \\ b & -c & c & -b \\ b & c & -c & -b \\ a & -b & -b & a \end{bmatrix}. \tag{26}$$

Throughout the whole paper, we assume that $\Lambda$ is invertible. We define the inverse RJ matrix as follows:

$$[RJ]_{2^k}^{-1} = C_k L_k, \quad k \in \{1, 2\}, \quad C_k \in \mathbb{R} \tag{27}$$

where $C_1 = \frac{-\text{signum}(a \cdot c)}{\det([RJ]_2)}$, $C_2 = \frac{1}{\text{lcm}(a,b,c)}$, and lcm denotes least common multiple

$$\begin{aligned} L_1 &= \text{signum}(a \cdot c) \begin{bmatrix} c & b \\ b & -a \end{bmatrix} \\ \text{signum}\, x &= \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} \\ L_2 &= \text{lcm}(a, b, c) \begin{bmatrix} 1/a & 1/b & 1/b & 1/a \\ 1/b & -1/c & 1/c & -1/b \\ 1/b & 1/c & -1/c & -1/b \\ 1/a & -1/b & -1/b & 1/a \end{bmatrix}. \end{aligned} \tag{28}$$

The matrices $[RJ]_{2^k}$, $k \geq 3$ will be defined subsequently. The Sylvester construction for RJ matrices can be expressed recursively in terms of Kronecker product [1], [13], [14]

$$[RJ]_{2^{k+1}} = [RJ]_{2^k} \otimes [H]_2 = [RJ]_{2^k} \otimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad k \geq 2. \tag{29}$$

For example, as $[RJ]_8$ is an RJ matrix, then it is

$$[RJ]_8 = \begin{bmatrix} [H]_2 & [H]_2 & [H]_2 & [H]_2 \\ [H]_2 & -2[H]_2 & 2[H]_2 & -[H]_2 \\ [H]_2 & 2[H]_2 & -2[H]_2 & -[H]_2 \\ [H]_2 & -[H]_2 & -[H]_2 & [H]_2 \end{bmatrix}. \tag{30}$$

Now, in a similar fashion as (9), we note that

$$\begin{aligned} [RC]_{2^{k+1}} &= [H]_{2^{k+1}}[RJ]_{2^{k+1}} \\ &= ([H]_{2^k} \otimes [H]_2)([RJ]_{2^k} \otimes [H]_2) \\ &= ([H]_{2^k}[RJ]_{2^k}) \otimes ([H]_2[H]_2) \\ &= [RC]_{2^k} \otimes 2[I]_{2^k}, \quad k \geq 2. \end{aligned} \tag{31}$$

Since $[H]_{2^k}^{-1} = 1/2^k [H]_{2^k}$, we have from (31)

$$[RJ]_{2^{k+1}} = \frac{1}{2^{k+1}}[H]_{2^{k+1}}[RC]_{2^{k+1}}. \tag{32}$$

It can be shown easily that $[RC]_{2^k}$ is a sparse matrix. Therefore, the fast RJ transform is simply the fast Hadamard transform followed by a sparse matrix $1/2^k[RC]_{2^k}$, $k \geq 2$.

## IV. Five Cases of Reverse Jacket Matrices and Their Fast Algorithm

There are five cases in the elements decision of basic symmetric matrix $\Lambda_2$, which is $[RJ]_2$. The matrix $\Lambda_2$ consists of three elements ($a$, $b$, and $c$), which are all non-zero and take the values of $\pm 2^n$, $n = 0, 1, 2$. Their conditions are: 1) $a = b = c$; 2) $a = b \neq c$; 3) $a = c \neq b$; 4) $a \neq b = c$; and 5) $a \neq b \neq c$. This matrix can be used for multidimensional subsampling of signals.

*1) Case I:* The elements are all the same, with $a = b = c$. For example, let the elements be all ones, $a = b = c = 2^0 = 1$, and the basic symmetric matrix $M_1 = \Lambda = [RJ]_2$ follows. This is the same as the Hadamard matrix. The Hadamard matrix can be considered a subset of the RJ matrix. We have from (24)

$$\begin{aligned} M_1 = [RJ]_2 &= \begin{bmatrix} a & b \\ b^T & -c \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ &= \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{33}$$

and its inverse matrix

$$[RJ]_2^{-1} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}. \tag{34}$$

From (32), we have

$$[RJ]_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

and its inverse matrix is (35)

$$[RJ]_4^{-1} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

The $[RJ]_2^{-1}$ is determined to $C_2L_2$ and is $\frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. Since the determinant of $[RJ]_2$ is 2, the sampling ratio and the number of subchannels are 2. The $[RJ]_4$ is the same as the Hadamard matrix $[H]_4$, which is orthogonal symmetric matrix and is special case of $[RJ]_4$ and subsampling matrix in this case [8]. The $[RJ]_4$ and its inverse matrix $[RJ]_4^{-1}$ consist of same elements. When this is used for data coding, the transmitter unit is the same as the receiver unit. This is the same as the inside and outside of the RJ [15], [16], [].

*2) Case II:* The elements' condition of Case II is $a = b \neq c$. If $a$ and $b$ are 1's, and $c$ is a 2, then in this case, $[RJ]_2$ is the same as $[WH]_2$ and the symmetric matrix. It is a polygonal subsampling matrix. The signs of each corresponding element between forward and inverse matrices are the same.

The $[RJ]_2$ is $\begin{bmatrix} 1 & 1 \\ 1 & -2 \end{bmatrix}$ and $[RJ]_2^{-1}$ is $\begin{bmatrix} 2 & 1 \\ 1 & -1 \end{bmatrix}$. In this case, the sampling ratio and subchannel are 3 and are easily decomposed as a Smith form [7]–[9]

$$\begin{bmatrix} 1 & 1 \\ 1 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \mathrm{U\Lambda V}. \tag{36}$$

The co-set vectors are $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $\begin{bmatrix} 2 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$. As an example

$$\begin{aligned} M_2 = [RJ]_2 &= \begin{bmatrix} a & b \\ b & -c \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -2 \end{bmatrix} \\ &= \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 2 & -1 \\ 0 & 3 \end{bmatrix} \end{aligned} \tag{37}$$

and its inverse matrix

$$[RJ]_2^{-1} = \frac{1}{3}\begin{bmatrix} 2 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{6}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 3 & 0 \\ 1 & 2 \end{bmatrix}. \tag{38}$$

Also, we have $[RJ]_4$

$$[RJ]_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -2 & 2 & -1 \\ 1 & 2 & -2 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

and its inverse matrix is (39)

$$[RJ]_4^{-1} = \frac{1}{8}\begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & -1 & 1 & -2 \\ 2 & 1 & -1 & -2 \\ 2 & -2 & -2 & 2 \end{bmatrix}$$

In geometric structure, the outside ($\pm 1$'s area) of $[RJ]_4$ is corresponding to inside ($\pm 2$'s area) of $[RJ]_4^{-1}$ and vice versa. The

Case II matrix is formed of Multivariate Gaussian channel capacity [15]–[18].

*3) Case III:* In this case the elements condition is $a = c \neq b$. Let the elements be $a = c = 1$ and $b = 2$. Then, we obtain the basic matrix $\Lambda_2$. $[RJ]_2$ is $\begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix}$ and the $[RJ]_2^{-1}$ is $\frac{1}{5}\begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix}$ which is $1/5[RJ]_2$. The $[RJ]_4$ and $[RJ]_4^{-1}$ are symmetric matrix as follows:

$$\begin{aligned} M_3 = [RJ]_2 &= \begin{bmatrix} a & b \\ b & -c \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix} \\ &= \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 3 & 1 \\ -1 & 3 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 2 & -1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \end{aligned} \quad (40)$$

and its inverse matrix

$$[RJ]_2^{-1} = \frac{1}{5}\begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix} = \frac{1}{10}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 3 & 1 \\ -1 & 3 \end{bmatrix}. \quad (41)$$

i.e., $[RJ]_4$ where

$$[RJ]_4 = \begin{bmatrix} 1 & 2 & 2 & 1 \\ 2 & -1 & 1 & -2 \\ 2 & 1 & -1 & -2 \\ 1 & -2 & -2 & 1 \end{bmatrix}$$

and its inverse matrix is (42)

$$[RJ]_4^{-1} = \frac{1}{8}\begin{bmatrix} 2 & 1 & 1 & 2 \\ 1 & -2 & 2 & -1 \\ 1 & 2 & -2 & -1 \\ 2 & -1 & -1 & 2 \end{bmatrix}.$$

The two positions of $[RJ]_4$ can be replaced by $[RJ]_4^{-1}$.

*4) Case IV:* The elements' condition is $a \neq b = c$, and let them be $a = 1$ and $b = c = 2$. Then the $[RJ]_2$ is $\begin{bmatrix} 1 & 2 \\ 2 & -2 \end{bmatrix}$, and $[RJ]_2^{-1}$ is $1/6\begin{bmatrix} 2 & 2 \\ 2 & -1 \end{bmatrix}$. The $4 \times 4$ forward and inverse RJ matrix is the following symmetric matrix:

$$\begin{aligned} M_4 = [RJ]_2 &= \begin{bmatrix} a & b \\ b & -c \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & -2 \end{bmatrix} \\ &= \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 3 & 0 \\ -1 & 4 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 2 & -1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 6 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \end{aligned} \quad (43)$$

and its inverse matrix

$$[RJ]_2^{-1} = \frac{1}{6}\begin{bmatrix} 2 & 2 \\ 2 & -1 \end{bmatrix} = \frac{1}{12}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 4 & 1 \\ 0 & 3 \end{bmatrix} \quad (44)$$

i.e, $[RJ]_4$ where

$$[RJ]_4 = \begin{bmatrix} 1 & 2 & 2 & 1 \\ 2 & -2 & 2 & -2 \\ 2 & 2 & -2 & -2 \\ 1 & -2 & -2 & 1 \end{bmatrix}$$

and its inverse matrix is (45)

$$[RJ]_4^{-1} = \frac{1}{8}\begin{bmatrix} 2 & 1 & 1 & 2 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 2 & -1 & -1 & 2 \end{bmatrix}.$$

*5) Case V:* In Case V, all elements are not equal; that is, $a \neq b \neq c$ and let $[RJ]_2$ be $\begin{bmatrix} 2 & 1 \\ 1 & -4 \end{bmatrix}$ and then $[RJ]_2^{-1}$ is $1/9\begin{bmatrix} 4 & 1 \\ 1 & -2 \end{bmatrix}$. This is the polygonal subsampling case of which its sampling ratio and subchannel is 9. It has nine coset vectors, i.e., $[RJ]_2$

$$\begin{aligned} M_5 = [RJ]_2 &= \begin{bmatrix} a & b \\ b & -c \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & -4 \end{bmatrix} \\ &= \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 3 & -3 \\ 1 & 5 \end{bmatrix} \\ &= \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & 9 \end{bmatrix}\begin{bmatrix} 1 & -4 \\ 0 & 1 \end{bmatrix} \end{aligned} \quad (46)$$

and its inverse matrix

$$[RJ]_2^{-1} = \frac{1}{9}\begin{bmatrix} 4 & 1 \\ 1 & -2 \end{bmatrix} = \frac{1}{18}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 5 & -1 \\ 3 & 3 \end{bmatrix}. \quad (47)$$

i.e., $[RJ]_4$ where

$$[RJ]_4 = \begin{bmatrix} 2 & 1 & 1 & 2 \\ 1 & -4 & 4 & -1 \\ 1 & 4 & -4 & -1 \\ 2 & -1 & -1 & 2 \end{bmatrix}$$

and its inverse matrix is

$$[RJ]_4^{-1} = \frac{1}{16}\begin{bmatrix} 2 & 4 & 4 & 2 \\ 4 & -1 & 1 & -4 \\ 4 & 1 & -1 & -4 \\ 2 & -4 & -4 & 2 \end{bmatrix}. \quad (48)$$

In this case, each matrix has three zones, which are ±1's, ±2's, and ±4's areas. The ±1's, ±2's, and ±4's areas of $[RJ]_4$ are able to be replaced by ±4's, ±2's, and ±1's areas of $[RJ]_4^{-1}$ respectively. The two positions of $[RJ]_4$ can be replaced by $[RJ]_4^{-1}$.

The algorithm for the fast RJT [FRJT] is similar fashion as [6]. The [FRJT] can be derived by decomposing $[RJ]_{2^k}$ into a product of sparse matrices given each rows/columns with only two nonzero elements. Fig. 1 shows a flow graph of the four-point Case II RJ transform algorithm. From this figure, it is clear that the first three iterations of the algorithm are those of the fast Hadamard transform. These are followed by the operation of the $[RC]_4$. The $N = 2^k$ point [FRJT] algorithm requires $kN + N/2$ real additions and $1.5N$ real multiplications in contrast with the $N$-point fast Hadamard transform which requires $kN$ real additions.

As an example of Case II, the fast RJ transform for $N = 8$, $[RJ]_8$ can be followed as (32)

$$\begin{aligned} [RJ]_8 &= \frac{1}{8}[H]_8[RC]_8 \\ &= \frac{1}{8}([H]_4 \otimes [H]_2)([RC]_4 \otimes 2[I]_2) \\ &= \frac{1}{8}([H]_4[RC]_4) \otimes 2[H]_2 = \frac{1}{2}([RJ]_4 \otimes 2[H]_2) \end{aligned} \quad (49)$$

In a similar manner as the forward Case II of the fast RJ transform, we note the inverse of $[RJ]_8$

$$\begin{aligned} [RJ]_8^{-1} &= ([H]_4 \otimes [H]_2)\left([RC]_4^{-1} \otimes \frac{1}{2}[I]_2\right) \\ &= ([H]_4[RC]_4^{-1}) \otimes \frac{1}{2}[H]_2 \\ &= [RJ]_4^{-1} \otimes \frac{1}{2}[H]_2. \end{aligned} \quad (50)$$

TABLE I

FORWARD MATRIX OF [RJ] FROM CASE I TO CASE V

| Case \ RJ | $[RJ]_2$ | $[RJ]_4$ |
|---|---|---|
| I<br>$a=b=c$ | $\begin{bmatrix}1&1\\1&-1\end{bmatrix}=\frac{1}{2}\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}2&0\\0&2\end{bmatrix}$ | $\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}=\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}\begin{bmatrix}4&0&0&0\\0&4&0&0\\0&0&4&0\\0&0&0&4\end{bmatrix}$ |
| II<br>$a=b\neq c$ | $\begin{bmatrix}1&1\\1&-2\end{bmatrix}=\frac{1}{2}\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}2&-1\\0&3\end{bmatrix}$ | $\begin{bmatrix}1&1&1&1\\1&-2&2&-1\\1&2&-2&-1\\1&-1&-1&1\end{bmatrix}=\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}\begin{bmatrix}4&0&0&0\\0&6&-2&0\\0&-2&6&0\\0&0&0&4\end{bmatrix}$ |
| III<br>$a=c\neq b$ | $\begin{bmatrix}1&2\\2&-1\end{bmatrix}=\frac{1}{2}\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}3&1\\-1&3\end{bmatrix}$ | $\begin{bmatrix}1&2&2&1\\2&-1&1&-2\\2&1&-1&-2\\1&-2&-2&1\end{bmatrix}=\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}\begin{bmatrix}6&0&0&-2\\0&6&2&0\\0&2&6&0\\-2&0&0&6\end{bmatrix}$ |
| IV<br>$a\neq b=c$ | $\begin{bmatrix}1&2\\2&-2\end{bmatrix}=\frac{1}{2}\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}3&0\\-1&4\end{bmatrix}$ | $\begin{bmatrix}1&2&2&1\\2&-2&2&-2\\2&2&-2&-2\\1&-2&-2&1\end{bmatrix}=\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}\begin{bmatrix}6&0&0&-2\\0&8&0&0\\0&0&8&0\\-2&0&0&6\end{bmatrix}$ |
| V<br>$a\neq b\neq c$ | $\begin{bmatrix}2&1\\1&-4\end{bmatrix}=\frac{1}{2}\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}3&-3\\1&5\end{bmatrix}$ | $\begin{bmatrix}2&1&1&2\\1&-4&4&-1\\1&4&-4&-1\\2&-1&-1&2\end{bmatrix}=\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}\begin{bmatrix}6&0&0&2\\0&10&-6&0\\0&-6&10&0\\2&0&0&6\end{bmatrix}$ |

In Tables I and II, [RJ] sparse matrices have a geometric structure such as

$$[RJ]_{2^k} = \frac{1}{2^k}[H]_{2^k}[RC]_{2^k} \tag{51}$$

$$[RC]_{2^k} = \begin{bmatrix} a+b^T & b+c \\ a-b^T & b-c \end{bmatrix}, \quad k \geq 2. \tag{52}$$

The forward or the inverse transform of the $N = 2^k$ point [RJ] sparse matrix requires $N^2 \log_2 N$ real additions; e.g., $[RJ]_2$ (*Case II*)

$$\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}1&1\\1&-2\end{bmatrix}=\begin{bmatrix}2&-1\\0&3\end{bmatrix} \tag{53}$$

and $[RJ]_4$ (*Case II*)

$$\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}\begin{bmatrix}1&1&1&1\\1&-2&2&-1\\1&2&-2&-1\\1&-1&-1&1\end{bmatrix}$$

$$=\begin{bmatrix}4&0&0&0\\0&6&-2&0\\0&-2&6&0\\0&0&0&4\end{bmatrix}. \tag{54}$$

In Tables I and II, *Case I–Case V* of the [RJ] matrix, with some examples, are shown. If we regard each basic sparse matrix of $[RJ]_2$, then we have found some regular extending geometric structure. Therefore, we have a blockwise circulant sparse matrix that looks like football sphere as follows.

- forward sparse matrix:

$$\begin{bmatrix}a&b\\c&d\end{bmatrix} \Rightarrow \begin{bmatrix}2a&0&0&2c\\0&2d&2b&0\\0&2b&2d&0\\2c&0&0&2a\end{bmatrix} \tag{55}$$

- inverse sparse matrix:

$$\begin{bmatrix}a&b\\c&d\end{bmatrix}^{-1} \Rightarrow \begin{bmatrix}d&0&0&-b\\0&a&-c&0\\0&-c&a&0\\-b&0&0&d\end{bmatrix}. \tag{56}$$

TABLE II

INVERSE MATRIX OF [RJ] FROM CASE I TO CASE V

| Case \ RJ | $[RJ]_2^{-1}$ | $[RJ]_4^{-1}$ |
|---|---|---|
| I | $\frac{1}{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} = \frac{1}{2}\bullet\frac{1}{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix}$ | $\frac{1}{4}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix} = \frac{1}{16}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}4 & 0 & 0 & 0\\ 0 & 4 & 0 & 0\\ 0 & 0 & 4 & 0\\ 0 & 0 & 0 & 4\end{bmatrix}$ |
| II | $\frac{1}{3}\begin{bmatrix}2 & 1\\ 1 & -1\end{bmatrix} = \frac{1}{2}\bullet\frac{1}{3}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}3 & 0\\ 1 & 2\end{bmatrix}$ | $\frac{1}{8}\begin{bmatrix}2 & 2 & 2 & 2\\ 2 & -1 & 1 & -2\\ 2 & 1 & -1 & -2\\ 2 & -2 & -2 & 2\end{bmatrix} = \frac{1}{16}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}4 & 0 & 0 & 0\\ 0 & 3 & 1 & 0\\ 0 & 1 & 3 & 0\\ 0 & 0 & 0 & 4\end{bmatrix}$ |
| III | $\frac{1}{5}\begin{bmatrix}1 & 2\\ 2 & -1\end{bmatrix} = \frac{1}{2}\bullet\frac{1}{5}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}3 & 1\\ -1 & 3\end{bmatrix}$ | $\frac{1}{8}\begin{bmatrix}2 & 1 & 1 & 2\\ 1 & -2 & 2 & -1\\ 1 & 2 & -2 & -1\\ 2 & -1 & -1 & 2\end{bmatrix} = \frac{1}{16}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}3 & 0 & 0 & 1\\ 0 & 3 & -1 & 0\\ 0 & -1 & 3 & 0\\ 1 & 0 & 0 & 3\end{bmatrix}$ |
| IV | $\frac{1}{6}\begin{bmatrix}2 & 2\\ 2 & -1\end{bmatrix} = \frac{1}{2}\bullet\frac{1}{6}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}4 & 1\\ 0 & 3\end{bmatrix}$ | $\frac{1}{8}\begin{bmatrix}2 & 1 & 1 & 2\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 2 & -1 & -1 & 2\end{bmatrix} = \frac{1}{16}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}3 & 0 & 0 & 1\\ 0 & 2 & 0 & 0\\ 0 & 0 & 2 & 0\\ 1 & 0 & 0 & 3\end{bmatrix}$ |
| V | $\frac{1}{9}\begin{bmatrix}4 & 1\\ 1 & -2\end{bmatrix} = \frac{1}{2}\bullet\frac{1}{9}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}5 & -1\\ 3 & 3\end{bmatrix}$ | $\frac{1}{8}\begin{bmatrix}2 & 4 & 4 & 2\\ 4 & -1 & 1 & -4\\ 4 & 1 & -1 & -4\\ 2 & -4 & -4 & 2\end{bmatrix} = \frac{1}{16}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}6 & 0 & 0 & -2\\ 0 & 5 & 3 & 0\\ 0 & 3 & 5 & 0\\ -2 & 0 & 0 & 6\end{bmatrix}$ |

Comparing the forward sparse matrix with the inverse sparse matrix, the matrix elements are changed as follows:

$$a \Leftrightarrow d, \quad b \Leftrightarrow c.$$

Fig. 2(a) and (b) shows the expanding blockwise circulant sparse matrix structure as in Tables I and II. This figure is a plane surface, and an interesting point is that the blockwise circulant sparse matrix is characterized in similiar fashion as the football and rotating pattern. This means that when the $2 \times 2$ sparse matrix is expanded to a $4 \times 4$ matrix, the element of the $2 \times 2$ sparse matrix becomes as shown in the pattern of Fig. 2. As the example of the *Case I* and *Case II* of $[RJ]_2$.

- [RC] sparse matrix of *Case I*:

$$[RJ]_2 \Rightarrow \begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix} \Rightarrow \begin{bmatrix}4 & 0 & 0 & 0\\ 0 & 4 & 0 & 0\\ 0 & 0 & 4 & 0\\ 0 & 0 & 0 & 4\end{bmatrix}$$

and its inverse of $[RJ]_2$ is

$$[RJ]_2^{-1} \Rightarrow \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \Rightarrow \begin{bmatrix}\frac{1}{2} & 0 & 0 & 0\\ 0 & \frac{1}{2} & 0 & 0\\ 0 & 0 & \frac{1}{2} & 0\\ 0 & 0 & 0 & \frac{1}{2}\end{bmatrix}. \qquad (57)$$

- [RC] sparse matrix of *Case II*:

$$[RJ]_2 \Rightarrow \begin{bmatrix}2 & -1\\ 0 & 3\end{bmatrix} \Rightarrow \begin{bmatrix}4 & 0 & 0 & 0\\ 0 & 6 & -2 & 0\\ 0 & -2 & 6 & 0\\ 0 & 0 & 0 & 4\end{bmatrix}$$

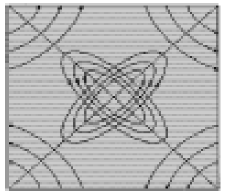
and its inverse of $[RJ]_2$ is

$$[RJ]_2^{-1} \Rightarrow \begin{bmatrix}3 & 0\\ 1 & 2\end{bmatrix} \Rightarrow \begin{bmatrix}4 & 0 & 0 & 0\\ 0 & 3 & 1 & 0\\ 0 & 1 & 3 & 0\\ 0 & 0 & 0 & 4\end{bmatrix}. \qquad (58)$$

## V. CONCLUSION

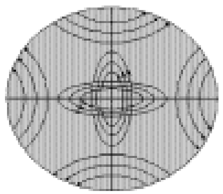
The [RJT] was introduced in this paper. The RJ matrix is a generalized form of the Weighted Hadamard and Hadamard.

a ⇔ d



(a)

b ⇔ c



(b)

Fig. 2. To illustrate general concept of sparse matrix [RJ] pattern. (a) Blockwise circulant sparse matrix pattern. (b) Sphere circulant sparse matrix, like football.

The $[RJ]_N$ matrix has a recursive structure and symmetric characteristics. The elements' positions in the forward matrix can be replaced by its inverse matrix, and the signs are not changed between the matrix and its inverse. The $[RJ]_N$ matrix has five cases of basic symmetric matrix according to the construction of elements. The Hadamard matrix is a special case of the RJ matrix. The fast $[RJ]_N$ transform algorithm is the matrix decomposition of the Hadamard matrices and succesively lower order $[RC]_N$. A blockwise $[RC]_N$ circulant sparse matrix of $[RJ]_N$ leads to very clear decomposition. This method is presented for its simplicity and clarity, decomposing an $[RJ]_N$ matrix in terms of the sparse matrices.

## Appendix I

The proof of (25) is as follow.

Let $A$ be a complex matrix, partitioned as $a : n_1 \times n_1, b : n_1 \times n_2, b^T : n_2 \times n_1, c : n_2 \times n_2$.

i) If $\det A_{11} \neq 0$, then

$$\det \begin{pmatrix} a & b \\ b^T & c \end{pmatrix} = \det a \det(c - b^T a^{-1} b). \tag{59}$$

ii) If $\det c \neq 0$, then

$$\det \begin{pmatrix} a & b \\ b^T & c \end{pmatrix} = \det c \det(a - bc^{-1} b^T). \tag{60}$$

For (59), first note that

$$\det \begin{pmatrix} I_{n_1} & -a^{-1}b \\ 0 & I_{n_2} \end{pmatrix} = 1$$

by

$$\det \begin{pmatrix} a & 0 \\ b^T & c \end{pmatrix} = \det \begin{pmatrix} a & b \\ 0 & c \end{pmatrix} = \det a \det c.$$

Therefore, by $\det(ab) = \det a \det b$

$$\det \begin{pmatrix} a & b \\ b^T & c \end{pmatrix} = \det \begin{pmatrix} a & b \\ b^T & c \end{pmatrix} \begin{pmatrix} I_{n_1} & -a^{-1}b \\ 0 & I_{n_2} \end{pmatrix}$$
$$= \det \begin{pmatrix} a & 0 \\ b^T & c - b^T a^{-1} b \end{pmatrix}$$
$$= \det a \det(c - b^T a^{-1} b).$$

The proof of (60) is similar to (59).

Below, we illustrate this relationship for the *Case of* RJ-II.

$$\begin{bmatrix} 1 & 1 \\ 1 & -2 \end{bmatrix} = \det(1)\det(-2 - 1 \cdot 1 \cdot 1) = -3. \tag{61}$$

Clearly, (61) is the determinant.

## References


[1] N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing*. Berlin, Germany: Springer-Verlag, 1975.

[2] M. H. Lee and M. Kaveh, "Fast Hadamard transform based on a simple matrix factorization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-6, pp. 1666–1667, 1986.

[3] M. H. Lee and Y. Yasuda, "Simple systolic array for hadamard transform," *Electron. Lett.*, vol. 26, no. 18, pp. 1478–1480, Aug. 30, 1990.

[4] M. H. Lee, "High speed multidimensional systolic arrays for discrete Fourier transform," *IEEE Trans. Circuits Syst. II*, vol. 39, pp. 876–879, Dec. 1992.

[5] M. H. Lee and D. Y. Kim, "Weighted Hadamard transform for S/N ratio enhancement in image transform," *IEEE ISCAS'84*, pp. 65–68.

[6] M. H. Lee, "The center weighted Hadamard transform," *IEEE Trans. Circuits Syst.*, vol. CAS-36, pp. 1247–1249, Sept. 1989.

[7] M. H. Lee, J. Y. Park, M. W. Kwon, and S.-R. Lee, "The Reverse Jacket matrix of weighted Hadamard transform for multidimensional signal processing," in *Proc. PIMRC'96*, Taipei, Taiwan, R.O.C., pp. 482–486.

[8] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[9] M. H. Lee and S. R. Lee, "On the Reverse Jacket matrix for weighted Hadamard transform," *IEEE Trans. Circuits Syst. II*, vol. 45, pp. 436–441, Mar. 1998.

[10] C.-P. Fan and J.-F. Yang, "Fast center weighted Hadamard transform algorithm," *IEEE Trans. Circuits Syst. II*, vol. 45, pp. 429–432, Mar. 1998.

[11] K. W. Schmidt and E. T. H. Wang, "The weighted of Hadamard matrices," *J. Combinatorial Theory*, ser. A 23, pp. 257–263, 1977.

[12] E. T. H. Brown, E. T. H. Sharpe, E. T. H. Hughes, and E. T. H. Post, *Lines, Waves and Antenna*. New York: Wiley, 1973.

[13] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, MA: Cambridge Univ. Press, 1991.

[14] A. Jennings and J. J. Mckeown, *Matrix Computation*. New York: Wiley, 1992.

[15] M. H. Lee, "The design of multidimensional filter bank using Reverse Jacket matrix," presented at the Proc. TenCon'99, Cheju, Korea.

[16] M. H. Lee, *Information and Coding Theory*. Korea: Bobdu, 1998.

[17] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[18] M. L. Eaton, *Multivariate Statistics*. New York: Wiley, 1983.

**Moon Ho Lee** (S'81–M'85–SM'86) received the B.S. and M.S. degrees in electrical engineering from the Chonbuk National University of Korea in 1967 and 1976, respectively. He received the Ph.D. degree in electronics engineering from the Chonnam National University in 1984 and the University of Tokyo, Japan, in 1990.

From 1970 to 1980, he was a Chief Engineer with Namyang Moonhwa Broadcasting. Since 1980, he has been a Professor with the Department of Information and Communication and a Director with the Institute of Information and Communication, both at Chonbuk National University. From August 1985 to August 1986, he was also with the University of Minnesota, as a Post Doctorate. He has held visiting positions with the University of Hannover, Germany, during 1990, the University of Aachen, Germany, during the winters of 1992 and 1996, and the University of Munchen, Germany, during the summer of 1998. He has authored 13 books, including *Digital Communictions* (Korea: Youngil, 1999), *Information and Coding* (Korea: Bokdu, 1998),
*Digital Image Processing*, (Korea: Daeyoung, 1994), and *Digital Filter Design* (Korea: Daeyoung, 1995). His research interests include multidimensional source and channel coding, mobile communication, and image processing. He is a registered Telecommunication Professional Engineer and a member of the National Academy of Engineering in Korea.

Dr. Lee was the recipient of paper prize awards from the Korean Institute of Communication Science in 1986 and 1987, the Korean Institute of Electronics Engineers in 1987, and Chonbuk Province in 1992.

# Jacket Matrices(Forward)

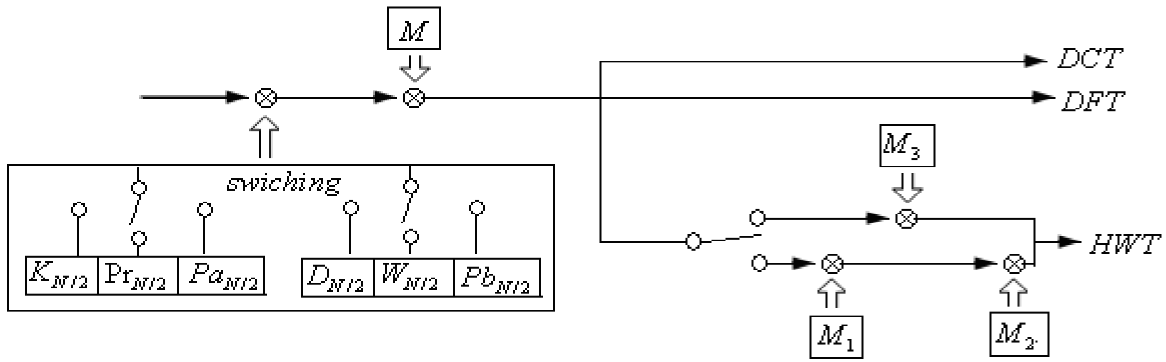
| Jacket | case | $[J]_1$ | $[J]_2$ |
|---|---|---|---|
| 1 | $a=b=c=1$ | $\begin{bmatrix}1&1\\1&-1\end{bmatrix}=\frac{1}{2}\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}2&0\\0&2\end{bmatrix}$ | $\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}=\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}\begin{bmatrix}4&0&0&0\\0&4&0&0\\0&0&4&0\\0&0&0&4\end{bmatrix}$ |
| 2 | $a=b\neq c$ <br> $a=b=1, c=w$ | (a) $w=real=2$ <br> $\begin{bmatrix}1&1\\1&-2\end{bmatrix}=\frac{1}{2}\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}2&-1\\0&3\end{bmatrix}$ | $\begin{bmatrix}1&1&1&1\\1&-2&2&-1\\1&2&-2&-1\\1&-1&-1&1\end{bmatrix}=\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}\begin{bmatrix}4&0&0&0\\0&6&-2&0\\0&-2&6&0\\0&0&0&4\end{bmatrix}$ |
| | | (b) $w=imaginary=j$ <br> $\begin{bmatrix}1&1\\1&-j\end{bmatrix}=\frac{1}{2}\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}2&1-j\\0&1+j\end{bmatrix}$ | $\begin{bmatrix}1&1&1&1\\1&-j&j&-1\\1&j&-j&-1\\1&-1&1&-1\end{bmatrix}=\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&\frac{1}{2}(1+j)&\frac{1}{2}(1-j)&0\\0&\frac{1}{2}(1-j)&\frac{1}{2}(1+j)&0\\0&0&0&1\end{bmatrix}$ |
| 3 | $a=c\neq b$ <br> $a=c=1, b=2$ | $\begin{bmatrix}1&2\\2&-1\end{bmatrix}=\frac{1}{2}\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}3&1\\-1&3\end{bmatrix}$ | $\begin{bmatrix}1&2&2&1\\2&-1&1&-2\\2&1&-1&-2\\1&-2&-2&1\end{bmatrix}=\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}\begin{bmatrix}6&0&0&-2\\0&6&2&0\\0&2&6&0\\-2&0&0&6\end{bmatrix}$ |
| 4 | $a\neq b=c$ <br> $a=1, b=c=2$ | $\begin{bmatrix}1&2\\2&-2\end{bmatrix}=\frac{1}{2}\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}3&0\\-1&4\end{bmatrix}$ | $\begin{bmatrix}1&2&2&1\\2&-2&2&-2\\2&2&-2&-2\\1&-2&-2&1\end{bmatrix}=\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}\begin{bmatrix}6&0&0&-2\\0&8&0&0\\0&0&8&0\\-2&0&0&6\end{bmatrix}$ |
| 5 | $a\neq b\neq c$ <br> $a=2, b=1, c=4$ | $\begin{bmatrix}2&1\\1&-4\end{bmatrix}=\frac{1}{2}\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}3&-3\\1&5\end{bmatrix}$ | $\begin{bmatrix}2&1&1&2\\1&-4&4&-1\\1&4&-4&-1\\2&-1&-1&2\end{bmatrix}=\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}\begin{bmatrix}6&0&0&-2\\0&10&-6&0\\0&-6&10&0\\-2&0&0&6\end{bmatrix}$ |

** Moon Ho Lee, "A New Reverse Jacket Transform and Its Fast Algorithm," IEEE Trans. On Circuit and System 2, vol. 47, no. 1, Jan. 2000. pp. 39-47*

# Jacket Matrices(Inverse)

| Jacket case | | $[J]_1^{-1}$ | $[J]_2^{-1}$ |
|---|---|---|---|
| 1 | $a=b$ $=c=1$ | $\frac{1}{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}=\frac{1}{2}\cdot\frac{1}{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix}$ | $\frac{1}{4}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}=\frac{1}{16}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}4 & 0 & 0 & 0\\ 0 & 4 & 0 & 0\\ 0 & 0 & 4 & 0\\ 0 & 0 & 0 & 4\end{bmatrix}$ |
| 2 | $a=b\neq c$ $a=b=1,$ $c=w$ | (a) $w=real=2$ $\frac{1}{2}\begin{bmatrix}2 & 1\\ 1 & -1\end{bmatrix}=\frac{1}{2}\cdot\frac{1}{3}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}3 & 0\\ 1 & 2\end{bmatrix}$ | $\frac{1}{8}\begin{bmatrix}2 & 2 & 2 & 2\\ 2 & -1 & 1 & -2\\ 2 & 1 & -1 & -2\\ 2 & -2 & -2 & 2\end{bmatrix}=\frac{1}{16}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}4 & 0 & 0 & 0\\ 0 & 3 & 1 & 0\\ 0 & 1 & 3 & 0\\ 0 & 0 & 0 & 4\end{bmatrix}$ |
| | | (b) $w=imaginary=j$ $\frac{\begin{bmatrix}j & 1\\ 1 & -1\end{bmatrix}}{j+1}=\frac{\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}j+1 & 0\\ j-1 & 2\end{bmatrix}}{2(j+1)}$ | $\frac{\begin{bmatrix}j & j & j & j\\ j & -1 & 1 & -j\\ j & 1 & -1 & -j\\ j & -j & -j & j\end{bmatrix}}{4j}=\frac{\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}4j & 0 & 0 & 0\\ 0 & 2j+2 & 2j-2 & 0\\ 0 & 2j-2 & 2j+2 & 0\\ 0 & 0 & 0 & 4j\end{bmatrix}}{16j}$ |
| 3 | $a=c\neq b$ $a=c=1,$ $b=2$ | $\frac{1}{5}\begin{bmatrix}1 & 2\\ 2 & -1\end{bmatrix}=\frac{1}{2}\cdot\frac{1}{5}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}3 & 1\\ -1 & 3\end{bmatrix}$ | $\frac{1}{8}\begin{bmatrix}2 & 1 & 1 & 2\\ 1 & -2 & 2 & -1\\ 1 & 2 & -2 & -1\\ 2 & -1 & -1 & 2\end{bmatrix}=\frac{1}{16}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}3 & 0 & 0 & 1\\ 0 & 3 & -1 & 0\\ 0 & -1 & 3 & 0\\ 1 & 0 & 0 & 3\end{bmatrix}$ |
| 4 | $a\neq b=c$ $a=1,$ $b=c=2$ | $\frac{1}{6}\begin{bmatrix}2 & 2\\ 2 & -1\end{bmatrix}=\frac{1}{2}\cdot\frac{1}{6}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}4 & 1\\ 0 & 3\end{bmatrix}$ | $\frac{1}{8}\begin{bmatrix}2 & 1 & 1 & 2\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 2 & -1 & -1 & 2\end{bmatrix}=\frac{1}{16}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}3 & 0 & 0 & 1\\ 0 & 2 & 0 & 0\\ 0 & 0 & 2 & 0\\ 1 & 0 & 0 & 3\end{bmatrix}$ |
| 5 | $a\neq b\neq c$ $a=2,$ $b=1, c=4$ | $\frac{1}{9}\begin{bmatrix}4 & 1\\ 1 & -2\end{bmatrix}=\frac{1}{2}\cdot\frac{1}{9}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}5 & -1\\ 3 & 3\end{bmatrix}$ | $\frac{1}{8}\begin{bmatrix}2 & 4 & 4 & 2\\ 4 & -1 & 1 & -4\\ 4 & 1 & -1 & -4\\ 2 & -4 & -4 & 2\end{bmatrix}=\frac{1}{16}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}6 & 0 & 0 & -2\\ 0 & 5 & 3 & 0\\ 0 & 3 & 5 & 0\\ -2 & 0 & 0 & 6\end{bmatrix}$ |

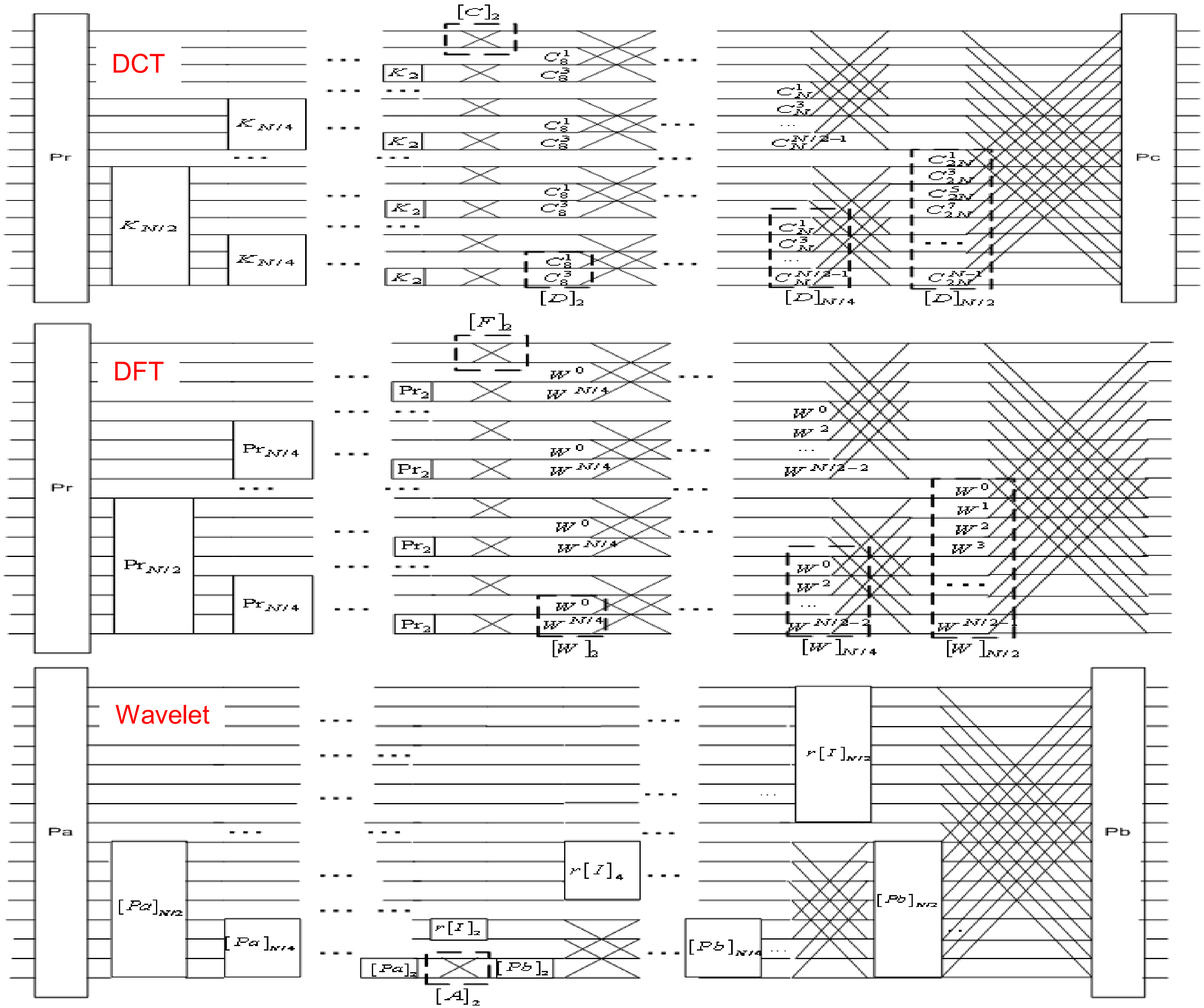
# DFT /DCT/ Wavelet




| | DFT, Fourier,1822 | DCT-II, K.R.Rao, 1974 | Wavelet, G.Strang,1996 |
|---|---|---|---|
| F O R M | $X(n)=\sum_{m=0}^{N-1} x(m)W^{nm}, \quad 0\le n\le N-1$<br>$[\tilde{F}]_4=[\mathrm{Pr}]_4[F]_4 = \left(\begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix}\begin{bmatrix} F_2 & 0 \\ 0 & E_2 \end{bmatrix}\right)^T$<br>$[E]_2=\begin{bmatrix} 1 & -j \\ 1 & j \end{bmatrix}$<br>* $([E]_2)^{-1}=\left(\begin{bmatrix} 1/1 & -1/j \\ 1/1 & 1/j \end{bmatrix}\right)^T=\begin{bmatrix} 1 & 1 \\ j & -j \end{bmatrix}$<br>$[\tilde{F}]_N == \begin{bmatrix} I_{\frac{N}{2}} & 0 \\ 0 & \mathrm{Pr}_{\frac{N}{2}} \end{bmatrix}\begin{bmatrix} \tilde{F}_{\frac{N}{2}} & 0 \\ 0 & \tilde{F}_{\frac{N}{2}} \end{bmatrix}\begin{bmatrix} I_{\frac{N}{2}} & 0 \\ 0 & W_{\frac{N}{2}} \end{bmatrix}\begin{bmatrix} I_{\frac{N}{2}} & I_{\frac{N}{2}} \\ I_{\frac{N}{2}} & -I_{\frac{N}{2}} \end{bmatrix}$<br>$[F]_N=[P_r]_N^{-1}[\tilde{F}]_N$ | $[C_N]_{m,n}=\sqrt{\frac{2}{N}}k_m \cos\frac{m(n+\frac{1}{2})\pi}{N}, \quad m,n=0,1,\ldots,N-1$<br>where $k_j=\begin{cases} 1, & j=1,2,\ldots,N-1 \\ \frac{1}{\sqrt{2}}, & j=0,N \end{cases}$<br>$[C]_2=\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_4^1 & C_4^3 \end{bmatrix}=\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$<br>* $[C]_2^{-1}=\frac{1}{2}\begin{bmatrix} \sqrt{2} & \sqrt{2} \\ \sqrt{2} & -\sqrt{2} \end{bmatrix}$<br>$[\tilde{C}]_N=\begin{bmatrix} I_{N/2} & 0 \\ 0 & K_{N/2} \end{bmatrix}\begin{bmatrix} C_{N/2} & 0 \\ 0 & C_{N/2} \end{bmatrix}\begin{bmatrix} I_{N/2} & 0 \\ 0 & D_{N/2} \end{bmatrix}\begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}$<br>$[C]_N=[P_r]_N^{-1}[\tilde{C}]_N[P_c]_N^{-1}$ | $[A]_2=\begin{bmatrix} r & r \\ r & -r \end{bmatrix}$<br>* $[A]_2^{-1}=\frac{1}{2}\begin{bmatrix} 1/r & 1/r \\ 1/r & -1/r \end{bmatrix}^T$<br>$[\tilde{A}]_4=[Pi]_4[A]_4[Pj]_4$<br>$=r\left(\begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix}\begin{bmatrix} I_2 & 0 \\ 0 & A_2 \end{bmatrix}\right)^T$<br>$[\tilde{A}]_N=r\begin{bmatrix} I_{N/2} & 0 \\ 0 & Pi_{N/2}^{-1} \end{bmatrix}\begin{bmatrix} I_{N/2} & 0 \\ 0 & \tilde{A}_{N/2} \end{bmatrix}\begin{bmatrix} I_{N/2} & 0 \\ 0 & Pj_{N/2}^{-1} \end{bmatrix}\begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}$<br>$[A]_N=[Pi]_N^{-1}[\tilde{A}]_N[Pj]_N^{-1}$ |

Common Form: $[\tilde{X}]_N = r\begin{bmatrix} I_{N/2} & 0 \\ 0 & Pi_{N/2} \end{bmatrix}\begin{bmatrix} \tilde{X}_{N/2}(or I_{N/2}) & 0 \\ 0 & \tilde{X}_{N/2} \end{bmatrix}\begin{bmatrix} I_{N/2} & 0 \\ 0 & Pj_{N/2} \end{bmatrix}\begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}$

* Jacket Matrices Based on Decomposition


DCT
DFT
Wavelet
Pr
Pc
Pa
Pb

**DCT Matrix:**

The DCT of a data sequence $X(m)$, $m = 0,1,\ldots,N-1$ is defined as

$$L_x(0) = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} X(m),$$

and

$$L_x(k) = \sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} X(m) \cos\frac{(2m+1)k\pi}{2N}, \quad k = 1,2,\ldots,N-1, \qquad (1)$$

where $L_x(k)$ is the $k$ th DCT coefficient.

It is worthwhile to note that the set of basis vector elements

$$\{\frac{1}{\sqrt{N}}, \sqrt{\frac{2}{N}} \cos\frac{(2m+1)k\pi}{2N}\}$$

is actually a class of discrete Chebyshev polynomials. This can be easily seen by examining the following definition of Chebyshev polynomials

$$T_0(p) = \frac{1}{\sqrt{N}}$$

and

$$T_k(Z_m) = \sqrt{\frac{2}{N}} \cos[k \cos^{-1}(Z_m)], \quad k, m = 1,2,\ldots,N-1, \qquad (2)$$

where $T_k(Z_m)$ is the $k$ -th Chebyshev polynomial.

Now, the zeros of the $N$ -th polynomial $T_N(Z_m)$ are given by

$$Z_m = \cos\frac{(2m+1)\pi}{2N}, \quad m = 0,1,\ldots,N-1. \tag{3}$$

Substituting (A-15) in (A-14), we evaluate $\{T_l(Z_m)\}$, $l = 0,1,\ldots,N-1$ at the zeros of $T_N(Z_m)$. This results in the set of Chebyshev polynomials

$$T_0(m) = \frac{1}{\sqrt{N}}$$

and

$$\begin{aligned} T_k(m) &= \sqrt{\frac{2}{N}}\cos[k\cos^{-1}(Z_m)] \\ &= \sqrt{\frac{2}{N}}\cos[k\cos^{-1}(\cos\frac{(2m+1)\pi}{2N})], \quad k, m = 1,2,\ldots,N-1, \\ &= \sqrt{\frac{2}{N}}\cos[\frac{(2m+1)k\pi}{2N}] \end{aligned} \tag{4}$$

which are equivalent to the basis set of the DCT.

Again, the inverse discrete cosine transform (IDCT) is defined as

$$X(m) = \frac{1}{\sqrt{N}}L_x(0) + \sqrt{\frac{2}{N}}\sum_{k=1}^{N-1}L_x(k)\cos\frac{(2m+1)k\pi}{2N}, \quad m = 0,1,..,N-1. \tag{5}$$

It can be shown that application of the orthogonal property

$$\sum_{m=0}^{N-1}T_p(m)T_q(m) = \begin{cases}1, p = q \\ 0, p \neq q\end{cases}, \tag{6}$$

Computational Considerations: It can be shown that the DCT can be equivalently expressed as

$$L_x(0) = \frac{1}{\sqrt{N}}\sum_{m=0}^{N-1}X(m)$$

and

$$L_x(k) = \sqrt{\frac{2}{N}}\,\mathrm{Re}\left\{e^{-i\frac{k\pi}{2N}}\sum_{m=0}^{2N-1}X(m)W^{km}\right\}, \quad k = 1,2,\ldots,N-1, \tag{7}$$

where $W = e^{-i2\pi/2N}$, $i = \sqrt{-1}$, $X(m) = 0$, when $m = N, N+1,\ldots,2N-1$, and $\mathrm{Re}(.)$ denotes the real part of the term enclosed.

Proof of (A-19): Since we have

$$e^{-i\frac{k\pi}{2N}}\sum_{m=0}^{2N-1}X(m)W^{km} = e^{-i\frac{k\pi}{2N}}\left(\sum_{m=0}^{2N-1}X(m)e^{-i2km\pi/2N}\right) = \sum_{m=0}^{2N-1}X(m)e^{-i(2m+1)k\pi/2N}$$

$$= \left(\sum_{m=0}^{2N-1}X(m)\left(\cos\frac{(2m+1)k\pi}{2N} - i\sin\frac{(2m+1)k\pi}{2N}\right)\right),$$

then

$$\begin{aligned}&\mathrm{Re}\left\{e^{-i\frac{k\pi}{2N}}\sum_{m=0}^{2N-1}X(m)W^{km}\right\}\\&=\mathrm{Re}\left\{\left(\sum_{m=0}^{2N-1}X(m)\left(\cos\frac{(2m+1)k\pi}{2N}-i\sin\frac{(2m+1)k\pi}{2N}\right)\right)\right\},\\&=\sum_{m=0}^{2N-1}X(m)\left(\cos\frac{(2m+1)k\pi}{2N}\right)\end{aligned} \tag{8}$$

and $X(m)=0$, when $m=N,N+1,\ldots,2N-1$, thus the (A-20) can be written as

$$\begin{aligned}&\mathrm{Re}\left\{e^{-i\frac{k\pi}{2N}}\sum_{m=0}^{2N-1}X(m)W^{km}\right\}=\sum_{m=0}^{2N-1}X(m)\left(\cos\frac{(2m+1)k\pi}{2N}\right)\\&=\sum_{m=0}^{N-1}X(m)\left(\cos\frac{(2m+1)k\pi}{2N}\right)\end{aligned}. \tag{9}$$

The result is DCT coefficient.

For example, a 4-by-4 Fourier matrix is as

$$F=\begin{bmatrix}1&1&1&1\\1&i&-1&-i\\1&-1&1&-1\\1&-i&-1&i\end{bmatrix}, \tag{10}$$

and we use the part of (A-22), as

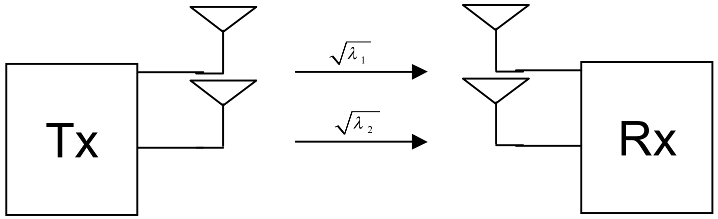
$$F'=\begin{bmatrix}1&1\\1&i\end{bmatrix},$$

By using the (A-14), we can evaluate 2-by-2 DCT matrix as

$$\begin{aligned}C&=\mathrm{Re}\left\{\begin{bmatrix}\frac{1}{\sqrt{N}}&0\\0&e^{-i\frac{k\pi}{2N}}\end{bmatrix}F'\right\}_{N=2,k=1}=\mathrm{Re}\left\{\begin{bmatrix}\frac{1}{\sqrt{2}}&0\\0&e^{-i\frac{\pi}{4}}\end{bmatrix}\begin{bmatrix}1&1\\1&i\end{bmatrix}\right\}\\&=\mathrm{Re}\left\{\begin{bmatrix}\frac{1}{\sqrt{2}}&\frac{1}{\sqrt{2}}\\e^{-i\frac{\pi}{4}}&\left(e^{-i\frac{\pi}{4}}\right)i\end{bmatrix}\right\}=\mathrm{Re}\left\{\begin{bmatrix}\frac{1}{\sqrt{2}}&\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}}i&\left(\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}}i\right)i\end{bmatrix}\right\}\\&=\mathrm{Re}\left\{\begin{bmatrix}\frac{1}{\sqrt{2}}&\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}}i&\left(-\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}}i\right)\end{bmatrix}\right\}=\begin{bmatrix}\frac{1}{\sqrt{2}}&\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}&-\frac{1}{\sqrt{2}}\end{bmatrix}.\end{aligned}$$

Thus we obtain the 2-by-2 DCT matrix.

# MIMO Channel Singular Value Decomposition




Eigenvalue Decomposition(EVD), MIMO SVD

| | **Fibonacci** (1175-1250)<br>Italy<br>Based on Sequence: $0,1,1,2,3,5,8,13,\cdots$ | **Markov** (1856-1922)<br>Russia<br>Based on Probability | **Jacket Moon Ho Lee**<br>(2000,2006) Korea<br>Kronecker High dimension |
|---|---|---|---|
| **FORM** | $A = S \wedge S^{-1}$ $Eig(A_2) = \lambda_{1,2}$<br>$F_{k+2} = F_{k+1} + F_K$ $\underline{A^k = S \wedge^k S^{-1}} = S\begin{bmatrix} \lambda_1^k & 0 \\ 0 & \lambda_2^k \end{bmatrix} S^{-1}$<br>$u_k = \begin{bmatrix} F_{k+1} \\ F_K \end{bmatrix}$ $A_2 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$<br>$u_{k+1} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} u_k \Leftrightarrow \begin{cases} F_{k+2} = F_{k+1} + F_K \\ F_{k+1} = F_{k+1} \end{cases}$<br>$u_k = A_2^k u_0 = S \wedge^k S^{-1} u_0$<br>$= \begin{bmatrix} \lambda_1 & \lambda_2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1^k & 0 \\ 0 & \lambda_2^k \end{bmatrix} \begin{bmatrix} 1 & -\lambda_2 \\ -1 & \lambda_1 \end{bmatrix} \frac{1}{\lambda_1 - \lambda_2} u_0$<br>* A is a fixed matrix: $A_2 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ | $A = S \wedge S^{-1}$ $Eig(A_2) = \lambda_{1,2}$<br>$A_2 = \begin{bmatrix} 0.9 & 0.2 \\ 0.1 & 0.8 \end{bmatrix}$, $\underline{A^k = S \wedge^k S^{-1}} = S\begin{bmatrix} \lambda_1^k & 0 \\ 0 & \lambda_2^k \end{bmatrix} S^{-1}$<br>$\begin{bmatrix} y_1 \\ z_1 \end{bmatrix} = A\begin{bmatrix} y_0 \\ z_0 \end{bmatrix}$, $\begin{bmatrix} y_k \\ z_k \end{bmatrix} = A^k \begin{bmatrix} y_0 \\ z_0 \end{bmatrix} = S \wedge^k S^{-1} \begin{bmatrix} y_0 \\ z_0 \end{bmatrix}$<br>$\lambda_1 = 1$ $\lambda_2 = 0.7$<br>$\begin{bmatrix} y_k \\ z_k \end{bmatrix} = A^k \begin{bmatrix} y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} 2/3 & 1/3 \\ 1/3 & -1/3 \end{bmatrix} \begin{bmatrix} 1^k & 0 \\ 0 & 0.7^k \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} y_0 \\ z_0 \end{bmatrix}$<br>* $A_2 = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ $a,c,b,d \geq 0$<br>$a + c = 1$ $b + d = 1$ | $A = S \wedge S^{-1}$ $Eig(A_2) = \lambda_{1,2}$<br>$\underline{A^k = S \wedge^k S^{-1}} = S\begin{bmatrix} \lambda_1^k & 0 \\ 0 & \lambda_2^k \end{bmatrix} S^{-1}$<br>If $A_2 = \begin{bmatrix} x_1 & x_2 \\ x_3 & x_1 \end{bmatrix}$, $S_2^{-1} = J_2 = \begin{bmatrix} 1 & -w \\ 1 & w \end{bmatrix} or \begin{bmatrix} a & \sqrt{ac} \\ \sqrt{ac} & -c \end{bmatrix}$<br>$w = \sqrt{\frac{x_2}{x_3}} or \frac{c}{a} = \frac{x_2}{x_3}$ $A^k = J^{-1} \wedge^k J$<br>$A_{2^n} = A_{2^{n-1}} \otimes A_2$ $J_{2^n} = J_{2^{n-1}} \otimes J_2$ $A_2 = \begin{bmatrix} 3 & 8 \\ 2 & 3 \end{bmatrix}$<br>$A_2^k = J_2^{-1} \wedge^k J_2 = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} (-1)^k & 0 \\ 0 & 7^k \end{bmatrix} \begin{bmatrix} 1 & -2 \\ 1 & 2 \end{bmatrix}$<br>** $A_2 = \begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix}$ $x_1 = x_4$<br>$det(A) \neq 0$ |

** http://en.wikipedia.org/wiki/Jacket matrix, http://en.wikipedia.org/wiki/Category:Matrices, http://en.wikipedia.org/wiki/user:leejacket

# The Applying of Jacket Matrices in CDMA

A general 4*4 Jacket matrix is:

$$[J]_4 = \begin{bmatrix} a & b & b & a \\ b & -c & c & -b \\ b & c & -c & -b \\ a & -b & -b & a \end{bmatrix}$$

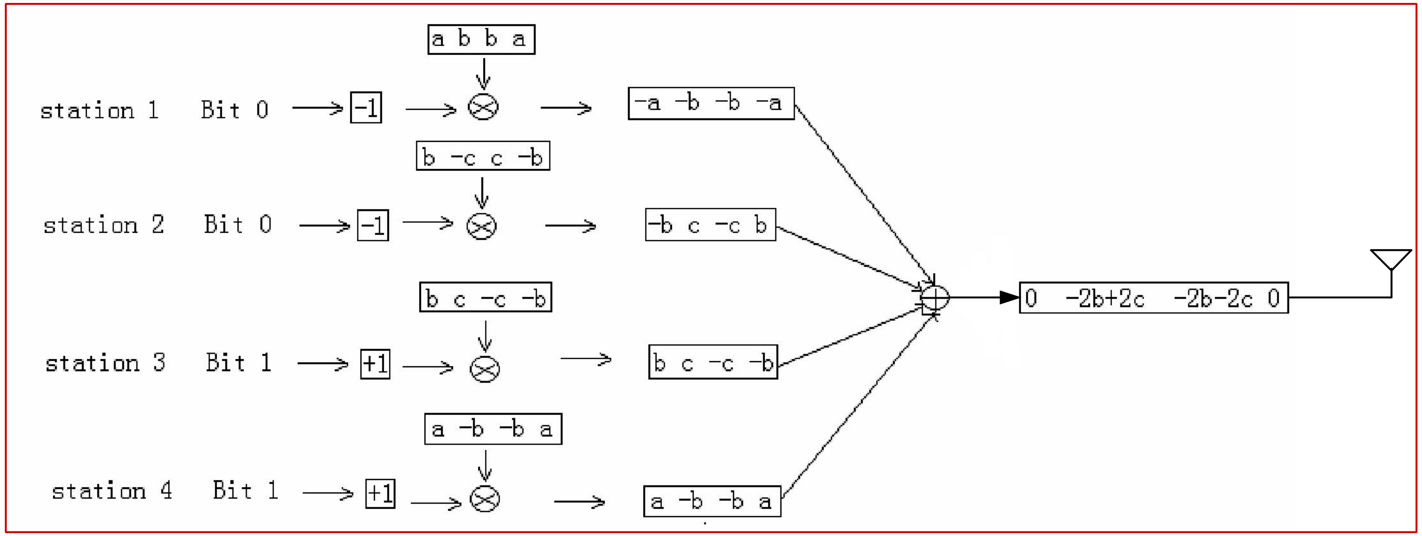
Its inverse matrix is:

$$[J]_4^{-1} = \frac{1}{4}\begin{bmatrix} \frac{1}{a} & \frac{1}{b} & \frac{1}{b} & \frac{1}{a} \\ \frac{1}{b} & -\frac{1}{c} & \frac{1}{c} & -\frac{1}{b} \\ \frac{1}{b} & \frac{1}{c} & -\frac{1}{c} & -\frac{1}{b} \\ \frac{1}{a} & -\frac{1}{b} & -\frac{1}{b} & \frac{1}{a} \end{bmatrix}$$

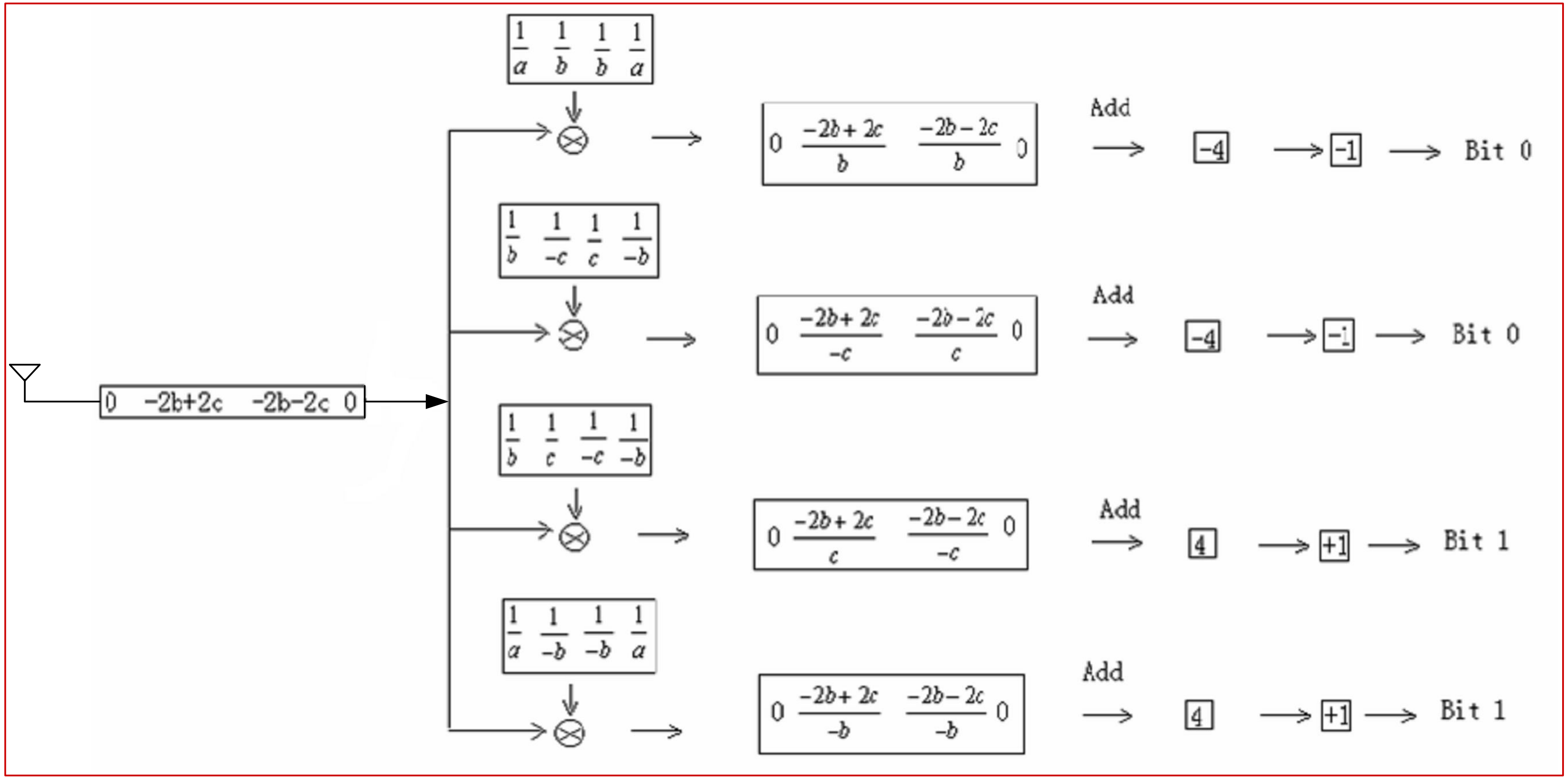
and: $[J]_4 [J]_4^{-1} = [I]_4$

## Sending message 0 0 1 1

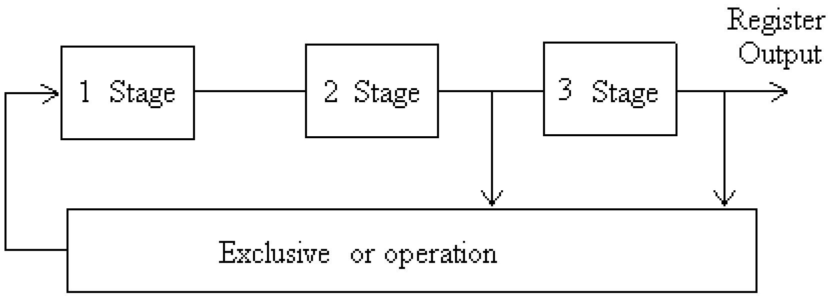
CDMA Multiplexer:



## CDMA Demultiplexer:



# PN Sequence




| | output | output | output | Output |
|---|---|---|---|---|
| movement | 1 stage | 2 stage | 3 stage | register |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 2 | 1 | 1 | 1 | 1 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 0 | 0 | 1 | 1 |
| 5 | 1 | 0 | 0 | 0 |
| 6 | 0 | 1 | 0 | 0 |
| 7 | 1 | 0 | 1 | 1 |

# Output matrix is Jacket matrix:

Output matrix is Jacket matrix: $(1 \longrightarrow -1,\ 0 \longrightarrow 1)$

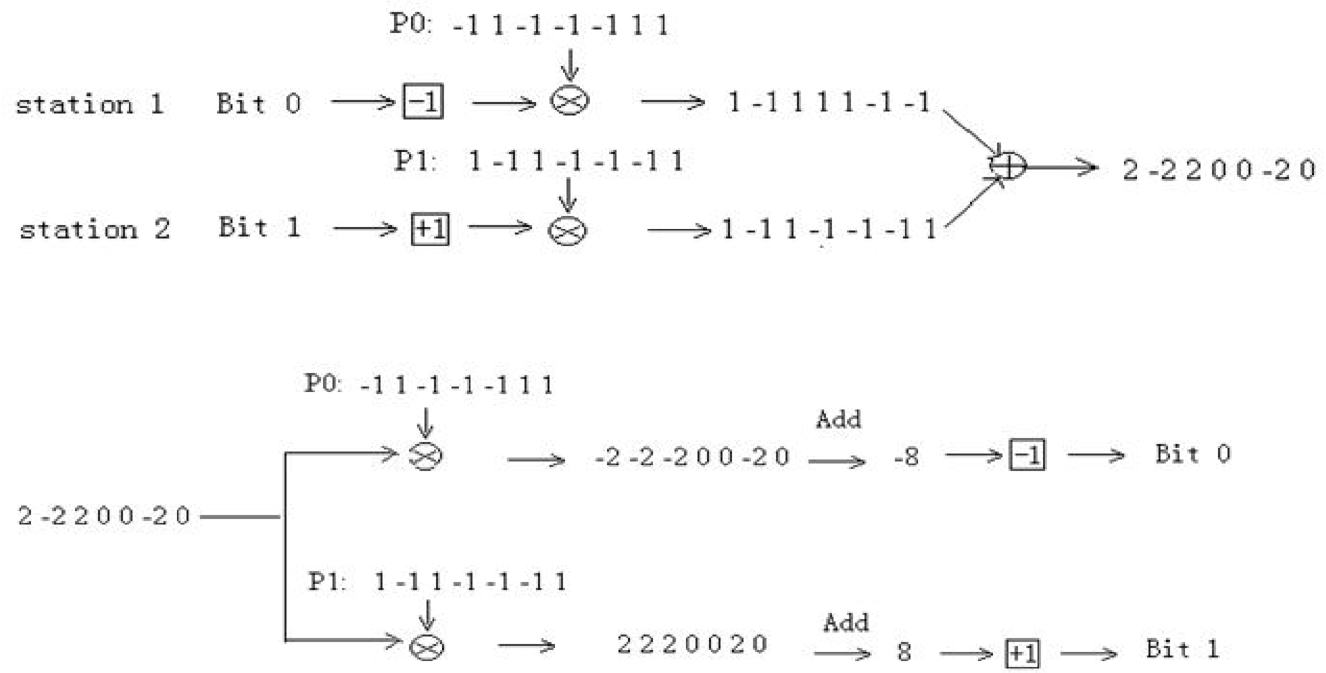
$$P = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 \\ 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & -1 & 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

P0
P1
⋮
P5
P6

and,

$$P^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 \end{bmatrix}$$

# Output matrix is Jacket matrix:



# Fibonacci Jacket Conference Matrix GF(7)

$$F_{n+1} = F_n + F_{n-1}, \quad F_0 = 0, F_1 = 1.$$

$$\begin{pmatrix} F_n & F_{n+1} \\ F_{n+1} & F_{n+2} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \cdots \cdot \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}}_{n}$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix}, \begin{pmatrix} 2 & 3 \\ 3 & 5 \end{pmatrix}, \begin{pmatrix} 3 & 5 \\ 5 & 8 \end{pmatrix}, \ldots$$

$$\frac{1}{2} \bmod 7 = 4 : 2 \times 4 \bmod 7 = 1$$

$$\frac{1}{3} \bmod 7 = 5 : 3 \times 5 \bmod 7 = 1$$

$$\frac{1}{4} \bmod 7 = 2 : 4 \times 2 \bmod 7 = 1$$

$$\frac{1}{5} \bmod 7 = 3 : 3 \times 5 \bmod 7 = 1$$

$$\frac{1}{6} \bmod 7 = 6 : 6 \times 6 \bmod 7 = 1$$

$$-1 = 6 \bmod 7$$

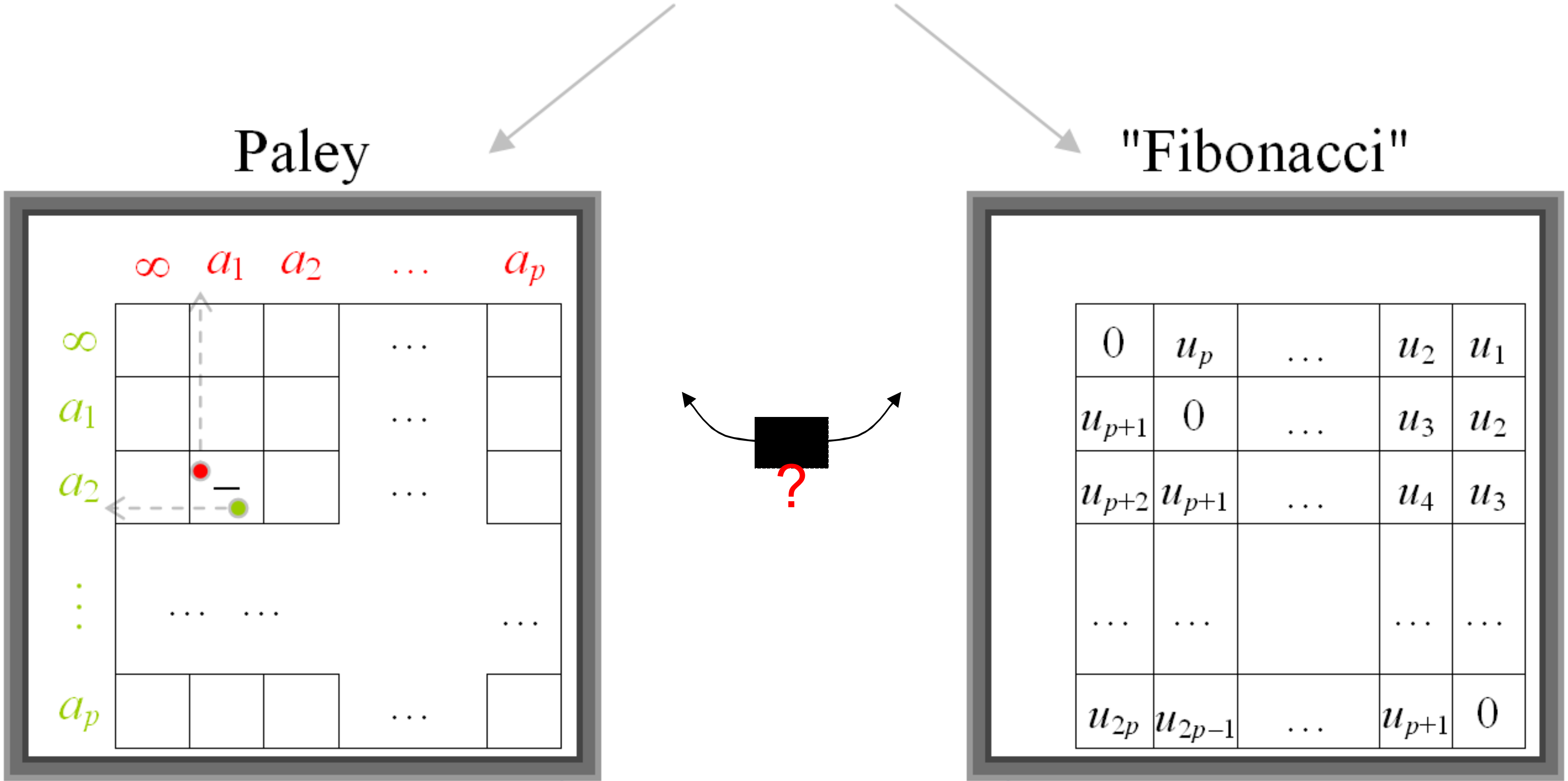
$$[J]_8 [J]_8^{-1} = (n-1) I_8 = 0 (: \bmod 7)$$

$\cdot -1$

$$J_8 = \begin{vmatrix} 0 & 6 & 1 & 5 & 3 & 2 & 1 & 1 \\ 6 & 0 & 6 & 1 & 5 & 3 & 2 & 1 \\ 6 & 6 & 0 & 6 & 1 & 5 & 3 & 2 \\ 5 & 6 & 6 & 0 & 6 & 1 & 5 & 3 \\ 4 & 5 & 6 & 6 & 0 & 6 & 1 & 5 \\ 2 & 4 & 5 & 6 & 6 & 0 & 6 & 1 \\ 6 & 2 & 4 & 5 & 6 & 6 & 0 & 6 \\ 1 & 6 & 2 & 4 & 5 & 6 & 6 & 0 \end{vmatrix} \bmod 7, \quad J_8^{-1} = \cdot \begin{vmatrix} 0 & 6 & 6 & 3 & 2 & 4 & 6 & 1 \\ 6 & 0 & 6 & 6 & 3 & 2 & 4 & 6 \\ 1 & 6 & 0 & 6 & 6 & 3 & 2 & 4 \\ 3 & 1 & 6 & 0 & 6 & 6 & 3 & 2 \\ 5 & 3 & 1 & 6 & 0 & 6 & 6 & 3 \\ 4 & 5 & 3 & 1 & 6 & 0 & 6 & 6 \\ 1 & 4 & 5 & 3 & 1 & 6 & 0 & 6 \\ 1 & 1 & 4 & 5 & 3 & 1 & 6 & 0 \end{vmatrix}.$$

# Fibonacci Jacket Conference Matrix GF(p)

$$F := \mathrm{GF}(p);$$




$F=\{a_1,a_2,\ldots,a_p\}.$ $u_{i+1}=\alpha.u_i-\alpha.u_{i-1},\ u_i\neq 0.$

$u_{i+1}=a.u_i-b.u_{i-1}$. → $(a,b)=(1,1)$ - Fibonacci sequence.

→ $(a,b)=(\alpha,-\alpha)$ - hire…

# Example: GF(5)

$$p = \left[\begin{array}{c|c|c|c|c|c|c} & \infty & 0 & 1 & 2 & 3 & 4 \\ \hline \infty & 0 & 1 & 1 & 1 & 1 & 1 \\ \hline 0 & 1 & 0 & 1 & 2 & 3 & 4 \\ \hline 1 & 1 & -1 & 0 & 1 & 2 & 3 \\ \hline 2 & 1 & -2 & -1 & 0 & 1 & 2 \\ \hline 3 & 1 & -3 & -2 & -1 & 0 & 1 \\ \hline 4 & 1 & -4 & -3 & -2 & -1 & 0 \end{array}\right]$$

$$J_6 = \left[\begin{array}{c|c|c|c|c|c} 0 & 4 & 4 & 1 & 3 & 1 \\ \hline 3 & 0 & 4 & 4 & 1 & 3 \\ \hline 4 & 3 & 0 & 4 & 4 & 1 \\ \hline 3 & 4 & 3 & 0 & 4 & 4 \\ \hline 2 & 3 & 4 & 3 & 0 & 4 \\ \hline 2 & 3 & 3 & 4 & 3 & 0 \end{array}\right]$$

$$J_6^{-1} = \left[\begin{array}{c|c|c|c|c|c} 0 & 2 & 4 & 2 & 3 & 3 \\ \hline 4 & 0 & 2 & 4 & 2 & 2 \\ \hline 4 & 4 & 0 & 2 & 4 & 2 \\ \hline 1 & 4 & 4 & 0 & 2 & 4 \\ \hline 2 & 1 & 4 & 4 & 0 & 2 \\ \hline 1 & 2 & 1 & 4 & 4 & 0 \end{array}\right]$$

Fix signs

$$u_{i+1} = 3.u_i - 3.u_{i-1}$$

$$u_1, u_2, u_3, \ldots$$

$$0,1,3,1,4,4,0,3,4,3,2,2,\ldots$$

# Fast Block Center Weighted Hadamard Transform

Moon Ho Lee, *Senior Member, IEEE*, and Xiao-Dong Zhang

***Abstract*—Motivated by the Hadamard transforms and center weighted Hadamard transforms, a new class of block center weighted Hadamard transforms (BCWHT) are proposed, which weights the region of midspatial frequencies of the signal more than the Hadamard transform. Based on the Kronecker product, direct sum operations, the identity matrix and recursive relations, the proposed one and 2-D fast BCWHTs algorithms through sparse matrix factorization are simply obtained.**

***Index Terms*—Block center weighted Hadamard transform (BCWHT), center weighted Hadamard transform (CWHT), fast algorithm, sparse matrix decomposition.**

## I. Introduction

THE Walsh–Hadamard transform (WHT) and discrete Fourier transform (DFT) are highly practical value for representing signals, images and mobile communications for orthogonal code designs ([1]–[4] and [5]). With the technology rapid development, communication systems will require more and more transmission and storage capacities of multilevel cases in cochannels for numerous clients. Recently, variations of WHT and DFT called center weighted Hadamard transform (CWHT) ([6], [7]) and complex reverse Jacket transform (CRJT) ([8]–[10] and [11]) have been proposed and their applications to image processing and communications have been reported. When the center part of data sequences or the middle range of frequency components are more important, the CWHT can offer better quality than the WHT ([2], [3] and [4]).

In this paper, motivated by the Hadamard transforms and CWHTs ([6] and [7]), we propose a new block center weighted Hadamard transform (BCWHT) in Section II, which may be applied to multilevel cases in communication. In Section III, based on the Kronecker product and direct sum operations, a fast 1-D BCWHT algorithm is proposed. In Section IV, a fast 2-D BCWHT algorithm is presented through sparse matrix factorization and the Kronecker product. Finally, in Section V, we make our conclusion.

## II. BCWHT

For the 1-D BCWHT matrix $[B]_{2N}$ of order $2N$ with $N = 2^k$, which is partitioned to the $N \times N$ block matrix, we can transform a temporal spatial vector $x$ into a transformed vector $y$ by

$$y = [B]_{2N}x. \qquad (1)$$

Let $[I]_2$ and $[H]_2$ be the $2\times 2$ identity and the lowest order Hadamard matrices, respectively,.i.e,

$$[I]_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad [H]_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \qquad (2)$$

Denote by $[A]_2 = (1/\sqrt{2})[H]_2$. The lowest order BCWHT matrix of order 8 is defined to be

$$[B]_{2\times 4} = \begin{pmatrix} [I]_2 & [I]_2 & [I]_2 & [I]_2 \\ [I]_2 & -[A]_2 & [A]_2 & -[I]_2 \\ [I]_2 & [A]_2 & -[A]_2 & -[I]_2 \\ [I]_2 & -[I]_2 & -[I]_2 & [I]_2 \end{pmatrix} \qquad (3)$$

with each block being $2\times 2$ submatrices. Since

$$[A]_2[A]_2^T = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^T = [I]_2 \qquad (4)$$

$$[A]_2^{-1} = [A]_2^T = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \qquad (5)$$

Since

$$\begin{pmatrix} [I]_2 & [I]_2 & [I]_2 & [I]_2 \\ [I]_2 & -[A]_2 & [A]_2 & -[I]_2 \\ [I]_2 & [A]_2 & -[A]_2 & -[I]_2 \\ [I]_2 & -[I]_2 & -[I]_2 & [I]_2 \end{pmatrix} \times \begin{pmatrix} [I]_2 & [I]_2 & [I]_2 & [I]_2 \\ [I]_2 & -[A]_2^{-1} & [A]_2^{-1} & -[I]_2 \\ [I]_2 & [A]_2^{-1} & -[A]_2^{-1} & -[I]_2 \\ [I]_2 & -[I]_2 & -[I]_2 & [I]_2 \end{pmatrix} = \begin{pmatrix} 4[I]_2 & 0 & 0 & 0 \\ 0 & 4[I]_2 & 0 & 0 \\ 0 & 0 & 4[I]_2 & 0 \\ 0 & 0 & 0 & 4[I]_2 \end{pmatrix} \qquad (6)$$

the inverse of (3) is

$$[B]_{2\times 4}^{-1} = \frac{1}{4}\begin{pmatrix} [I]_2 & [I]_2 & [I]_2 & [I]_2 \\ [I]_2 & -[A]_2^{-1} & [A]_2^{-1} & -[I]_2 \\ [I]_2 & [A]_2^{-1} & -[A]_2^{-1} & -[I]_2 \\ [I]_2 & -[I]_2 & -[I]_2 & [I]_2 \end{pmatrix} = \frac{1}{4}\begin{pmatrix} [I]_2 & [I]_2 & [I]_2 & [I]_2 \\ [I]_2 & -[A]_2^T & [A]_2^T & -[I]_2 \\ [I]_2 & [A]_2^T & -[A]_2^T & -[I]_2 \\ [I]_2 & -[I]_2 & -[I]_2 & [I]_2 \end{pmatrix}. \qquad (7)$$

Manuscript received August 29, 2006; revised April 18, 2007. This work was supported by the Ministry of information and Communication (MIC), Korea, under the IT Foreign Specialist Inviting Program (ITFSIP) supervised by the Institute of Information Technology Assessment (IITA) of the Ministry of Science and Technology, Korea. This paper was recommended by Associate Editor S.-M. Phoong.

M. H. Lee is with the Institute of Information and Communication, Chonbuk National University, Jeonju 561-756, Korea (e-mail: moonho@chonbuk.ac.kr).

X.-D. Zhang is with the Department of Mathematics, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: xiaodong@sjtu.edu.cn).

Digital Object Identifier 10.1109/TCSI.2007.905655

Certainly,

$$[B]_{2\times4}[B]^{-1}_{2\times4} = [I]_8. \tag{8}$$

This choice of block weighting was indicated, to a large extent, by requirement of digital hardware simplicity. With the aid of Kronecker product and Hadamard matrices, the higher order BCWHT matrix is given by the following recursive relation:

$$[B]_{2N} \equiv [B]_N \otimes [H]_2, \quad N \geq 8 \tag{9}$$

where $\otimes$ is the Kronecker product and $[H]_2$ is the lowest order Hadamard matrix. We are able to show that

$$[B]^{-1}_{2N} = \frac{1}{N}[B]^T_{2N}. \tag{10}$$

We can use the induction method to prove this assertion. From (7), (10) holds for $2N = 8$. Assume that (10) holds for $N$, i.e.,

$$[B]_N[B]^T_N = \frac{N}{2}[I]_N, \ [B]^{-1}_N = \frac{2}{N}[B]^T_N. \tag{11}$$

Now we show that (10) holds for $2N$.

$$\begin{aligned}[B]_{2N}[B]^T_{2N} &= ([B]_N \otimes [H]_2)([B]_N \otimes [H]_2)^T \\ &= ([B]_N \otimes [H]_2)([B]^T_N \otimes [H]^T_2) \\ &= ([B]_N[B]^T_N) \otimes ([H]_2[H]^T_2) \\ &= \frac{N}{2}[I]_N \otimes (2[I]_2) \\ &= N[I]_{2N}.\end{aligned}$$

Hence, (10) holds. Therefore, BCWHT is a class of transforms which are simple to calculate and easily inverted. Further, the inverse BCWHT can be written as follows:

$$[x]_{2N} = [B]^{-1}_{2N}[y]_{2N} = \frac{1}{N}[B]^T_{2N}[y]_{2N}. \tag{12}$$

## III. Fast 1-D BCWHT Algorithm

In order to develop the fast 1-D BCWHT algorithm, we first introduce three block permutation matrices $[P]^r_4$, $[J]_4$ and $[P]^c_4$ as follows:

$$[P]^r_8 \equiv \begin{pmatrix} [I]_2 & [0]_2 & [0]_2 & [0]_2 \\ [0]_2 & [0]_2 & [0]_2 & [I]_2 \\ [0]_2 & [0]_2 & [I]_2 & [0]_2 \\ [0]_2 & [I]_2 & [0]_2 & [0]_2 \end{pmatrix}$$

$$[J]_4 \equiv \begin{pmatrix} [0]_2 & [I]_2 \\ [I]_2 & [0]_2 \end{pmatrix} \tag{13}$$

$$[P]^c_8 \equiv \begin{pmatrix} [I]_4 & [0]_4 \\ [0]_4 & [J]_4 \end{pmatrix} \equiv [I]_4 \oplus [J]_4 \tag{14}$$

where $\oplus$ denotes the direct sum operator(see [12]). Then we transform the BCWHT of order 8 as

$$[\widetilde{B}]_8 \equiv [P]^r_8[B]_8[P]^c_8 = \left(\begin{array}{cc|cc} [I]_2 & [I]_2 & [I]_2 & [I]_2 \\ [I]_2 & -[I]_2 & [I]_2 & -[I]_2 \\ \hline [I]_2 & [A]_2 & -[I]_2 & -[A]_2 \\ [I]_2 & -[A]_2 & -[I]_2 & [A]_2 \end{array}\right). \tag{15}$$

The general permutation matrices $[P]^r_{2N}$ and $[P]^c_{2N}$ are defined to be

$$[P]^r_{2N} = [P]^r_8 \otimes [I]_{N/4} \tag{16}$$

and

$$\begin{aligned}[P]^c_{2N} &= ([I]_4 \oplus [J]_4) \otimes [I]_{N/4} \\ &= [I]_N \oplus ([J]_4 \otimes [I]_{N/4}).\end{aligned} \tag{17}$$

It is easily checked that both permutation matrices are unitary which satisfy

$$[P]^r_{2N}([P]^r_{2N})^T = ([P]^r_{2N})^T\,[P]^r_{2N} = [I]_{2N} \tag{18}$$

and

$$[P]^c_{2N}([P]^c_{2N})^T = ([P]^c_{2N})^T[P]^c_{2N} = [I]_{2N}. \tag{19}$$

Using the two permutation matrices and the definition of BCWHT, the order of $2N$ BCWHT matrix $[B]_{2N}$ can be transformed to the following

$$\begin{aligned}[\widetilde{B}]_{2N} &\equiv [P]^r_{2N}[B]_{2N}[P]^c_{2N} \\ &= [P]^r_{2N}([B]_N \otimes [H]_2)[P]^c_{2N} \\ &= ([P]^r_8 \otimes [I]_{N/4})([B]_8 \otimes [H]_{N/4})([P]^c_8 \otimes [I]_{N/4}) \\ &= ([P]^r_8[B]_8[P]^c_8) \otimes ([I]_{N/4}[H]_{N/4}[I]_{N/4}) \\ &= [\widetilde{B}]_8 \otimes [H]_{N/4}.\end{aligned} \tag{20}$$

Hence $[\widetilde{B}]_{2N}$ can be written as (21) shown at the bottom of the next page. Let $[Q]_4$ be the permutation matrix of order 4 as follows:

$$[Q]_4 = \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right). \tag{22}$$

Then

$$[H]_2 \otimes [I]_2 = [Q]_4([I]_2 \otimes [H]_2)[Q]^T_4. \tag{23}$$

TABLE I

COMPUTATIONAL COMPLEXITY OF THE PROPOSED FAST ALGORITHMS FOR 1-D BCWHT TRANSFORM OF ORDER $N = 2^k$

| | Directed Computation | Proposed algorithm $N = 2^k$ |
|---|---|---|
| Additions | $N(N-1)$ | $N\log_2 N$ |
| Multiplications | $N \times N$ | $\frac{N}{4}$ |

From (21) and (23), we have

$$
\begin{aligned}
[E]_N &\equiv \begin{pmatrix} [I]_2 \otimes [H]_{N/4} & [I]_2 \otimes [H]_{N/4} \\ [I]_2 \otimes [H]_{N/4} & -[I]_2 \otimes [H]_{N/4} \end{pmatrix} \\
&= [H]_2 \otimes ([I]_2 \otimes [H]_{N/4}) \\
&= ([H]_2 \otimes [I]_2) \otimes [H]_{N/4} \\
&= ([Q]_4([I]_2 \otimes [H]_2)[Q]_4^T) \otimes ([I]_{N/4}[H]_{N/4}[I]_{N/4}) \\
&= ([Q]_4 \otimes [I]_{N/4})([I]_2 \otimes [H]_2 \otimes [H]_{N/4}) \times ([Q]_4^T \otimes [I]_{N/4}) \\
&= ([Q]_4 \otimes [I]_{N/4})([I]_2 \otimes [H]_{N/2})([Q]_4^T \otimes [I]_{N/4}).
\end{aligned}
$$

From (21) and (23), we have

$$
\begin{aligned}
[F]_N &\equiv \begin{pmatrix} [I]_2 \otimes [H]_{N/4} & [A]_2 \otimes [H]_{N/4} \\ [I]_2 \otimes [H]_{N/4} & -[A]_2 \otimes [H]_{N/4} \end{pmatrix} \\
&= \begin{pmatrix} [I]_2 & [A]_2 \\ [I]_2 & -[A]_2 \end{pmatrix} \otimes [H]_{N/4} \\
&= \left( \begin{pmatrix} [I]_2 & [I]_2 \\ [I]_2 & -[I]_2 \end{pmatrix} \begin{pmatrix} [I]_2 & [0]_2 \\ [0]_2 & [A]_2 \end{pmatrix} \right) \otimes [H]_{N/4} \\
&= \left( ([H]_2 \otimes [I]_2) \begin{pmatrix} [I]_2 & [0]_2 \\ [0]_2 & [A]_2 \end{pmatrix} \right) \otimes [H]_{N/4} \\
&= (([Q]_4([I]_2 \otimes [H]_2)[Q]_4^T) \\
&\quad \times \begin{pmatrix} [I]_2 & [0]_2 \\ [0]_2 & [A]_2 \end{pmatrix} ) \otimes ([H]_{N/4}[I]_{N/4}) \\
&= (([Q]_4([I]_2 \otimes [H]_2)[Q]_4^T) \otimes [H]_{N/4}) \\
&\quad \times \left( \begin{pmatrix} [I]_2 & [0]_2 \\ [0]_2 & [A]_2 \end{pmatrix} \otimes [I]_{N/4} \right) \\
&= ([Q]_4 \otimes [I]_{N/4})([I]_2 \otimes [H]_{N/2})([Q]_4^T \otimes [I]_{N/4}) \\
&\quad \times \left( \begin{pmatrix} [I]_2 & [0]_2 \\ [0]_2 & [A]_2 \end{pmatrix} \otimes [I]_{N/4} \right) \\
&\equiv [E]_N[U]_N
\end{aligned}
$$

where

$$
[U]_N \equiv \begin{pmatrix} [I]_2 & [0]_2 \\ [0]_2 & [A]_2 \end{pmatrix} \otimes [I]_{N/4}.
$$

Based on the matrix identity in [12]

$$
\begin{pmatrix} E_N & [E]_N \\ [F]_N & -[F]_N \end{pmatrix} = ([E]_N \oplus [F]_N)([H]_2 \otimes [I]_N). \quad (24)
$$

Therefore, the proposed fast 1-D BCWHT algorithm is written as

$$
[B]_{2N} = (P_{2N}^r)^T \left([E]_N \oplus [E]_N[U]_N\right)(H_2 \otimes [I]_N)\,(P_{2N}^c)^T. \quad (25)
$$

The permutation matrices $(P_{2N}^r)^T$ and $(P_{2N}^c)^T$ do not require computation, since they just perform data permutation. The operation $[H]_2 \otimes [I]_N$ requires $N$ additions, since it can be performed by $N/2$ butterflies. Since the permutation matrices $[Q]_4 \otimes [I]_{N/4}$ and $([Q]_4 \otimes [I]_{N/4})^T$ do not require computation, the operations of $[E]_N = ([Q]_4 \otimes [I]_{N/4})([I]_2 \otimes [H]_{N/2})([Q]_4^T \otimes [I]_{N/4})$ is the same as these of $[H]_{N/2}$. It is known that the Hadamard transform $[H]_{N/2}$ of order $N/2$ requires $(N/2)\log_2(N/2)$ additions. Thus, the operation $[E]_N$ requires $(N/2)\log_2(N/2)$ additions. On the other hand, the operation $[E]_N[U]_N$ first performs $[U]_N$, which is equivalent to $N/4$ direct connections and $N/4$ multiplications, then executes an operation $[E]_N$ which needs $(N/2)\log_2(N/2)$ additions. Because the direct sum $[E]_N \oplus [E]_N[U]_N$ can be independently divided into two parts $[E]_N$ and $[E]_N[U]_N$, the operation $[E]_N \oplus [E]_N[U]_N$ requires $N\log_2(N/2)$ additions and $N/4$ multiplications. Hence, the proposed algorithm depicted in (25) requires, in total, $N + (N/2)\log_2(N/2) + (N/2)\log_2(N/2) = N\log_2 N$ real additions and $N/4$ real multiplications for $N = 2^k$, while the direct computation for the BCWHT transform requires $N(N-1)$ additions and $N^2$ multiplications. The results are summed up as the following Table I. Moreover, Fig. 1 presents an example for a fast BCWHT flow graph with $N = 8$. The first, second and last steps stand for data permutation, while the third and fourth steps stand for additions and multiplications.

## IV. 2-D FAST ALGORITHM TRANSFORM

The 2-D transforms a temporal/spatial matrix $X$ into a transformed matrix $Y$ as (see [12])

$$
Y = [B]_{2N}\, X\, ([B]_{2N})^T. \quad (26)
$$

$$
[\widetilde{B}]_{2N} = \left(\begin{array}{cc|cc} [I]_2 \otimes [H]_{N/4} & [I]_2 \otimes [H]_{N/4} & [I]_2 \otimes [H]_{N/4} & [I]_2 \otimes [H]_{N/4} \\ [I]_2 \otimes [H]_{N/4} & -[I]_2 \otimes [H]_{N/4} & [I]_2 \otimes [H]_{N/4} & -[I]_2 \otimes [H]_{N/4} \\ \hline [I]_2 \otimes [H]_{N/4} & [A]_2 \otimes [H]_{N/4} & -[I]_2 \otimes [H]_{N/4} & -[A]_2 \otimes [H]_{N/4} \\ [I]_2 \otimes [H]_{N/4} & -[A]_2 \otimes [H]_{N/4} & -[I]_2 \otimes [H]_{N/4} & [A]_2 \otimes [H]_{N/4} \end{array}\right) \equiv \begin{pmatrix} E_N & [E]_N \\ [F]_N & -[F]_N \end{pmatrix} \quad (21)
$$

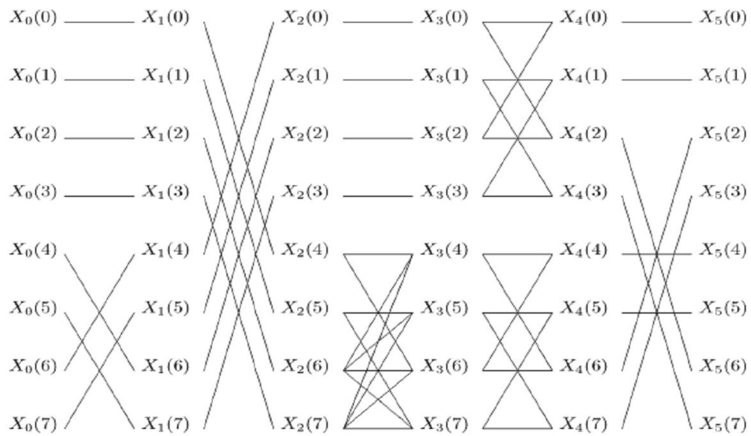



Fig. 1. Fast BCWHT flow graph for $8 \times 8$.

Generally, the linear transform of matrix $X$ shown as $AXB = Y$ can be expressed by the transformation of the column-wise stacking vector of $X$ as (see [12])

$$(B^T \otimes A) \cdot \text{vec}(X) = \text{vec}(Y).$$

Thus, the 2-D BCWHT matrix in (26) can be expressed by

$$\text{vec}(Y) = ([B]_{2N} \otimes [B]_{2N}) \cdot \text{vec}(X). \quad (27)$$

In order to develop the fast 2-D BCWHT algorithm, we start to define a coefficient block matrix $[S]_{2N}$ by

$$[S]_{2N} \equiv ([H]_4 \otimes [I]_2 \otimes [H]_{N/4})[B]_{2N}, \quad N \geq 4. \quad (28)$$

Hence, $[S]_{2N}$ can be partitioned to an $N \times N$ block matrix whose blocks are $2 \times 2$ submatrices. It can be shown that $[S]_{2N}$ is a sparse block matrix with at most two nonzero blocks for each row and each column block. In order to prove this assertion, we first start to compute the lowest $4 \times 4$ block matrix

$$\begin{aligned}
[S]_8 &= ([H]_4 \otimes [I]_2 \otimes [H]_{4/4})[B]_8 \\
&= \begin{pmatrix} [I]_2 & [I]_2 & [I]_2 & [I]_2 \\ [I]_2 & -[I]_2 & [I]_2 & -[I]_2 \\ [I]_2 & [I]_2 & -[I]_2 & -[I]_2 \\ [I]_2 & -[I]_2 & -[I]_2 & [I]_2 \end{pmatrix} \\
&\quad \times \begin{pmatrix} [I]_2 & [I]_2 & [I]_2 & [I]_2 \\ [I]_2 & -[A]_2 & [A]_2 & -[I]_2 \\ [I]_2 & [A]_2 & -[A]_2 & -[I]_2 \\ [I]_2 & -[I]_2 & -[I]_2 & [I]_2 \end{pmatrix} \\
&= \begin{pmatrix} 4[I]_2 & [0]_2 & [0]_2 & [0]_2 \\ [0]_2 & 2([I]_2+[A]_2) & 2([I]_2-[A]_2) & [0]_2 \\ [0]_2 & 2([I]_2-[A]_2) & 2([I]_2+[A]_2) & [0]_2 \\ [0]_2 & [0]_2 & [0]_2 & 4[I]_2 \end{pmatrix}.
\end{aligned} \quad (29)$$

Clearly, $[S]_8$ is a sparse matrix whose each row and each column block is only at most two nonzero blocks. Moreover, by a calculation, we have

$$\begin{aligned}
[S]_8 &= ([H]_4 \otimes [I]_2 \otimes [H]_{4/4})[B]_8 \\
&= [B]_8([H]_4 \otimes [I]_2 \otimes [H]_{4/4}).
\end{aligned}$$

Clearly, using the properties of Hadamard matrix and the Kronecker product, (28) can be rewritten

$$\begin{aligned}
[S]_{2N} &= ([H]_4 \otimes [I]_2 \otimes [H]_{N/4})([B]_N \otimes [H]_2) \\
&= \left(([H]_4 \otimes [I]_2 \otimes [H]_{N/8}) \otimes [H]_2\right)([B]_N \otimes [H]_2) \\
&= \left(([H]_4 \otimes [I]_2 \otimes [H]_{N/8})[B]_N\right) \otimes ([H]_2[H]_2) \\
&= [S]_N \otimes (2[I]_2).
\end{aligned} \quad (30)$$

Since $[S]_8$ is block symmetric and has at most two nonzero blocks in each row and each column blocks, $[S]_{2N}$ is a sparse block matrix with at most two nonzero blocks for each row and each column block by using the recursion relation of (30). Further from $[H]_4^{-1} = (1/4)[H]_4$, $[H]_{N/4}^{-1} = (4/N)[H]_{N/4}$ and (28), the BCWHT transform $[B]_{2N}$ can be written

$$\begin{aligned}
[B]_{2N} &= ([H]_4 \otimes [I]_2 \otimes [H]_{N/4})^{-1}[S]_{2N} \\
&= \frac{1}{N}([H]_4 \otimes [I]_2 \otimes [H]_{N/4})[S]_{2N}.
\end{aligned} \quad (31)$$

Based on (27), (28) and (31), the 2-D BCWHT transform in (27) can be written as

$$\begin{aligned}
&[B]_{2N} \otimes [B]_{2N} \\
&= ([I]_{2N}[B]_{2N}) \otimes ([B]_{2N}[I]_{2N}) \\
&= ([I]_{2N} \otimes [B]_{2N})([B]_{2N} \otimes [I]_{2N}) \\
&= \left([I]_{2N} \otimes \left(\frac{1}{N}([H]_4 \otimes [I]_2 \otimes [H]_{N/4})[S]_{2N}\right)\right) \\
&\quad \times \left(\left(\frac{1}{N}([H]_4 \otimes [I]_2 \otimes [H]_{N/4})[S]_{2N}\right) \otimes [I]_{2N}\right)
\end{aligned}$$

TABLE II
COMPUTATIONAL COMPLEXITY OF THE PROPOSED FAST ALGORITHMS FOR 2-D BCWHT TRANSFORM OF ORDER $N^2 = 2^{2k}$

| | Directed Computation | Proposed algorithm $N = 2^k$ |
|---|---|---|
| Additions | $N^2(N^2-1)$ | $(1/2 + 2\log_2^2 N)N^2$ |
| Multiplications | $N^2 \times N^2$ | $9/2N^2$ |

$$= \frac{1}{N^2}([I]_{2N} \otimes ([H]_4 \otimes [I]_2 \otimes [H]_{N/4})) \times ([I]_{2N} \otimes [S]_{2N})(([H]_4 \otimes [I]_2 \otimes [H]_{N/4}) \otimes [I]_{2N})([S]_{2N} \otimes [I]_{2N}). \quad (32)$$

From (32), The fast algorithm for the 2-D BCWHT transform requires four iterations. It is known that the Hadamard transform of order $N = 2^k$ requires $kN = N\log_2 N$ additions. Thus, $[I]_N \otimes ([H]_4 \otimes [I]_2 \otimes [H]_{N/4})$ require $(kN)^2 = N^2\log_2^2 N$ additions. Since $[S]_{2N}$ is a sparse block matrix with at most two nonzero blocks for each row and each column block, it is easy to see that $[S]_{2N}$ requires $N/2$ additions and $(3/2)N$ multiplications. Then $([I]_{2N} \otimes [S]_{2N})$ needs $(N/2)^2 = N^2/4$ additions and $((3/2)N)^2 = (9/4)N^2$ multiplications. Therefore, the 2-D fast algorithm for BCWHT transform requires $(1/2 + 2\log_2^2 N)N^2$ additions and $(9/2)N^2$ multiplications, while the direct computation needs $N^2(N^2-1)$ additions and $N^4$ multiplications. Table II shows that our proposed algorithm in (32) requires fewer operations.

## V. CONCLUSION

A new nonorthogonal transform, the BCWHT was introduced in this paper. Based on the Kronecker product, direct sum operators and sparse matrix factorization, one and 2-D fast algorithms for the BCWHT are proposed and their inverses are derived. With low complexity and highly regular modularity, the proposed one and 2-D fast algorithms advance the applicability of the BCWHT for image, signal processing and orthogonal code design for mobile communications ([2]–[4] and [5]).

## ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for valuable comments, corrections and suggestions, which results in an improvement of the original manuscript.

## REFERENCES

[1] N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing*. New York: Springer-Verlag, 1975.
[2] D. C. Park, M. H. Lee, and E. A. Choi, "Revisited DFT matrix via the reverse Jacket transform and its applications to communication," in *Proc. 22nd Symp. Inf. Theory Appl. (SITA'99)*, Yuzawa, Niigata, Japan, Nov.-Dec. 30–3, 1999, pp. 427–430.
[3] J. Hou and M. H. Lee, "Enhancing data throughput and lower correlations quasi-orthogonal functions for 3G CDMA systems.," *Int. J. Commun. Syst*, to be published.
[4] J. Hou and M. H. Lee, "QPSK differential space time coding on different unitary matrices sets and initialization," *Int. J. Commun. Syst*, to be published.
[5] J. Hou and M. H. Lee, "Construction of the dual OVSF codes with lower correlations," *IEICE Trans.*, to be published.
[6] M. H. Lee, "The center weighted Hadamard transform," *IEEE Trans. Circuits Syst.*, vol. 36, no. 9, pp. 1247–1249, Sep. 1989.
[7] C. P. Fan and J. F. Yang, "Fast center weighted Hadamard transform algorithms," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 45, no. 3, pp. 429–432, Mar. 1998.
[8] M. H. Lee, "A new reverse Jacket tranform and its fast algorithm," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 47, no. 1, pp. 39–47, Jan. 2000.
[9] K. Finlayson, M. H. Lee, J. Seberry, and M. Yamada, "Jacket matrices constructed from Hadamard matrix and generalized Hadamard matrices," *Australas. J. Combinatorics*, vol. 35, no. 1, pp. 83–88, Jun. 2006.
[10] K. J. Horadam, "A generalised Hadamard transform," in *Proc. 2005 IEEE Int. Symp. Inf. Theory (ISIT)*, Australia, Sep. 4–9, 2005.
[11] M. H. Lee and J. Hou, "Fast block inverse Jacket transform," *IEEE Signal Process. Lett.*, vol. 13, pp. 461–464, Aug. 2006.
[12] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. New York: Cambridge Univ. Press, 1991.

**Moon Ho Lee** (S'81–M'85–SM'86) received the Ph.D. degree in electronic engineering from Chonnam National University, Kwangju, Korea in 1984, and the University of Tokyo, Tokyo, Japan, in 1990, respectively.

From 1970 to 1980, he was a Chief Engineer with Namyang Moonhwa Broadcasting Corp., Jeju, Korea. Since 1980, he was been a Professor in the Department of information and Communication at Chonbuk National University, Jeonju, Korea. From 1985 to 1986, he was also with the University of Minnesota as a Post-Doctoral Researcher. He has held visiting positions with the University of Hannover (1990), University of Aachen (1992,1996), University of Munich (1998), University of Kaislerlautern (2001), RMIT (2004), and the University of Wollongong, Australia. His research interests include multidimensional source and channel coding, mobile communication, and heterogeneous network. He has authored 33 books including *Digital Communication* (Youngil, 1999), 95 SCI international journal papers, and 20 patents.

Dr. Lee is a Registered Telecommunication Professional Engineer and a member of the National Academy of Engineering in Korea. He was recipient of the paper prize award from the Korea Institute of Communication Science in 1986 and 1997, the Korea Institute of Electronics Engineers in 1987, Chonbuk Province in 1992, Chonbuk National University in 2002, commendation of the Prime Minister (2002) and Science Technology Doyak Gold Medal from Korea Government (2007) for the invention of the Jacket matrix, which is widely helpful in communication signal processing, mobile networking, and cryptography. He was the Guest Editor of the *IEEE Communications Magazine*Special Issue on the Quality of Service Based Routing Algorithms for Heterogeneous Networks.

**Xiao-Dong Zhang** was born in Jiangsu, China. He received the Master's degree in pure mathematics from Anhui University, Hefei, China in 1992 and the Ph.D. degree in applied mathematics from University of Science and Technology of China, Hefei, China, in 1998.

He is a Professor in the Department of Mathematics at Shanghai JiaoTong University (SJTU), Shanghai, China. Before he joined SJTU in 2002, he took various academic positions at Universitdad de Chile, Kyungpook National University, the Technion-Israel Institute of Technology (Lady Davis Postdoctoral fellowship) and Anhui University. His research interests include combinatorics, graph theory, complex network, matrix theory and their applications in computer science, digital communication, signal processing. He has authored or coauthored over 50 refereed papers.

# Applications-Jacket matrix is useful to apply in:

## ■ Signal Processing

[M.H.Lee and J.Hou, "Fast block inverse Jacket transform,"*IEEE Signal Processing Letters*, vol.13, no.8, pp.461-464, Aug.2006.]

## ■ Encoding

[M.H.Lee and K.Finalayson, "A simple element inverse Jacket transform coding," *IEEE Signal Processing Letters*, vol. 14, no.3, March 2007.]

## ■ Mobile Communication

[X.J.Jiang, and M.H.Lee,"Higher dimensional Jacket code for mobile communications," *APWC* 2005, Sapporo, Japan,4-5,Aug.2005.]

[J.Hou and M.H.Lee, "Enhancing data throughput and lower correlations quasi orthogonal functions for 3G CDMA systems," *International Journal of Communicational Systems*, John Wiley and Sons, published online 13 Jan. 2006.]

## ■ Sequence

[M.G.Parker and Moon Ho Lee,"Optinal Bipolar Sequences for the Complex Reverse Jacket transform," *International Symposium on Information Theory and Its Applications*, Honolulu, Hawaii,USA, Nov.5-8,2000.]

## ■ Cryptography

[J.Hou, and Moon Ho Lee,"Cocyclic Jacket matrices and Its application to cryptosystems," Springer Berlin/Heidelberg, LNCS,3391,2005.]

## ■ Space Time Code

[Jia Hou, Moon Ho Lee, "Matrices Analysis of Quasi Orthogonal Space Time Block Codes," *IEEE communication. Letters*, vol.7, no, 8, Aug.2003.]

[Jia Hou, and Moon Ho Lee,"J-rotation Space Time Block Codes," *IEEE International Symposium on Information Theory*, Yokohama, Japan, pp.125,June 29-July 4, 2003.]

## References


- M.H. Lee, The Center Weighted Hadamard Transform, IEEE Trans.1989 AS-36, (9), pp.1247-1249.
- S.-R.Lee and M.H.Lee, On the Reverse Jacket Matrix for Weighted Hadamard Transform, IEEE Trans. on Circuit Syst.II, vol.45.no.1, pp.436-441,Mar.1998.
- M.H. Lee, A New Reverse Jacket Transform and its Fast Algorithm, IEEE Trans. Circuits Syst.-II , vol 47, pp.39-46, 2000.
- M.H. Lee and B.S. Rajan, A Generalized Reverse Jacket Transform, IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 48 no.7 pp 684-691, 2001.
- J. Hou, M.H. Lee and J.Y. Park, New Polynomial Construction of Jacket Transform, IEICE Trans. Fundamentals, vol. E86-A no. 3, pp.652-659, 2003.
- W.P. Ma and M. H. Lee, Fast Reverse Jacket Transform Algorithms, Electronics Letters, vol. 39 no. 18 , 2003.
- Moon Ho Lee, Ju Yong Park, and Jia Hou,Fast Jacket Transform Algorithm Based on Simple Matrices Factorization, IEICE Trans. Fundamental, vol.E88-A, no.8, Aug.2005.
- Moon Ho Lee and Jia Hou, Fast Block Inverse Jacket Transform, IEEE Signal Processing Letters, vol.13. No.8, Aug.2006.
- Jia Hou and Moon Ho Lee ,Construction of Dual OVSF Codes with Lower Correlations, IEICE Trans. Fundamentals, Vol.E89-A, No.11 pp 3363-3367, Nov 2006.
- Jia Hou , Moon Ho Lee and Kwang Jae Lee,Doubly Stochastic Processing on Jacket Matricess, IEICE Trans. Fundamentals, vol E89-A, no.11, pp 3368-3372, Nov 2006.
- Ken Finlayson, Moon Ho Lee, Jennifer Seberry, and Meiko Yamada, Jacket Matrices constructed from Hadamard Matrices and Generalized Hadamard Matrices, Australasian

■ Chang Hue Choe, M. H. Lee, Gi Yeon Hwang, Seong Hun Kim, and Hyun Seuk Yoo, Key Agreement Protocols Based on the Center Weighted Jacket Matrix as a Sysmmetric Co-cyclic Matrix, Lecture Notes in Computer Science, vol 4105, pp 121-127 Sept 2006 . Journal of Combinatorics, vol.35, pp 83-88, June 2006.

■ Moon Ho Lee, and Ken.Finlayson, A Simple Element Inverse Jacket Transform Coding, Information Theory Workshop 2005, ITW 2005, Proc. of IEEE ITW 2005, 28.Aug-1.Sept., New Zealand, also IEEE Signal Processing Letters, vol. 14 no. 5, May 2007.

■ Moon Ho Lee, X. D. Zhang, Fast Block Center Weighted Hadamard Transform, IEEE Trans. Circuits Syst., vol. 54 no.12 pp 2741-2745, Dec 2007.

■ Zhu Chen, Moon Ho Lee, Fast Cocyclic Jacket Transform, IEEE Signal Processing, vol. 15 no.5 May 2008.

■ Guihua Zeng, Moon Ho Lee, A Generalized Reverse Block Jacket Transform, Accepted IEEE Trans. Circuits Syst.-I, vol. 55 no.? July 2008.

■ Guihua Zeng, Yuan Li, Ying Guo and Moon Ho Lee, Stabilizer quantum codes over the Clifford algebra, J. Phys. A: Math. Theor. vol. 41, 2008.

Retrieved from *http://en.wikipedia.org/wiki/Jacket_matrix*, *Categories:Matrices*

# Eigenvalue Decomposition of Jacket Transform and Its Application to Alamouti Code

# Eigenvalue decomposition of Jacket matrix of order 2

Now we assume that a matrix

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

is able to Jacket eigenvalue decomposition. In other words, $A$ can be rewritten as follows:

$$A = J\Lambda J^{-1},$$

where $\Lambda = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$. By matrix multiplication and the property of Jacket matrix, we have

$$J\Lambda J^{-1} = \begin{pmatrix} d_1e_1 & d_1e_2 \\ d_2e_1 & -d_2e_2 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \frac{1}{2} \begin{pmatrix} \frac{1}{d_1e_1} & \frac{1}{d_2e_1} \\ \frac{1}{d_1e_2} & -\frac{1}{d_2e_2} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \lambda_1 + \lambda_2 & \frac{d_1}{d_2}(\lambda_1 - \lambda_2) \\ \frac{d_2}{d_1}(\lambda_1 - \lambda_2) & \lambda_1 + \lambda_2 \end{pmatrix}$$

Hence we have

$$a = \frac{1}{2}(\lambda_1 + \lambda_2), \quad b = \frac{d_1}{2d_2}(\lambda_1 - \lambda_2), \quad c = \frac{d_2}{2d_1}(\lambda_1 - \lambda_2), \quad d = \frac{1}{2}(\lambda_1 + \lambda_2)$$

**Theorem 2.1** *A $2 \times 2$ matrix $A$ is Jacket similar to the diagonal matrix if and only if $A$ has the following form*

$$A = \begin{pmatrix} a & b \\ c & a \end{pmatrix},$$

*ie., the entries of the main diagonal of a matrix are equal.*

A $2 \times 2$ symmetric Jacket matrix pattern is $J = \begin{pmatrix} d_1 & -\sqrt{d_1 d_2} \\ -\sqrt{d_1 d_2} & -d_2 \end{pmatrix}$

If $A$ is eigenvalue decomposition by the symmetric matrix, then

$$J\Lambda J^{-1} = \begin{pmatrix} d_1 & -\sqrt{d_1 d_2} \\ -\sqrt{d_1 d_2} & -d_2 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} d_1 & -\sqrt{d_1 d_2} \\ -\sqrt{d_1 d_2} & -d_2 \end{pmatrix}^{-1}$$

$$= \begin{pmatrix} d_1 & -\sqrt{d_1 d_2} \\ -\sqrt{d_1 d_2} & -d_2 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \frac{1}{2} \begin{pmatrix} \frac{1}{d_1} & -\frac{1}{\sqrt{d_1 d_2}} \\ -\frac{1}{\sqrt{d_1 d_2}} & -\frac{1}{d_2} \end{pmatrix}$$

$$= \frac{1}{2} \begin{pmatrix} \lambda_1 + \lambda_2 & \sqrt{\frac{d_1}{d_2}}(\lambda_2 - \lambda_1) \\ \sqrt{\frac{d_2}{d_1}}(\lambda_2 - \lambda_1) & \lambda_1 + \lambda_2 \end{pmatrix}.$$

Hence $\lambda_1 + \lambda_2 = 2a, \quad \sqrt{\frac{d_1}{d_2}}(\lambda_2 - \lambda_1) = 2b, \quad \sqrt{\frac{d_2}{d_1}}(\lambda_2 - \lambda_1) = 2c$

By solving these equations, we can get $\lambda_2 - \lambda_1 = \pm 2\sqrt{bc}$

Hence we have $\lambda_1 = a \pm \sqrt{bc}, \quad \lambda_2 = a \mp \sqrt{bc}$

and $d_1 = b$ and $d_2 = c$.

$$\begin{pmatrix} a & b \\ c & a \end{pmatrix} = \begin{pmatrix} b & -\sqrt{bc} \\ -\sqrt{bc} & -c \end{pmatrix} \begin{pmatrix} a \pm \sqrt{bc} & 0 \\ 0 & a \mp \sqrt{bc} \end{pmatrix} \begin{pmatrix} b & -\sqrt{bc} \\ -\sqrt{bc} & -c \end{pmatrix}^{-1}$$

For two by two symmetric matrix $A = \begin{pmatrix} a & b \\ b & a \end{pmatrix}$

$A$ can be eigenvalue decomposition by unitary symmetric Jacket pattern

$$J = \begin{pmatrix} x & x \\ x & -x \end{pmatrix}_{x=1:\, Hadamard\, matrix.}$$

$A$ can be eigenvalue decomposition as the follows:

$$\begin{pmatrix} a & b \\ b & a \end{pmatrix} = \begin{pmatrix} a & a \\ a & -a \end{pmatrix} \begin{pmatrix} a+b & 0 \\ 0 & a-b \end{pmatrix} \begin{pmatrix} a & a \\ a & -a \end{pmatrix}^{-1}$$

# Eigenvalue Decomposition of Jacket matrix of order 3

We just recall the $3 \times 3$ Jacket patterns. $J_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{pmatrix}$ where $\omega^3 = 1$ and $\omega \neq 1$.

We assume that $A = (a_{ij})$ of order 3 is Jacket similar to diagonal matrix $\Lambda$. In other words, there exists a Jacket matrix $B$ such that

$$A = B\Lambda B^{-1}$$

we may assume that $B = DJ$ and $D = diag(d_1, d_2, d_3)$. Then

$$DJ\Lambda(DJ)^{-1} = \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{pmatrix}^{-1} \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{pmatrix}^{-1}$$

$$= \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{pmatrix} \begin{pmatrix} \lambda_1 & \lambda_1 & \lambda_1 \\ \lambda_2 & \lambda_2\omega^2 & \lambda_2\omega \\ \lambda_3 & \lambda_3\omega & \lambda_3\omega^2 \end{pmatrix} \begin{pmatrix} \frac{1}{d_1} & 0 & 0 \\ 0 & \frac{1}{d_2} & 0 \\ 0 & 0 & \frac{1}{d_3} \end{pmatrix}$$

$$= \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{pmatrix} \begin{pmatrix} \lambda_1+\lambda_2+\lambda_3 & \lambda_1+\lambda_2\omega^2+\lambda_3\omega & \lambda_1+\lambda_2\omega+\lambda_3\omega^2 \\ \lambda_1+\lambda_2\omega+\lambda_3\omega^2 & \lambda_1+\lambda_2+\lambda_3 & \lambda_1+\lambda_2\omega^2+\lambda_3\omega \\ \lambda_1+\lambda_2\omega^2+\lambda_3\omega & \lambda_1+\lambda_2\omega+\lambda_3\omega^2 & \lambda_1+\lambda_2+\lambda_3 \end{pmatrix}$$

$$\begin{pmatrix} \frac{1}{d_1} & 0 & 0 \\ 0 & \frac{1}{d_2} & 0 \\ 0 & 0 & \frac{1}{d_3} \end{pmatrix} = \begin{pmatrix} \lambda_1+\lambda_2+\lambda_3 & \frac{d_1}{d_2}(\lambda_1+\lambda_2\omega^2+\lambda_3\omega) \\ \frac{d_2}{d_1}(\lambda_1+\lambda_2\omega+\lambda_3\omega^2) & \lambda_1+\lambda_2+\lambda_3 \\ \frac{d_3}{d_1}(\lambda_1+\lambda_2\omega^2+\lambda_3\omega) & \frac{d_3}{d_2}(\lambda_1+\lambda_2\omega+\lambda_3\omega^2) \end{pmatrix}.$$

$$\begin{pmatrix} \frac{d_1}{d_3}(\lambda_1+\lambda_2\omega+\lambda_3\omega^2) \\ \frac{d_2}{d_3}(\lambda_1+\lambda_2\omega^2+\lambda_3\omega) \\ \lambda_1+\lambda_2+\lambda_3 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}.$$

we have the following equations: $a_{11} = a_{22} = a_{33} = \lambda_1 + \lambda_2 + \lambda_3,$

$$a_{12}a_{21} = (\lambda_1+\lambda_2\omega^2+\lambda_3\omega)(\lambda_1+\lambda_2\omega+\lambda_3\omega^2),$$

$$a_{13}a_{31} = (\lambda_1+\lambda_2\omega^2+\lambda_3\omega)(\lambda_1+\lambda_2\omega+\lambda_3\omega^2),$$

$$a_{23}a_{32} = (\lambda_1+\lambda_2\omega^2+\lambda_3\omega)(\lambda_1+\lambda_2\omega+\lambda_3\omega^2).$$

Hence $A$ must have the following properties

$$a_{11} = a_{22} = a_{33} \quad \textit{and} \quad a_{12}a_{21} = a_{13}a_{31} = a_{23}a_{32}.$$

This explains that any matrix of order 3 which is Jacket similar to diagonal matrix must own the above two equations. In other words, $A$ must have the following the form

$$A = \begin{pmatrix} a & a_{12} & a_{13} \\ \frac{k}{a_{12}} & a & a_{23} \\ \frac{k}{a_{13}} & \frac{k}{a_{23}} & a \end{pmatrix}.$$

**Theorem 3.1** *Let $A = (a_{ij})$ is of order 3. The $A$ is Jacket similar to a diagonal matrix $\Lambda$ if and only if $A$ must be satisfied Eq. (25). In other words, such the matrix must be eigenvalue decomposition.*

For example, let

$$A = \begin{pmatrix} a & b & c \\ c & a & b \\ b & c & a \end{pmatrix}.$$

We assume that

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{pmatrix}^{-1}.$$

Then from the above equation, we are able to obtain the following equation.

$$\lambda_1 + \lambda_2 + \lambda_3 = 3a, \quad \lambda_1 + \lambda_2\omega^2 + \lambda_3\omega = 3b, \quad \lambda_1 + \lambda_2\omega + \lambda_3\omega^2 = 3c$$

By solving the above equations, we obtain

$$\lambda_1 = a + b + c, \quad \lambda_2 = a + b\omega + c\omega^2, \quad \lambda_3 = a + b\omega^2 + c\omega.$$

Hence the matrix $A$ can be decomposed as the following form:

$$\begin{pmatrix} a & b & c \\ c & a & b \\ b & c & a \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{pmatrix} \begin{pmatrix} a+b+c & 0 & 0 \\ 0 & a+b\omega+c\omega^2 & 0 \\ 0 & 0 & a+b\omega^2+c\omega \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{pmatrix}^{-1}.$$

# Eigenvalue Decomposition of Jacket matrix of order 4

**Theorem 4.1** *Any Jacket matrix of order 4 is equivalent to the following Jacket matrix.*

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -w & w & -1 \\ 1 & w & -w & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}.$$

We assume that $A$ is Jacket similar to diagonal matrix $\Lambda$.

Then $A = (WD)\Lambda(WD)^{-1}$. Hence $D^{-1}AD = W\Lambda W^{-1}$ Since

$$W\Lambda W^{-1} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -w & w & -1 \\ 1 & w & -w & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1/w & 1/w & -1 \\ 1 & 1/w & -1/w & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

$$= \frac{1}{4} \begin{pmatrix} a & b & c & d \\ f & a & d & e \\ e & d & a & f \\ d & c & b & a \end{pmatrix},$$

Where

$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = a, \quad \lambda_1 - \lambda_2/w + \lambda_3/w - \lambda_4 = b$$

$$\lambda_1 + \lambda_2/w - \lambda_3/w - \lambda_4 = c, \quad \lambda_1 - \lambda_2 - \lambda_3 + \lambda_4 = d$$

$$\lambda_1 + \lambda_2 w - \lambda_3 w - \lambda_4 = e, \quad \lambda_1 - \lambda_2 w + \lambda_3 w - \lambda_4 = f$$

Then by solving the above equations, we obtain $\lambda_1 = (1/4)(a+b+c+d)$, $\lambda_2 = (1/4)(a-bw+cw-d)$, $\lambda_3 = (1/4)(a+bw-cw-d)$, $\lambda_4 = (1/4)(a-b-c+d)$, $w = \sqrt{\frac{2e-(b+c)}{c-b}}$, $f = b+c-e$.

Now we compare with both sides of the equations $D^{-1}AD = W\Lambda W^{-1}$ which implies that $A$ must have the following form

$$A = \begin{pmatrix} a & (bd_1)/d_2 & (cd_1)/d_3 & (dd_1)/d_4 \\ (fd_2)/d_1 & a & (dd_2)/d_3 & (ed_2)/d_4 \\ (ed_3)/d_1 & (dd_3)/d_2 & a & (fd_3)/d_4 \\ (dd_4)/d_1 & (cd_4)/d_2 & (bd_4)/d_3 & a \end{pmatrix}$$

**Theorem 4.2** *Let $A$ be an matrix of order 4. Then $A$ is Jacket similar to diagonal matrix if and only if $A$ has the form Eq. (31) where $f = b + c - e$.*

For example,
$$A = \begin{pmatrix} a & b & c & d \\ f & a & d & e \\ e & d & a & f \\ d & c & b & a \end{pmatrix}$$

is a symmetric matrix and is factored into a diagonal matrix with using Jacket matrix and whose eigenvalues are easily computed.

$$\begin{pmatrix} a & b & c & d \\ b & a & d & c \\ c & d & a & b \\ d & c & b & a \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}^{-1}$$

where $\lambda_1 = a + b + c + d$, $\lambda_2 = a - b + c - d$, $\lambda_3 = a + b - c - d$, $\lambda_4 = a - b - c + d$.

$$\begin{pmatrix} a & b & c & d \\ c & a & d & b \\ b & d & a & c \\ d & c & b & a \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & i & -1 \\ 1 & i & -i & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & i & -1 \\ 1 & i & -i & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}^{-1},$$

where $\lambda_1 = a + b + c + d, \lambda_2 = a + bi - ci - d,\ \lambda_3 = a + b - c - d,\ \lambda_4 = a - b - c + d.$

$$\begin{pmatrix} 2 & -1 & 1 & 1 \\ -4 & 2 & 1 & 4 \\ 4 & 1 & 2 & -4 \\ 1 & 1 & -1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -2 & 2 & -1 \\ 1 & 2 & -2 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -2 & 2 & -1 \\ 1 & 2 & -2 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}^{-1},$$

# Eigenvalue Decomposition of Jacket matrix of order n

we use Kronecker product to construct eigenvalue decomposition of a matrix of order $2^k \times 3^l$.

For example, $\begin{pmatrix} a & b \\ b & a \end{pmatrix} = \begin{pmatrix} a & a \\ a & -a \end{pmatrix} \begin{pmatrix} a+b & 0 \\ 0 & a-b \end{pmatrix} \begin{pmatrix} a & a \\ a & -a \end{pmatrix}^{-1}$ and

$$\begin{pmatrix} a & b & c \\ c & a & b \\ b & c & a \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{pmatrix} \begin{pmatrix} a+b+c & 0 & 0 \\ 0 & a+b\omega+c\omega^2 & 0 \\ 0 & 0 & a+b\omega^2+c\omega \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{pmatrix}^{-1}.$$

We can construct matrix $A$ of order 6 can have eigenvalue decomposition

$$\begin{pmatrix} a & b \\ b & a \end{pmatrix} \otimes \begin{pmatrix} a & b & c \\ c & a & b \\ b & c & a \end{pmatrix} = \left( \begin{pmatrix} a & a \\ a & -a \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{pmatrix} \right)$$

$$\left( \begin{pmatrix} a+b & 0 \\ 0 & a-b \end{pmatrix} \right) \otimes \begin{pmatrix} a+b+c & 0 & 0 \\ 0 & a+b\omega+c\omega^2 & 0 \\ 0 & 0 & a+b\omega^2+c\omega \end{pmatrix} \left( \begin{pmatrix} a & a \\ a & -a \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{pmatrix} \right)^{-1}$$

In general, we assume that $$A_1 = J_1 \Lambda_1 J_1^{-1},$$

where $A_1$ of order 2 has eigenvalue decomposition with Jacket matrix $J_1$.

$$A_2 = J_2 \Lambda_2 J_2^{-1},$$

where $A_2$ of order 3 has eigenvalue decomposition with Jacket matrix $J_2$. Then we have

$$A_1 \otimes A_2 = (J_1 \otimes J_2)(\Lambda_1 \otimes \Lambda_2)(J_1 \otimes J_2)^{-1}$$

With the aid of Kronecker[1, 25], the eigenvalue decomposition of $A_{p^n}(p = 2, 3)$ based on $J_{p^n}$ is given as follow:

$$A_{p^n} = J_{p^n}^{-1} \Lambda_{p^n} J_{p^n},$$

where $A_{p^n} = A_{p^{n-1}} \otimes A_p = A_p^{\otimes n}$, $J_{p^n} = J_{p^{n-1}} \otimes J_p = J_p^{\otimes n}$, and $\Lambda_{p^n} = \Lambda_{p^{n-1}} \otimes \Lambda_p = \Lambda_p^{\otimes n}$. In general, assuming $A_{2^m} = J_{2^m} \Lambda_{2^m} J_{2^m}^{-1}$ and $A_{3^n} = J_{3^n} \Lambda_{3^n} J_{3^n}^{-1}$, we can get

$$A_{2^m \times 3^n} = A_{2^m} \otimes A_{3^n} = (J_{2^m} \Lambda_{2^m} J_{2^m}^{-1}) \otimes (J_{3^n} \Lambda_{3^n} J_{3^n}^{-1}) = (J_{2^m} \otimes J_{3^n})(\Lambda_{2^m} \otimes \Lambda_{3^n})(J_{2^m} \otimes J_{3^n})^{-1}.$$

For example, we have

$$J_{12} = J_{2^2} \otimes J_3 = (J_2 \otimes J_2) \otimes J_3 \quad = (J_2 \otimes I_2 \otimes I_3)(I_2 \otimes J_2 \otimes I_3)(I_2 \otimes I_2 \otimes J_3),$$

$J_{12}$ matrix is given as

$$J_{12} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & 1 & \omega & \omega^2 & 1 & \omega & \omega^2 & 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega & 1 & \omega^2 & \omega & 1 & \omega^2 & \omega & 1 & \omega^2 & \omega \\ 1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & \omega & \omega^2 & -1 & -\omega & -\omega^2 & 1 & \omega & \omega^2 & -1 & -\omega & -\omega^2 \\ 1 & \omega^2 & \omega & -1 & -\omega^2 & -\omega & 1 & \omega^2 & \omega & -1 & -\omega^2 & -\omega \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & \omega & \omega^2 & 1 & \omega & \omega^2 & -1 & -\omega & -\omega^2 & -1 & -\omega & -\omega^2 \\ 1 & \omega^2 & \omega & 1 & \omega^2 & \omega & -1 & -\omega^2 & -\omega & -1 & -\omega^2 & -\omega \\ 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & -1 & -\omega & -\omega^2 & -1 & -\omega & -\omega^2 & 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega & -1 & -\omega^2 & -\omega & -1 & -\omega^2 & -\omega & 1 & \omega^2 & \omega \end{pmatrix} =$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \omega & \omega^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \omega^2 & \omega & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \omega & \omega^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \omega^2 & \omega & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \omega & \omega^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \omega^2 & \omega & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \omega & \omega^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \omega^2 & \omega \end{pmatrix}.$$

**TABLE 1**

Computation Complexity of The Fast Algorithm Based on Block Jacket Matrices.

| | DCA | JA FOR $N = p^k$ | JA FOR $N = p^m q^n$ |
|---|---|---|---|
| ADD | $(N-1)N$ | $(p-1)Nk$ | $mN(p-1) + nN(q-1)$ |
| MUL | $N^2$ | $\frac{(p-1)^2}{p}Nk$ | $\frac{mN}{p}(p-1)^2 + \frac{nN}{q}(q-1)^2$ |

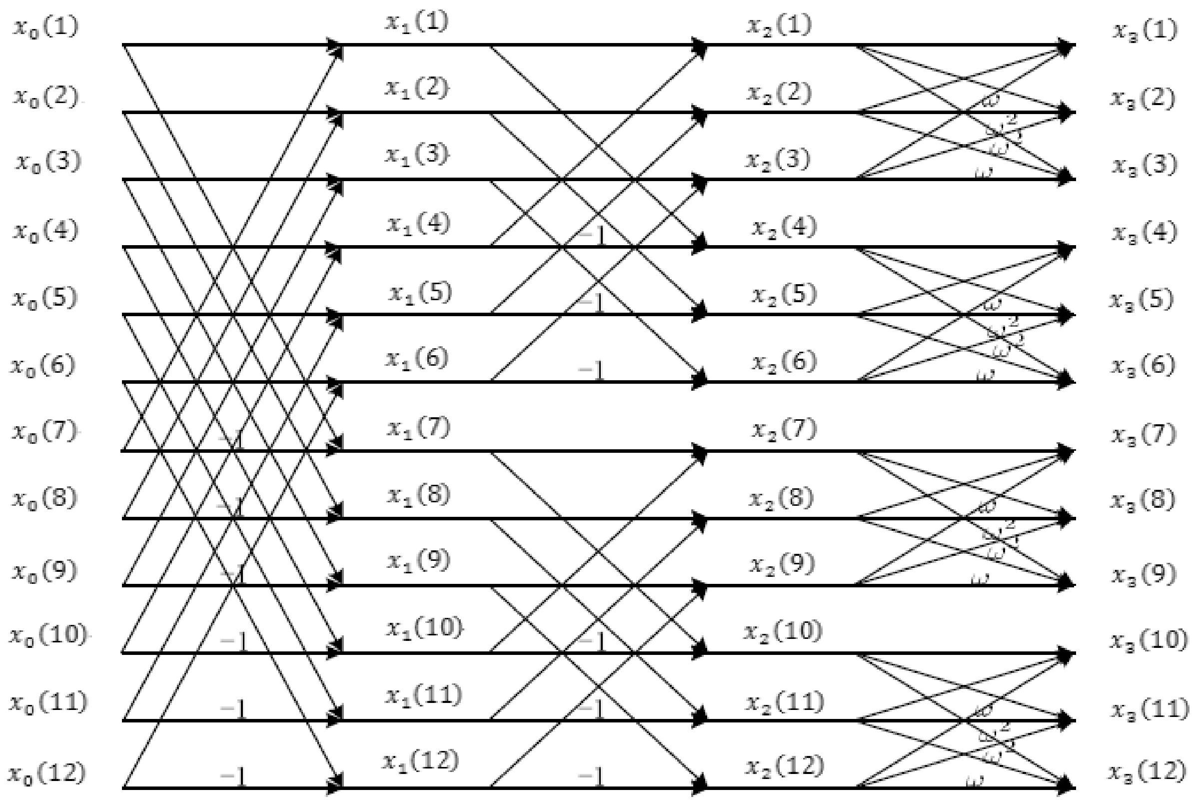



Figure 1: Fast $J_{12}$ signal flow graph, $N = 12$ .

# Cooperative Relaying in Alamouti Code Analysis Based on Jacket Matrices

Alamouti code is proposed to cooperative relaying in downlink for IEEE 802.16$j$ , a single time-frequency resource within a frame is assigned to one RS(Relay station) in relay downlink to MS(Mobile station) as shown in Fig.2.

This diversity gain can be accomplished in 3 ways:

i). cooperative source diversity, where the same signal is transmitted from different sources;

ii). cooperative transmit diversity, where the signal is space-time coded and transmitted from different sources;

iii). cooperative hybrid diversity, which is a hybrid mixture of source and transmit diversity.

For example, for 2 transmit antennas using Alamouti Matrix , encoding follows the coding scheme. That is, it represents the operation $[x_1, x_2] \rightarrow [x_1, -x_2^*]$ for antenna $\sharp 0$, $[x_2, x_1^*]$ for antenna $\sharp 1$. For 4 transmit antennas using Matrix , encoding follows the coding scheme. That is, it represents the operation $[x_1, x_2, x_3, x_4] \rightarrow [x_1, -x_2^*, 0, 0]$ for antenna $\sharp 0$, $[x_2, x_1^*, 0, 0]$ for antenna $\sharp 1$, $[0, 0, x_3, -x_4^*]$ for antenna $\sharp 3$, $[0, 0, x_4, x_3^*]$ for antenna $\sharp 1$, with rate 1.

The most important Alamouti code matrix

$$C = \begin{pmatrix} x_1 & x_2 \\ -x_2^* & x_1^* \end{pmatrix}. \tag{46}$$

Now we discuss how to factor this matrix by using Jacket matrix. We assume that Eq. (46) can be factored into $J_1 \Lambda J_2$, where $J_1$ and $J_2$ are Jacket matrices. We assume that

$$J_1 = \begin{pmatrix} a & a \\ b & -b \end{pmatrix}, J_2 = \begin{pmatrix} c & d \\ c & -d \end{pmatrix}, \Lambda = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix},$$

then

$$J_1 \Lambda J_2 = \begin{pmatrix} ac(\lambda_1 + \lambda_2 & ad(\lambda_1 - \lambda_2) \\ bc(\lambda_1 - \lambda_2) & bd(\lambda_1 + \lambda_2) \end{pmatrix}.$$

By comparing with both sides of $C = J_1 \Lambda J_2$, it is easy to get the following equations,

$$ac(\lambda_1 + \lambda_2) = x_1, \quad ad(\lambda_1 - \lambda_2) = x_2, \quad bc(\lambda_1 - \lambda_2) = -x_2^*, \quad bd(\lambda_1 + \lambda_2) = x_1^*.$$

By solving the above equations, we have

$$\frac{c(\lambda_1 + \lambda_2)}{d(\lambda_1 - \lambda_2)} = \frac{x_1}{x_2}, \quad \frac{c(\lambda_1 - \lambda_2)}{d(\lambda_1 + \lambda_2)} = \frac{-x_2^*}{x_1^*}.$$

Hence by

$$(\frac{c}{d})^2 = -\frac{x_1 x_2^*}{x_2 x_1^*} = \frac{-x_1^2 (x_2^*)^2}{x_1 x_1^* x_2 x_2^*}$$

We can put $c = \frac{i x_1}{|x_1|}$, $d = \frac{x_2}{|x_2|}$, Similarly, we have $\frac{a(\lambda_1+\lambda_2)}{b(\lambda_1-\lambda_2)} = -\frac{x_1}{x_2^*}$, $\frac{a(\lambda_1-\lambda_2)}{b(\lambda_1+\lambda_2)} = \frac{x_2}{x_1^*}$. Therefore, we have $a = \frac{i x_1}{|x_1|}$, $b = \frac{x_2^*}{|x_2|}$. So $-\frac{x_1^2}{|x_1|^2}(\lambda_1 + \lambda_2) = x_1$, $\frac{i x_1 x_2}{|x_1||x_2|}(\lambda_1 - \lambda_2) = x_2$.

Solving the equations, we obtain $\lambda_1$ and $\lambda_2$.

$$\begin{pmatrix} x_1 & x_2 \\ -x_2^* & x_1^* \end{pmatrix} = \begin{pmatrix} a & a \\ b & -b \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} c & d \\ c & -d \end{pmatrix} = J_1 \Lambda J_2.$$

Consider $4 \times 4$ Alamouti matrix with rate $1/2$ , we are able to find that

$$\begin{pmatrix} x_1 & x_2 & 0 & 0 \\ -x_2^* & x_1^* & 0 & 0 \\ 0 & 0 & x_1 & x_2 \\ 0 & 0 & -x_2^* & x_1^* \end{pmatrix} = \begin{pmatrix} a & a & 0 & 0 \\ b & -b & 0 & 0 \\ 0 & 0 & a & a \\ 0 & 0 & b & -b \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_1 & 0 \\ 0 & 0 & 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} c & d & 0 & 0 \\ c & -d & 0 & 0 \\ 0 & 0 & c & d \\ 0 & 0 & c & -d \end{pmatrix}$$

$$= I_2 \otimes (J_1 \Lambda J_2) = (I_2 \otimes J_1)(I_2 \otimes \Lambda)(I_2 \otimes J_2).$$

and

$$\begin{pmatrix} x_1 & x_2 & x_1 & x_2 \\ -x_2^* & x_1^* & -x_2^* & x_1^* \\ 0 & 0 & x_1 & x_2 \\ 0 & 0 & -x_2^* & x_1^* \end{pmatrix} = \begin{pmatrix} J_1 & 0 \\ 0 & J_1 \end{pmatrix} \begin{pmatrix} \Lambda & \Lambda \\ 0 & \Lambda \end{pmatrix} \begin{pmatrix} J_2 & 0 \\ 0 & J_2 \end{pmatrix}$$

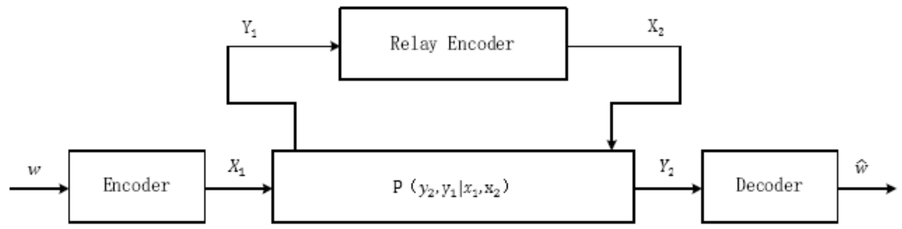
four symbols Alamouti matrix given

$$\begin{pmatrix} x_1 & x_2 & 0 & 0 \\ -x_2^* & x_1^* & 0 & 0 \\ 0 & 0 & x_3 & x_4 \\ 0 & 0 & -x_4^* & x_3^* \end{pmatrix} = \begin{pmatrix} a_1 & a_1 & 0 & 0 \\ b_1 & -b_1 & 0 & 0 \\ 0 & 0 & a_2 & a_2 \\ 0 & 0 & b_2 & -b_2 \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{pmatrix} \begin{pmatrix} c_1 & d_1 & 0 & 0 \\ c_1 & -d_1 & 0 & 0 \\ 0 & 0 & c_2 & d_2 \\ 0 & 0 & c_2 & -d_2 \end{pmatrix},$$

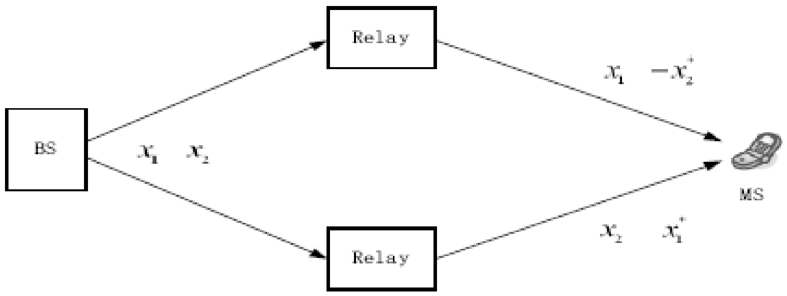
where
$$a_1 = c_1 = \frac{ix_1}{|x_1|}, \quad b_1 = \frac{x_2^*}{|x_2|}, \quad d_1 = \frac{x_2}{|x_2|}, \quad a_2 = c_2 = \frac{ix_3}{|x_3|}, \quad b_2 = \frac{x_4^*}{|x_4|}, \quad d_2 = \frac{x_4}{|x_4|},$$

$$-\frac{x_1^2}{|x_1|^2}(\lambda_1+\lambda_2) = x_1, \quad \frac{ix_1x_2}{|x_1||x_2|}(\lambda_1-\lambda_2) = x_2, \quad -\frac{x_3^2}{|x_3|^2}(\lambda_3+\lambda_4) = x_3 \quad and \quad \frac{ix_3x_4}{|x_3||x_4|}(\lambda_3-\lambda_4) = x_4.$$

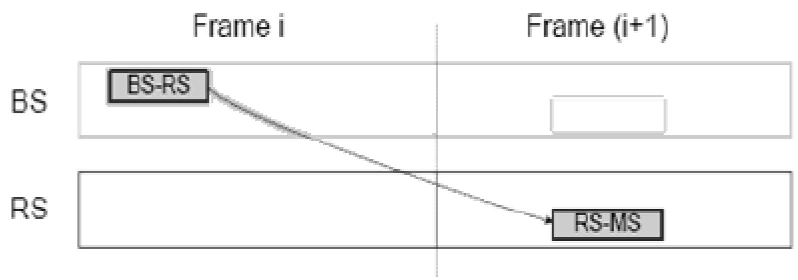
eigenvector is block-wise diagonal, however, eigenvalue is linearly diagonal, depend on signal frame relay $i$ and $i+1$, as Fig.2(c).



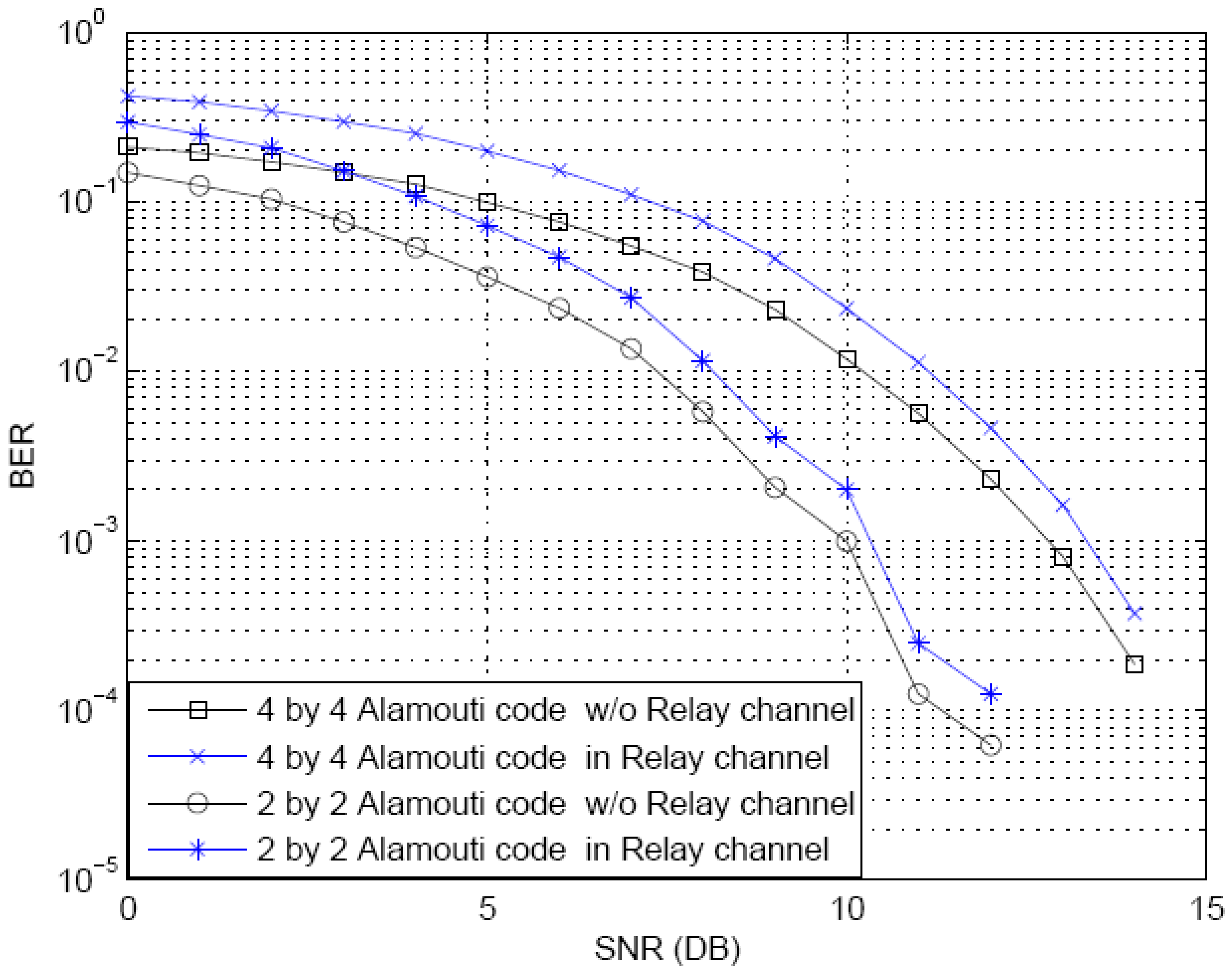

(a) Relay Channel




(b) Alamouti in Relay Channel




(c) Frame in Relay Channel




(d) Simulation

# On Jacket Matrices Based on Center Weighted Hadamard Matrices

# OUTLINE

- Hadamard Matrices
- Jacket Matrices Definition and Examples
- Jacket Matrices of Small Orders
- Construction of Jacket Matrices

An $n \times n$ matrix whose entries are $+1$ or $-1$ is called a Hadamard matrix if

$$HH^T = nI_n,$$

where $T$ denote transpose of $H$, $I_n$ identity matrix.

- For example:

$$H = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad HH^T = 2I_2.$$

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \quad HH^T = 4I_4.$$

## Center Weighted Hadamard Matrices

$$W_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -2 & 2 & -1 \\ 1 & 2 & -2 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \qquad W_4^{\dagger} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -\frac{1}{2} & \frac{1}{2} & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

i.e. transpose matrix of elements inverse of W.

Clearly by a simple calculation,

$$W_4 W_4^{\dagger} = 4I_4.$$

In particularly, if w=1, it is a Hadamard matrix and if w=2 it is a special Center Weighted Hadamard Matrix.

and $$W_{2n} = W_n \otimes H_2, where\, H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Furthermore, there exists a permutation matrix *P* (each row and column of *P* has exactly 1) such that

$$PW_{2n}P^T = P\left(W_n \otimes H_2\right)P^T = H_2 \otimes W_n,$$

where $P^T$ is the transpose matrix of *P*. Hence,

$$W_{2n}W_{2n}^{\dagger} = P^T\left(H_2 \otimes W_n\right)PP^T\left(H_2 \otimes W_n\right)P$$

$$= P^T\begin{pmatrix} W_n & W_n \\ W_n & W_n \end{pmatrix}\begin{pmatrix} W_n & W_n \\ W_n & W_n \end{pmatrix}^{\dagger} P = 2nI_{2n}.$$

$$WW^{\dagger} = 2nI_{2n}, \quad W^{-1} = \frac{1}{2n}W^{\dagger}$$

■ Turyn-type Hadamard Matrices

An $n \times n$ matrix $H$ whose entries ±1, ±i ($i^2$=-1) is called Turyn-type Hadamard matrix, if

$$HH^* = nI_n,$$

where * denote conjugate transpose.

■ Example:

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & i^2 & i^3 \\ 1 & i^2 & i^4 & i^6 \\ 1 & i^3 & i^6 & i^9 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}, \quad A^\dagger = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{i} & \frac{1}{i^2} & \frac{1}{i^3} \\ 1 & \frac{1}{i^2} & \frac{1}{i^4} & \frac{1}{i^6} \\ 1 & \frac{1}{i^3} & \frac{1}{i^6} & \frac{1}{i^9} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}$$

$$AA^{\dagger} = AA^{*} = 4I_4$$

Hence $A$ is a Turyn-type Hadamard matrix and also a DFT matrix, but not a real Hadamard matrix.

■ An $n \times n$ matrix $H$ whose entries are power of $q$-th root of unity is called a Butson-type Hadamard matrix if

$$HH^{*} = nI_n$$

For example, let $\omega$ is a third root of unity, i.e. $w = e^{2\pi i/3}, i^2 = -1$ and

$$B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & w & w^2 \\ 1 & w^2 & w \end{pmatrix}, \quad B^{-1} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \frac{1}{w} & \frac{1}{w^2} \\ 1 & \frac{1}{w^2} & \frac{1}{w} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & w^2 & w \\ 1 & w & w^2 \end{pmatrix} \text{ so } BB^{\dagger} = BB^{*} = 3I_3$$

## Complex Hadamard Matrices

An n x n matrix $A=(a_{jk})$ is called a complex Hadamard matrix if

$$|a_{jk}| = 1, \qquad AA^* = nIn$$

$$C = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & w & w^2 & w^2 & w & 1 \\ i & iw & iw^2 & -iw^2 & -iw & -i \\ 1 & w^2 & w & w & w^2 & 1 \\ i & iw^2 & iw & -iw & -iw^2 & -i \\ i & i & i & -i & -i & -i \end{pmatrix}$$

$$C^\dagger = C^* = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \frac{1}{w} & \frac{1}{iw} & \frac{1}{w^2} & \frac{1}{iw^2} & \frac{1}{i} \\ 1 & \frac{1}{w^2} & \frac{1}{iw^2} & \frac{1}{w} & \frac{1}{iw} & \frac{1}{i} \\ 1 & \frac{1}{w^2} & -\frac{1}{iw^2} & \frac{1}{w} & -\frac{1}{iw} & -\frac{1}{i} \\ 1 & \frac{1}{w} & -\frac{1}{iw} & \frac{1}{w^2} & -\frac{1}{iw^2} & -\frac{1}{i} \\ 1 & 1 & -\frac{1}{i} & 1 & -\frac{1}{i} & -\frac{1}{i} \end{pmatrix}$$

## Jacket Matrix Definition and Examples

Let $A = (a_{jk})$ be an $n \times n$ matrix whose elements are in a Field $F$ (including real fields, complex fields and finite fields, etc). Denoted by $A^\dagger$ the transpose matrix of elements inverse $A$.ie. $A^\dagger = \left(a_{kj}^{-1}\right)$. $A$ is called Jacket matrix if

$$AA^\dagger = A^\dagger A = nI_n,$$

where $I_n$ is the identity matrix over field $F$.

- For example:

$$A = \begin{pmatrix} a & \sqrt{ac} \\ \sqrt{ac} & -c \end{pmatrix}, A^{-1} = \begin{pmatrix} \frac{1}{a} & \frac{1}{\sqrt{ac}} \\ \frac{1}{\sqrt{ac}} & -\frac{1}{c} \end{pmatrix},$$

So A is a 2x2 Jacket matrix. When a=c=1, it is a 2x2 Hadamard matrix.

- The class of Jacket matrices contains
  - Real Hadamard Matrices;
  - Turyn-type Hadamard matrices;
  - Butson-Type Hadamard matrices
  - Complex Hadamard matrices;
  - Center Weighted Hadamard matrices.

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & -2 & 2 & -2 \\ 3 & 3 & -3 & -3 \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix}, A^{-1} = \frac{1}{4}\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & 2 \\ 1 & -\frac{1}{2} & \frac{1}{3} & -2 \\ 1 & \frac{1}{2} & -\frac{1}{3} & -2 \\ 1 & -\frac{1}{2} & -\frac{1}{3} & 2 \end{pmatrix}.$$

Any pair of rows of $A$ are orthogonal and $A$ is a Jacket matrix since $AA^{\dagger} = 4I_4$ but it is not a real Hadamard matrix.

## Properties of Jacket Matrices

An $n \times n$ matrix $A = (a_{ij})$ over a field $F$ is a Jacket matrix if and only if

$$for\,all\ j \neq k \qquad \sum_{i=1}^{n} \frac{a_{ji}}{a_{ki}} = 0$$

$$for\,all\ j \neq k \qquad \sum_{i=1}^{n} \frac{a_{ji}}{a_{ik}} = 0$$

For any integer n, there exists at least a Jacket matrix of order n.
There exists a Jacket matrix of any order.

- Let $A = (a_{jk})$ be an $n \times n$ Jacket matrix.

(I)If $|a_{jk}| = 1 \ for\ all\ j,k = 1,\ldots,n$ then is a complex Hadamard matrix.

(II)If $a_{jk}$ is real and, $a_{jk}^2 = 1 \ for\ all\ j,k = 1,\ldots,n$ then A is a real Hadamard matrix.

■ Let $A$ be a Jacket matrix.

(I) Then $A^{\dagger}, A^{-1} \ and\ A^{T}$ are also Jacket matrices.

(II) $(\det A)(\det A^{\dagger}) = n^{n}$.

■ Let A be nxn Jacket Matrix and let D and E be diagonal matrices. Then DAE is also a Jacket matrix.

■ Let A be an n x n Jacket matrix and let P and Q be n x n permutation matrices. Then PAQ is also a Jacket matrix

## Jacket matrices of small orders

Theorem : (1) Any Jacket matrix $A$ of order 2 is equivalent to the following matrix

$$J_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

(2) Any Jacket matrix $A$ of order 3 is equivalent to the following Jacket matrix

$$J_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & w & w^2 \\ 1 & w^2 & w \end{pmatrix}.$$

Proof: Let $A = \begin{pmatrix} 1 & 1 \\ 1 & a_{22} \end{pmatrix}$ be a normalized Jacket matrix, then $1+a_{22} = 0$.

Let

$$B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & b_{22} & b_{23} \\ 1 & b_{32} & b_{33} \end{pmatrix} \quad B^{-1} = \frac{1}{3}\begin{pmatrix} 1 & 1 & 1 \\ 1 & \dfrac{1}{b_{22}} & \dfrac{1}{b_{32}} \\ 1 & \dfrac{1}{b_{32}} & \dfrac{1}{b_{33}} \end{pmatrix}$$

be a normalized Jacket matrix and its inverse matrix and from properties of Jacket matrix, we have

$$1 + b_{22} + b_{23} = 0, \qquad 1 + b_{32} + b_{33} = 0, \qquad 1 + \frac{1}{b_{32}} + \frac{1}{b_{33}} = 0,$$

$$1 + \frac{1}{b_{22}} + \frac{1}{b_{23}} = 0, \qquad 1 + \frac{b_{22}}{b_{32}} + \frac{b_{23}}{b_{33}} = 0, \qquad 1 + \frac{b_{32}}{b_{22}} + \frac{b_{33}}{b_{23}} = 0.$$

Theorem: Any Jacket matrix of order 4 is equivalent to the following Jacket matrix.

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -w & w & -1 \\ 1 & w & -w & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}.$$

if $w = 1$, then it is a real Hadamard matrix;

if $w = i$, then it is a Turyn-type Hadamard matrix;

If $w = 2$,then it is a Center Weighted Hadamard matrix.

- Jacket matrices of order 2, 3, and 4 are unique under equivalent relationship.

- It is nature to ask whether Jacket matrices of order 5 are unique.

$\varphi = e^{2\pi i/5}$,

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \varphi & \varphi^2 & \varphi^3 & \varphi^4 \\ 1 & \varphi^2 & \varphi^4 & \varphi & \varphi^3 \\ 1 & \varphi^3 & \varphi & \varphi^4 & \varphi^2 \\ 1 & \varphi^4 & \varphi^3 & \varphi^2 & \varphi \end{pmatrix}, \quad A^\dagger = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & \varphi^4 & \varphi^3 & \varphi^2 & \varphi \\ 1 & \varphi^3 & \varphi & \varphi^4 & \varphi^2 \\ 1 & \varphi^2 & \varphi^4 & \varphi & \varphi^3 \\ 1 & \varphi & \varphi^2 & \varphi^3 & \varphi^4 \end{pmatrix},$$

$$AA^\dagger = 5I_5.$$

Let $a = \frac{-5+\sqrt{5}}{2}$

$$B = \begin{pmatrix} a+1 & 1 & 1 & 1 & 1 \\ 1 & a+1 & 1 & 1 & 1 \\ 1 & 1 & a+1 & 1 & 1 \\ 1 & 1 & 1 & a+1 & 1 \\ 1 & 1 & 1 & 1 & a+1 \end{pmatrix}$$

$$= aI_5 + T,$$

where $T$ is a matrix of all one.

$$BB\dagger = (aI_5 + T)(\frac{-a}{a+1}I_5 + T)$$

$$= \frac{-a^2}{a+1}\ I_5 + \frac{a^2+5a+5}{a+1}\ T = 5I_5.$$

$A$ and $B$ are Jacket matrix of order 5.
$A$ is not equivalent to $B$.

There are at least two Jacket matrices of any order $n \geq 5$ which are not equivalent.

## Construction of Jacket Matrices

Proposition 1: The Kronecker product of two Jacket matrices is also a Jacket matrix.

Proof:

Let A be an $m \times m$ Jacket matrix and B be an $n \times n$ Jacket matrix. Then $AA^{\dagger} = mI_m$ and $BB^{\dagger} = nI_n$. Clearly $(A \otimes B)^{\dagger} = A^{\dagger} \otimes B^{\dagger}$. Hence

$$(A \otimes B)(A \otimes B)^{\dagger} = (A \otimes B)(A^{\dagger} \otimes B^{\dagger}) = (AA^{\dagger}) \otimes (BB^{\dagger}) = mnI_{mn}.$$

So $A \otimes B$ is a Jacket matrix.

Proposition 2: Let $A$ and $B$ be two $n \times n$ Jacket matrices. Then

$$\begin{pmatrix} A & \lambda B \\ A & -\lambda B \end{pmatrix}$$

is also a Jacket matrices of order 2n, where $\lambda \neq 0$.

Example : Let

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & w & w^2 \\ 1 & w^2 & w \end{pmatrix}, B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & w^2 & w \\ 1 & w & w^2 \end{pmatrix}.$$

*if* $\lambda = 1$, then

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & w & w^2 & 1 & w^2 & w \\ 1 & w^2 & w & 1 & w & w^2 \\ 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & w & w^2 & -1 & -w^2 & -w \\ 1 & w^2 & w & -1 & -w & -w^2 \end{pmatrix}$$

is a Jacket matrix of order 6. If $\lambda$=2, then

$$
A = \begin{pmatrix}
1 & 1 & 1 & 2 & 2 & 2 \\
1 & w & w^2 & 2 & 2w^2 & 2w \\
1 & w^2 & w & 2 & 2w & 2w^2 \\
1 & 1 & 1 & -2 & -2 & -21 \\
1 & w & w^2 & -2 & -2w^2 & -2w \\
1 & w^2 & w & -2 & -2w & -2w^2
\end{pmatrix}
$$

is also a Jacket matrix.

Theorem : Let $A_1, B_1, C_1$ and $D_1$ be the core of Jacket matrices of $A, B, C, D$ of order n, respectively. Then $AC^\dagger = BD^\dagger$ if and only if

$$X=\begin{pmatrix}1 & e^T & e^T & 1\\ e & A_1 & B_1 & e\\ e & C_1 & -D_1 & -e\\ 1 & e^T & -e^T & -1\end{pmatrix}$$

is a Jacket matrix of order 2n , where e is a column vector of all one.

Proof: Let $A=\begin{pmatrix}1 & e^T\\ e & A_1\end{pmatrix}$ is a Jacket matrix,

$$\begin{pmatrix}1 & e^T\\ e & A_1\end{pmatrix}\begin{pmatrix}1 & e^T\\ e & A_1^\dagger\end{pmatrix}=\begin{pmatrix}1 & e^T\\ e & A_1^\dagger\end{pmatrix}\begin{pmatrix}1 & e^T\\ e & A_1\end{pmatrix}=nI_n.$$

Hence,

$$e^T+e^TA_1=0,(A_1+I)e=0,ee^T+A_1A_1^\dagger=nI_{n-1}.$$

Similarly, we have

$$e^T + e^T B_1 = 0, \left(B_1 + I\right)e = 0, ee^T + B_1 B_1^\dagger = nI_{n-1}.$$

$$e^T + e^T C_1 = 0, \left(C_1 + I\right)e = 0, ee^T + C_1 C_1^\dagger = nI_{n-1}.$$

$$e^T + e^T D_1 = 0, \left(D_1 + I\right)e = 0, ee^T + D_1 D_1^\dagger = nI_{n-1}.$$

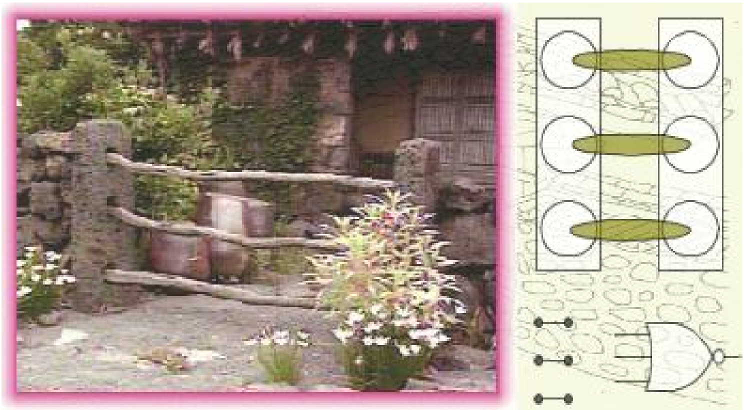
So

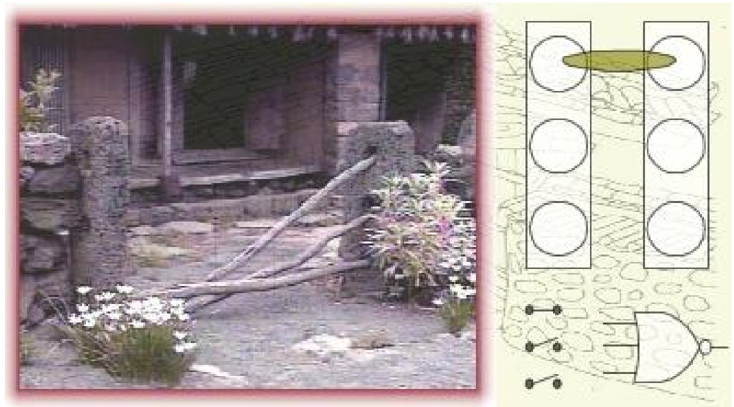
$$AC^\dagger = \begin{pmatrix} 1 & e^T \\ e & A_1 \end{pmatrix} \begin{pmatrix} 1 & e^T \\ e & C_1^\dagger \end{pmatrix} = \begin{pmatrix} n & 0 \\ 0 & ee^T + A_1 C_1^\dagger \end{pmatrix}$$

$$BD^\dagger = \begin{pmatrix} 1 & e^T \\ e & B_1 \end{pmatrix} \begin{pmatrix} 1 & e^T \\ e & D_1^\dagger \end{pmatrix} = \begin{pmatrix} n & 0 \\ 0 & ee^T + B_1 D_1^\dagger \end{pmatrix}.$$

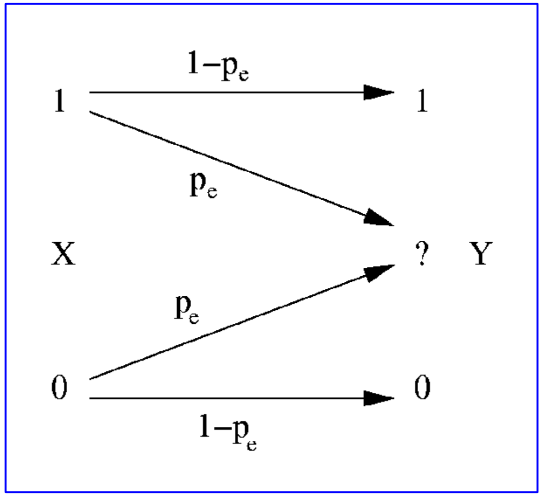
Therefore, $AC^\dagger = BD^\dagger$ if and only if $A_1 C_1^\dagger = B_1 D_1^\dagger$.

# Binary erasure Channel



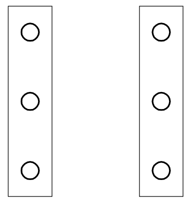
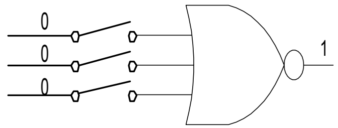
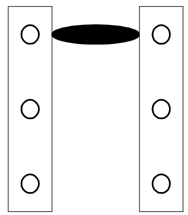
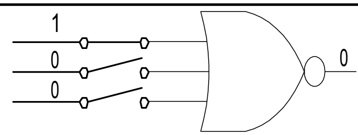
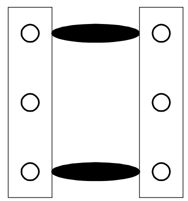
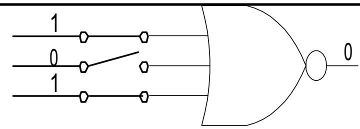
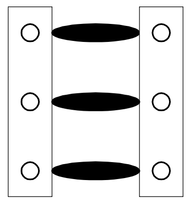
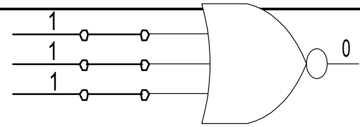


# Binary erasure channel



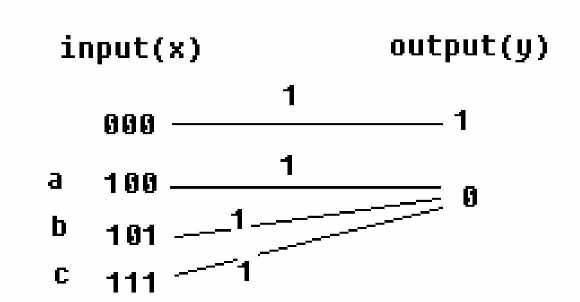

- A **binary erasure channel with erasure probability *p*** is a channel with binary input, ternary output, and probability of erasure *p*. That is, let *X* be the transmitted random variable with alphabet {0, 1}. Let *Y* be the received variable with alphabet {0, 1, *e*}, where *e* is the erasure symbol. Then, the channel is characterized by the conditional probabilities
- Pr( $Y = 0 \mid X = 0) = 1\text{-}p$
- Pr( $Y = e \mid X = 0) = p$
- Pr( $Y = 1 \mid X = 0) = 0$
- Pr( $Y = 0 \mid X = 1) = 0$
- Pr( $Y = e \mid X = 1) = p$
- Pr( $Y = 1 \mid X = 1) = 1\text{-}p$

# Jong Nang Binary Code

| □□□□□ | □□ □□□ □□ | □□□(digit) | □□ □□□/□□□□□ | □□□ □□ |
|---|---|---|---|---|
|  | □□ □□□ □□ (□□3□ □□ □□ □□ □□) | 000 |  | □(□) |
|  | □□ □□ □ (□ □□ □ □□ □□ □□ □□) | 100 |  | □(□) |
|  | □□ □□□ □□□(□, □ □□□ □□ □□) | 101 |  | □(□) |
|  | □□□ □□ □□□(□□ □□□□□□) | 111 |  | □(□) |

# Example 1

- *P* (y=0|x=100)=1 *P* (y=0|x=101)=1
  *P* (y=0|x=111)=1 *P* (y=1|x=000)=1




To calculate capacity:

C=max ***I*** (x;y)=max{*H* (y)- *H* (y|x)}

Let *P* (x=100)=a, *P* (x=101)=b, *P* (x=111)=c, *P* (x=000)=1-(a+b+c)

*P* (y=0)=a+b+c, *P* (y=1)=1-(a+b+c)

$$H(Y) = -\sum_{j=1}^{m} p(y_i)\log_2 p(y_i)$$

***H* (y)**=-*P* (y=0)*log *P* (y=0)- *P* (y=1)*log *P* (y=1)

= -q*logq-(1-q)*log(1-q) where q=a+b+c

$$H(Y|X) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i, y_j)\log_2 p(y_j|x_i)$$

Since $\log_2 p(y_j|x_i) = 0$ ***H* (y|x)**=0.

# Example 1

- ***I*** (x;y)=*H* (y)- *H* (y|x)= *H* (y)= -q*logq-(1-q)*log(1-q)

$$\frac{dI(x;y)}{dq} = -\log q - q\frac{1}{q\ln 2} + \log(1-q) + (1-q)\frac{1}{(1-q)\ln 2} = \log(\frac{1-q}{q})$$

$$\log(\frac{1-q}{q}) = 0 \Rightarrow q = \frac{1}{2}$$

***I*** (x;y)=*H* (y)- *H* (y|x)= *H* (y)= -q*logq-(1-q)*log(1-q)=1

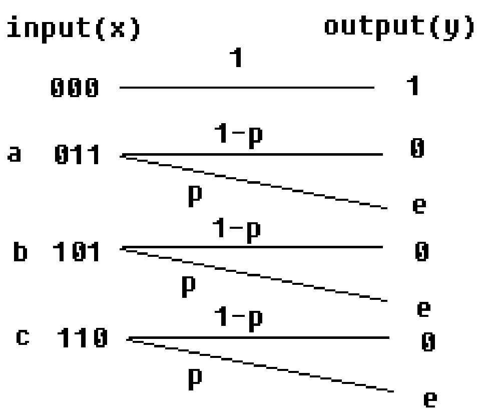
Thus C=max ***I*** (x;y)=1

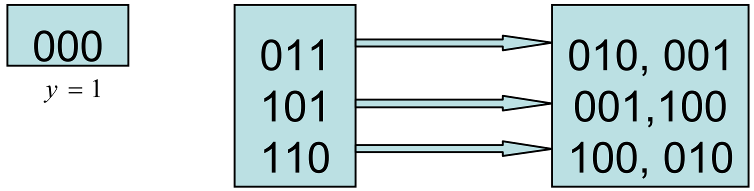
***Formula:***

$$\frac{d[u(x)v(x)]}{dx} = v(x)\frac{du(x)}{dx} + u(x)\frac{dv(x)}{dx}$$

$$\frac{d\log_2 x}{dx} = \frac{d\frac{\ln x}{\ln 2}}{dx} = \frac{1}{\ln 2}\frac{d\ln x}{dx} = \frac{1}{\ln 2}\frac{1}{x}$$

# Example 2




Assumption: At most one error occurs from "1" to "0"

Let *P* (x'=010 or 001|x=011)=p
*P* (x'=001 or 100|x=101)=p
*P* (x'=100 or 010|x=110)=p

*P* (x=011|x=011)=1-p
*P* (x=101|x=101)=1-p
*P* (x=110|x=110)=1-p

*P* (y=1)=1-(a+b+c)=1-q
*P* (y=e)=p*(a+b+c)=pq
*P* (y=0)=(1-p)(a+b+c)=(1-p)q
where q=a+b+c

$$H(Y) = -\sum_{j=1}^{m} p(y_i)\log_2 p(y_i)$$
$$= -(1-q)\log_2(1-q) - pq\log_2 pq - (1-p)q\log(1-p)q$$

*P* (x=000,y=1)=1-q
*P* (x=011,y=e)=ap
*P* (x=101,y=e)=bp
*P* (x=110,y=e)=cp
*P* (x=011,y=0)=a(1-p)
*P* (x=101,y=0)=b(1-p)
*P* (x=110,y=0)=c(1-p)

$$H(Y|X) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i, y_j)\log_2 p(y_j|x_i)$$
$$= -(1-q)\log 1 - ap\log p - bp\log p - cp\log p$$
$$-a(1-p)\log(1-p) - b(1-p)\log(1-p) - c(1-p)\log(1-p)$$
$$= -qp\log p - q(1-p)\log(1-p)$$

# Example 2

C=max ***I*** (x;y)=max{*H* (y)- *H* (y|x)}

***I*** (x;y)= *H* (y)- *H* (y|x)

$$= -(1-q)\log_2(1-q) - pq\log_2 pq - (1-p)q\log(1-p)q \; -[-qp\log p - q(1-p)\log(1-p)]$$

$$= -(1-q)\log_2(1-q) - pq\log_2 p - pq\log_2 q - (1-p)q\log(1-p) - (1-p)q\log q + qp\log p + q(1-p)\log(1-p)$$

$$= -(1-q)\log_2(1-q) - pq\log_2 q - (1-p)q\log q$$

$$\frac{dI(x;y)}{dq} = \log(1-q) + (1-q)\frac{1}{(1-q)\ln 2} - p\log q - pq\frac{1}{q\ln 2} - (1-p)\log q - (1-p)q\frac{1}{q\ln 2}$$

$$= \log(1-q) - \log(q) = 0 \Rightarrow q = \frac{1}{2}$$

$$I(x;y) = -0.5\log 0.5 - 0.5p\log 0.5 - 0.5(1-p)\log 0.5$$

$$= -0.5\log 0.5 - 0.5\log 0.5$$

$$= 1$$

Thus, $C = 1$

# A Novel Class of Element-wise-inverse Jacket Transform With Many Parameters

# OUTLINE

- some preliminaries and notations are presented
- Element inverse jacket transform with many parameters
- the proposed EIJT transform is efficient algorithm
- results

# PRELIMINARIES AND NOTATIONS

$$W_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -w & w & -1 \\ 1 & w & -w & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

$$W_4^{-1} = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -\frac{1}{w} & \frac{1}{w} & -1 \\ 1 & \frac{1}{w} & -\frac{1}{w} & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

**Definition 2.1** *A matrix* $[J]_{N\times N} = (j_{ik})$ *of order* $N$ *whose entries are complex is called a jacket matrix, if the element in the entries* $(i,k)$ *of its inverse matrix is equal to product of* $\frac{1}{N}$ *and the inverse of the element in the entries* $(k,i)$ *of* $[J]_{N\times N}$. *In other words, if*

$$[J]_{N\times N} = \begin{pmatrix} j_{00} & j_{01} & \cdots & j_{0,N-1} \\ j_{10} & j_{11} & \cdots & j_{1,N-1} \\ \cdots & \cdots & \cdots & \cdots \\ j_{N-1,0} & j_{N-1,1} & \cdots & j_{N-1,N-1} \end{pmatrix}$$

*and its inverse*

$$[J]^{-1}_{N\times N} = \frac{1}{N} \begin{pmatrix} \frac{1}{j_{00}} & \frac{1}{j_{10}} & \cdots & \frac{1}{j_{N-1,0}} \\ \frac{1}{j_{01}} & \frac{1}{j_{11}} & \cdots & \frac{1}{j_{N-1,1}} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{1}{j_{0,N-1}} & \frac{1}{j_{1,N-1}} & \cdots & \frac{1}{j_{N-1,N-1}} \end{pmatrix}$$

*then* $J$ *is called a jacket matrix.*

For example,

$$H_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \qquad H^{-1} = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}.$$

$$J_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega & \omega^2 \\ 1 & \omega^2 & \omega \end{pmatrix}$$

The general form of $3 \times 3$ jacket transforms can be writing as follows:

$$J_3 = \begin{pmatrix} a & b & c \\ d & \frac{bd}{a}\omega & \frac{dc}{a}\omega^2 \\ e & \frac{be}{a}\omega^2 & \frac{ce}{a}\omega \end{pmatrix} \qquad P_3 = \frac{1}{3}\begin{pmatrix} \frac{1}{a} & \frac{1}{d} & \frac{1}{e} \\ \frac{1}{b} & \frac{a}{bd\omega} & \frac{a}{be\omega^2} \\ \frac{1}{c} & \frac{a}{cd\omega^2} & \frac{a}{ce\omega} \end{pmatrix}$$

$$J_3 P_3 = I_3,$$

In order to propose EIJT transform, we need following notations. For an integer $N = 3 \times 2^r$, where $r$ is any arbitrary positive integer. Then any integer $n$ with $0 \leq n \leq N-1$ can be written as

$$n = n_r(3 \times 2^{r-1}) + n_{r-1}2^{r-1} + n_{r-2}2^{r-1} + \cdots + n_1 2 + n_0,$$

where $n_r, n_{r-2}, \ldots, n_0$ take values from $\{0,1\}$ and $n_{r-1}$ takes values from $\{0,1,2\}$. Then $n$ can be represented by a vector

$$n = (n_r, n_{r-1}, \cdots, n_1, n_0).$$

It is easily to see that this representation is unique. In other words, if $n \neq m$, then the vector $(n_r, \cdots, n_0)$ of $n$ is not equal to the vector $(m_r, \cdots, m_0)$ of $m$.

For example, let $N = 3 \times 2^1$. Then the representation of integers $\{0, 1, 2, 3, 4, 5\}$ is $(0, 0)$, $(0, 1)$, $(0, 2)$, $(1, 0), (1, 1)$, $(1, 2)$ respectively. If $N = 3 \times 2^2$, then the representation of $\{0, 1, 2,\ 3,\ 4, 5, 6, 7, 8, 9, 10, 11\}$ is $(0, 0, 0)$, $(0, 0, 1)$, $(0, 1, 0)$, $(0, 1, 1)$, $(0, 2, 0)$, $(0, 2, 1)$, $(1, 0, 0)$, $(1, 0, 1)$, $(1, 1, 0)$, $(1, 1, 1)$, $(1, 2, 0)$, $(1, 2, 1)$ respectively. For any two integers $0 \leq n, m \leq N - 1$ with $n = (n_r, \cdots, n_0)$ and $m = (m_r, \cdots, m_0)$, denoted by

$$\varphi(n, m) = n_{r-2}m_{r-2} + n_{r-3}m_{r-3} + \cdots + n_0m_0,$$

if $r \geq 2$; $\varphi(n, m) = 0$ if $r = 1$ or $r = 0$. For example, if $n = 4$ and $m = 5$, then $\varphi(n, m) = 0$. If $n = 9$, $m = 11$, then $n$ is corresponding to $(1, 1, 1, )$,i.e.,

$$9 = 1 \times (3 \times 2^{2-1}) + 1 \times 2^1 + 1$$

and $m$ is corresponding to $(1, 2, 1)$, i.e.,

$$11 = 1 \times (3 \times 2^{2-1}) + 2 \times 2^1 + 1.$$

Hence

$$\varphi(9, 11) = 1 \times 1 = 1.$$

- The Jacket Transform has following three advantages
- element inverse orthogonality
- the entries of the forward and inverse transforms have reciprocal relation
- the fast algorithm

# PROPOSED FOR EIJT WITH MANY PARAMETERS

**Definition 3.1** *Let $N = 3 \times 2^r$ be a positive integer with arbitrary nonnegative integer $r$ and $\omega$ be a primitive third root of unitary. For any two integers $0 \leq k, m \leq N-1$, let $k$ and $m$ be represented as $k = (k_r, \cdots, k_0)$ and $m = (m_r, \cdots, m_0)$. Denoted by $\varphi(k,m) = k_{r-2}m_{r-2} + \cdots + k_0 m_0$ for $r \geq 2$, otherwise $0$ for $r = 0$ or $r = 1$. Let $\psi(k) = 3k_r + (1-k_r)k_{r-1} + k_r(2-k_{r-1})$. For any complex sequence $(x(0), \cdots, x_{N-1})$ of order $N = 3 \times 2^r$, an EIJT transform is defined as*

$$y(k) = \sum_{m=0}^{N-1} x(m) a_{k,m} (-1)^{\varphi(k,m)} \omega^{\psi(k)\psi(m)}$$

*for $k = 0, 1, \cdots, N-1$, where $\omega$ is a primitive third root of -1, i.e., $\omega^3 = -1$ and the parameters $a_{k,m}$ must satisfy the following conditions*

$$a_{00}a_{km} = a_{k0}a_{0m}, \quad k, m = 1, \cdots, N-1.$$

For example, for $N = 3 \times 2^0$, then the EIJT transform of 3-point can be used as follows:

$$
\begin{aligned}
y(0) &= x(0)a_{00} + x(1)a_{01} + x(2)a_{02} \\
y(1) &= x(0)a_{10} + x(1)\omega^{\frac{a_{10}a_{01}}{a_{00}}} + x(2)\omega^{2\frac{a_{10}a_{02}}{a_{00}}} \\
y(2) &= x(0)a_{20} + x(1)\omega^{2\frac{a_{20}a_{01}}{a_{00}}} + x(2)\omega^{\frac{a_{20}a_{02}}{a_{00}}}
\end{aligned}
$$

**Theorem 3.2** *The inverse transform of the EIJT transform defined by (12) is given by*

$$x(k) = \frac{1}{N}\sum_{m=0}^{N-1} y(m)\frac{1}{a_{mk}}(-1)^{\varphi(k,m)}\omega^{-\psi(k)\psi(m)},$$

*for* $k = 0, 1, \cdots, N-1$.

**Proof.**

First, we establish the following equalities

$$\sum_{m=0}^{N-1} \frac{a_{mp}}{a_{mk}} (-1)^{\varphi(m,p)-\varphi(m,k)} \omega^{(\psi(p)-\psi(k))\psi(m)} = N \qquad \text{for } p = k;$$

$$\sum_{m=0}^{N-1} \frac{a_{mp}}{a_{mk}} (-1)^{\varphi(m,p)-\varphi(m,k)} \omega^{(\psi(p)-\psi(k))\psi(m)} = 0 \qquad \text{for } p \neq k.$$

If $p = k$, then

$$\begin{aligned} & \sum_{m=0}^{N-1} \frac{a_{mp}}{a_{mk}} (-1)^{\varphi(m,p)-\varphi(m,k)} \omega^{(\psi(p)-\psi(k))\psi(m)} \\ = & \sum_{m=0}^{N-1} \frac{a_{mp}}{a_{mp}} (-1)^{\varphi(m,p)-\varphi(m,p)} \omega^{(\psi(p)-\psi(p))\psi(m)} \\ = & \sum_{m=0}^{N-1} 1 = N. \end{aligned}$$

If $p \neq k$, we have $\quad a_{mp} = \frac{a_{m0}a_{0p}}{a_{00}}$ and $a_{mk} = \frac{a_{m0}a_{0k}}{a_{00}}$.

Hence $\quad \frac{a_{mp}}{a_{mk}} = \frac{a_{m0}a_{0p}}{a_{00}} \frac{a_{00}}{a_{m0}a_{0k}} = \frac{a_{0p}}{a_{0k}}$.

Let $m$ be corresponding to the vector $(m_r, \cdots, m_0)$. Then

$$\begin{aligned}
&\sum_{m=0}^{N-1} \frac{a_{mp}}{a_{mk}} (-1)^{\varphi(m,p)-\varphi(m,k)} \omega^{(\psi(p)-\psi(k))\psi(m)} \\
= &\sum_{m=0}^{N-1} \frac{a_{0p}}{a_{0k}} (-1)^{\varphi(m,p)-\varphi(m,k)} \omega^{(\psi(p)-\psi(k))\psi(m)} \\
= &\sum_{(m_r,\cdots,m_0)} \frac{a_{0p}}{a_{0k}} (-1)^{\varphi(m,p)-\varphi(m,k)} \omega^{(\psi(p)-\psi(k))\psi(m)} \\
= &\frac{a_{0p}}{a_{0k}} \sum_{(m_{r-2},\cdots,m_0)} \sum_{m_r=0}^{1} \\
&\sum_{m_{r-1}=0}^{2} (-1)^{\varphi(m,p)-\varphi(m,k)} \omega^{(\psi(p)-\psi(k))\psi(m)} \\
= &\frac{a_{0p}}{a_{0k}} \sum_{(m_{r-2},\cdots,m_0)} [(-1)^{m_{r-2}(p_{r-2}-k_{r-2})+\cdots+m_0(p_0-k_0)} \times \\
&(1+\omega^{\alpha}+\omega^{2\alpha}+\omega^{3\alpha}+\omega^{4\alpha}+\omega^{5\alpha}),
\end{aligned}$$

where the sum in the second equality is taken over all $(m_r, \cdots, m_0)$ which corresponding to $m$ from 1 to $N-1$; and $\alpha = (3p_r + (1-p_r)p_{r-1} + p_r(2-p_{r-1}))\times$
$(3k_r + (1-k_r)k_{r-1} + k_r(2-k_{r-1}))$. Since $\omega$ is a primitive third root of -1, we have $1 + \omega^{\alpha} + \omega^{2\alpha} + \omega^{3\alpha} + \omega^{4\alpha} + \omega^{5\alpha} = 0$. we have

$$x(k) = \frac{1}{N}\sum_{p=0}^{N-1}\sum_{m=0}^{N-1}\frac{a_{mp}}{a_{mk}} \times$$
$$(-1)^{\varphi(m,p)-\varphi(m,k)}\omega^{(\psi(p)-\psi(k))\psi(m)}x(p)$$
$$= \frac{1}{N}\sum_{m=0}^{N-1}\sum_{p=0}^{N-1}\frac{a_{mp}}{a_{mk}} \times$$
$$(-1)^{\varphi(m,p)-\varphi(m,k)}\omega^{(\psi(p)-\psi(k))\psi(m)}x(p)$$
$$= \frac{1}{N}\sum_{m=0}^{N-1}\{\sum_{p=0}^{N-1} x(p)a_{mp}(-1)^{\varphi(m,p)} \times$$
$$\omega^{\psi(p)\psi(m)}\}\frac{1}{a_{mk}}(-1)^{\varphi(k,m)}\omega^{-\psi(k)\psi(m)}$$
$$= \frac{1}{N}\sum_{m=0}^{N-1} y(m)\frac{1}{a_{mk}}(-1)^{\varphi(k,m)}\omega^{-\psi(k)\psi(m)}.$$

Hence we get The inverse transform of the EIJT transform and this completes the proof.

Then the proposed EIJT transform and its inverse transform can be presented in term of the matrix form as follows:

$$Y = P_N X$$

$$X = P_N^{-1} Y,$$

In other words,

$$\left(P^{-1}\right)_{N\times N} = \frac{1}{N}\left(\frac{1}{p_{mk}}\right)_{N\times N}$$

*Example 1* For $N = 3^1$, the forward matrix is as follows:

$$P_6 = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} & a_{05} \\ a_{10} & a_{11}\omega & a_{12}\omega^2 & -a_{13}\omega^2 & -a_{14}\omega & -a_{15} \\ a_{20} & a_{21}\omega^2 & -a_{22}\omega & -a_{23}\omega & a_{24}\omega^2 & a_{25} \\ a_{30} & -a_{31}\omega^2 & -a_{32}\omega & a_{33}\omega & a_{34}\omega^2 & -a_{35} \\ a_{40} & -a_{41}\omega & a_{42}\omega^2 & a_{43}\omega^2 & -a_{44}\omega & a_{45} \\ a_{50} & -a_{51} & a_{52} & -a_{53} & a_{54}\omega & -a_{55} \end{pmatrix}$$

# FAST EFFICIENT ALGORITHM FOR THE EIJT TRANSFORM

$$f(m) = a_{0,m}x(m) + a_{0,N-1-m}x(N-1-m).$$

$$g(m) = a_{0,m}x(m) - a_{0,N-1-m}x(N-1-m).$$

$$\begin{pmatrix} f(m) \\ g(m) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} a_{0,m} & 0 \\ 0 & a_{0,N-1-m} \end{pmatrix} \times \begin{pmatrix} x(m) \\ x(N-1-m) \end{pmatrix}$$

For $N = 6 = 2 \times 3$ with $r = 1$, the EIJT transform is as follows:

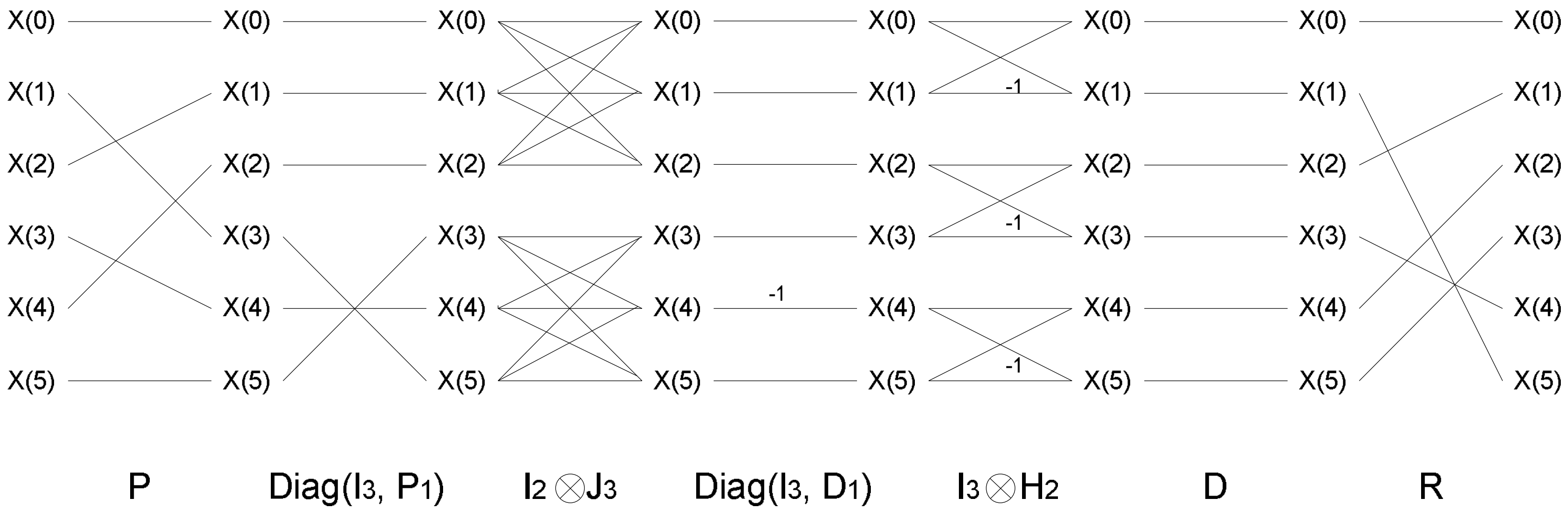
$$Y = [T]_6 X.$$

$$[T]_6 = P \begin{pmatrix} [I]_3 & 0 \\ 0 & P_1 \end{pmatrix} ([I]_2 \otimes [J]_3) \begin{pmatrix} [I]_3 & 0 \\ 0 & D_1 \end{pmatrix} ([I]_3 \otimes [H]_2) DR$$

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad J_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \omega^2 & -\omega \\ 1 & -\omega & \omega^2 \end{pmatrix} \qquad P_1 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

$$D_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad D_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{a_{20}}{a_{00}} & 0 \\ 0 & 0 & \frac{a_{40}}{a_{00}} \end{pmatrix} \qquad D_3 = \begin{pmatrix} \frac{a_{10}}{a_{00}} & 0 & 0 \\ 0 & \frac{a_{30}}{a_{00}} & 0 \\ 0 & 0 & \frac{a_{50}}{a_{00}} \end{pmatrix}$$

$$D = \begin{pmatrix} a_{00} & 0 & 0 & 0 & 0 & 0 \\ 0 & a_{05} & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{01} & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{04} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{02} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{03} \end{pmatrix} \qquad R = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$




$$[T]_6 = P \begin{pmatrix} [I]_3 & 0 \\ 0 & P_1 \end{pmatrix} ([I]_2 \otimes [J]_3) \begin{pmatrix} [I]_3 & 0 \\ 0 & D_1 \end{pmatrix} ([I]_3 \otimes [H]_2)\, DR$$

# CONCLUSION

- A new reciprocal-orthogonal parametric jacket transform (EIJT) is proposed by combining the kernel of the basic jacket transform with five parameters and the well-known WHT transform. On the one hand, the proposed EIJT of a sequence $N = 3^r$ has 2N-1 independent parameters, while the WHT transforms has no independent parameters and the ROP transform is for sequence of power of two. On the other hand, the proposed EIJT still preserves the nice properties of the WHT transforms, such as reciprocal orthogonality, reciprocal relation and fast algorithm.

THANKS

# On Odd Order Jacket Matrices with Some Applications

**joint work with**
Stefan M. Dodunekov
Institute of Mathematics and Informatics (IMI),
Bulgarian Academy of Sciences (BAS)

- Motivation
- Definition & Examples
- Main Result
- Applications in Cryptography

# Motivation

- In 2001, Lee et al. [1] introduced a generalized reverse Jacket transforms (GRJT) as a multi-phase or multilevel generalizations of the WHT and the even-length DFT. The matrix representing a primary GRJT is equivalent to the DFT matrix, and in addition has a border consisting entirely of ±1's.

- However, it can be proven that such matrices with entries from the field of complex numbers, do exist only for even orders [2]. So, it is naturally to ask about **the existence of similar transforms on different spaces of odd dimension**.

# Definition & Examples

- W.l.o.g. we are focussed on the fields GF(p), where p is an odd prime:

DEFINITION 0.1. **A Jacket modulo prime (JMP)** *matrix* $\mathbf{J}$ *of order* $n$ *over* $\mathbf{GF}(p)$ *is an* $n \times n$ *nonsingular matrix of* $\pm\mathbf{1}$*'s of that field such that*

$$\mathbf{J}\mathbf{J}^T = n\mathbf{I},$$

*where* $\mathbf{I}$ *is the identity matrix of order* $n$*, while* $\mathbf{J}^T$ *denotes the transpose matrix of matrix* $\mathbf{J}$*.*

**Remark:** The conventional Hadamard matrices are JMP modulo any prime p, which does not divide the order n.

# Definition & Examples

- **An JMP modulo 3 matrix of order 7:**

$$\mathbf{J}_7 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & -1 & -1 & -1 & 1 \end{pmatrix}.$$

# Definition & Examples

- **More general example:**

Let $J_{p*k+4}$ be a $(p*k+4) \times (p*k+4)$ matrix of $\pm 1$'s, where $k$ is a positive integer, such that:

- its first row and column consist entirely of 1's;
- its last row and column consist of −1's with exception of the corner's entries;
- all other entries are equal to 1 with exception of those on the main diagonal.

**The previous example is obtained if p = 3 and k = 1.**

# Definition & Examples

## Sketch of the Proof:

Any two among the upper $p * k + 3$'s rows agree in exactly $p * k + 2$ places, while the last row has exactly $2$ agreements with each of the others. Thus, the inner product of pair of rows equals to $\pm p * k$, i.e., to $0$ in $\mathbf{GF}(p)$.

$\mathbf{J}_{p*k+4}$ is nonsingular since $p$ and $p * k + 4$ are relatively prime, whenever $p$ is an odd.

**So, there exists JMP modulo arbitrary odd prime p matrix, whose order is also odd.**

# Main Result

- **It is well-known that a necessary condition for the existence of a conventional Hadamard matrix is its order** n **to obey:** n = 1, 2 **or** n  0 mod 4**.**

- **Analogously in this work, we try to find (and found) some necessary conditions for the existence of odd order JMP matrices.**

# Main Result

**The following proposition holds:**

**Proposition 0.2.** *If there exist a JMP modulo odd prime $p$ matrix of order $n \equiv 1 \ (mod\ 2)$ then $n$ is a quadratic residue modulo $p$,*

*i.e.,* $\exists \ \ x : n = x^2 \ (mod\ p)$.

# Main Result

**Example 1:**

Take p = 3. We have: $2^0 = 1$, $2^1 = 2$.

Proposition 0.2 claims:

- the order of any JMP matrix mod 3 must be  1 (mod 6);
- and there exists no such a matrix of order  5 (mod 6). For example, one cannot find a construction of JMP matrix mod 3 whose order is 5 or 11, etc.

**Of course, the requirement for non-singularity implies that the order cannot be multiple of 3.**

# Main Result

**Example 2:**

Take p = 5. We have: $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 3$.

Proposition 0.2 claims:

- we cannot construct JMP matrices of orders 3, 7 (mod 10);
- $J_9$ (p = 5, k = 1) is an JMP matrix mod 5 of order 9. We also, construct an JMP modulo 5 matrix of order 21, which is 1 (mod 10) by using finite projective plane of order 4.

**Of course, the requirement for non-singularity implies that the order cannot be multiple of 5, i.e., 5 (mod 10).**

# An Application in Cryptography

**Definition for $(s, v)$- all-or-noting transform**:

Let $X$ be a finite set, called an alphabet with $|X| = v$. Let $s$ be a positive integer, and assume that $\phi : X^s \to X^s$

Informally, the function $\phi$ is an *all-or-nothing transform* provided that the following properties are satisfied:

- □. $\phi$ is a bijection.
- □. If any $s-1$ of the output $y_1, \ldots, y_s$ are fixed, then the value of any one input variable $x_i, (1 \leq i \leq s)$ is completely undetermined.

**We shall denote such a function as an (s, v)−AONT.**

# An Application in Cryptography

## The cryptographic significance of all-or-nothing transform:

The usage of such a transform **affords a certain amount of additional security (over the block cipher encryption)** since it requires an adversary to decrypt all s blocks of cipher-text (by means of a brute force key search, say) in order to determine any one block of plain-text. As such, it can be thought of as an additional mode of operation that could be used instead of the usual ECB, CFB, CBC or OFB modes.

# An Application in Cryptography

**Theorem 0.3.** *(***Stinson'1999***)*

*Suppose that $q$ is a prime power, and $\mathbf{M}$ is an invertible square matrix of order $s$ with entries from $\mathbb{F}_q$, such that no entry of $\mathbf{M}$ is equal to $0$. Then the function $\phi : \mathbb{F}_q^s \to \mathbb{F}_q^s$ defined by $\phi(\mathbf{x}) = \mathbf{x}\mathbf{M}^{-1}$ is a linear $(s, q)-$AONT.*

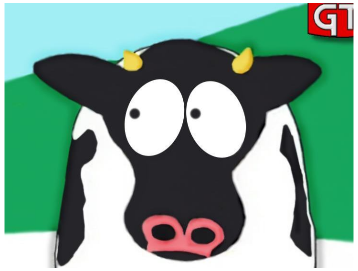
# An Application in Cryptography

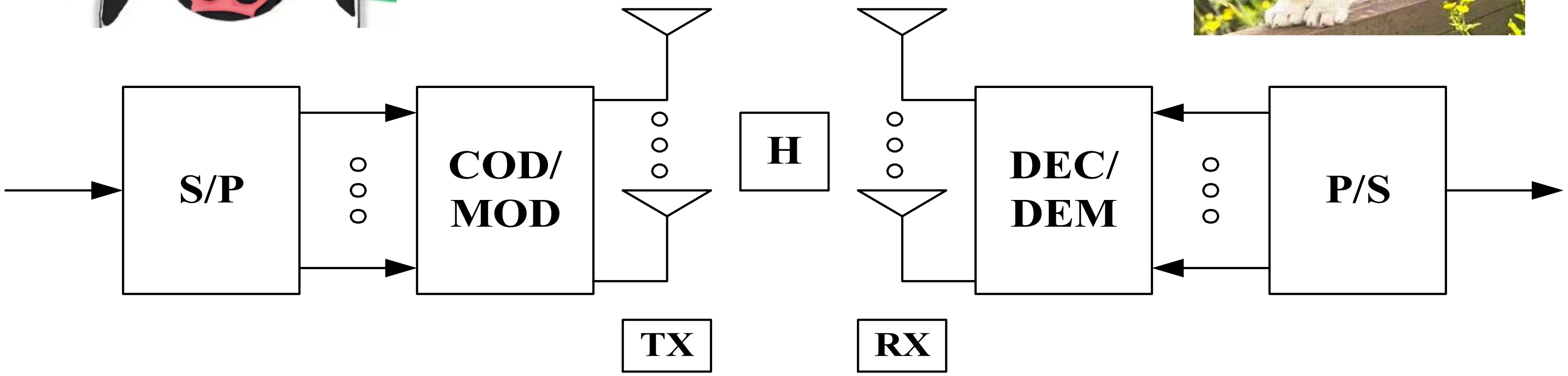
**Finally, we would like to mention 3 points:**

- Stinson gave an example of a linear (s, p)−AONT, where s  0 mod 4 and p > 2 is a prime number, by taking in the role of M (from Theorem 0.3) an ordinary Hadamard matrix whose entries are reduced modulo p (of course, when such matrix does exist for given s).
- The existence of JMP matrices (and the corresponding constructions) whose orders are not  0 mod 4 (discussed in the present work) extends in an obvious way the above described cryptographic application.
- Yet another cryptographic application of JMP matrices was announced by Chang-Hui Choe, who proposed their usage in a running Hill cipher.

# PART II

# MIMO






**multiplexing (=rate) gain** vs **diversity (power) gain**

**Multiple-Antenna (MIMO) System**

# Index Mapping of STBC

$$O_{Tarokh} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ -x_2 & x_1 & -x_4 & x_3 \\ -x_3 & x_4 & x_1 & -x_2 \\ -x_4 & -x_3 & x_2 & x_1 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 2 & 3 & 4 \\ -2 & 1 & -4 & 3 \\ -3 & 4 & 1 & -2 \\ -4 & -3 & 2 & 1 \end{bmatrix}$$

*Change the variables to fixed number*

$$O_4^T O_4 = \begin{bmatrix} x_1 & -x_2 & -x_3 & -x_4 \\ x_2 & x_1 & x_4 & -x_3 \\ x_3 & -x_4 & x_1 & x_2 \\ x_4 & x_3 & -x_2 & x_1 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ -x_2 & x_1 & -x_4 & x_3 \\ -x_3 & x_4 & x_1 & -x_2 \\ -x_4 & -x_3 & x_2 & x_1 \end{bmatrix} = \left(x_1^2 + x_2^2 + x_3^2 + x_4^2\right) I_{4\times 4}$$

$$Ind^T Ind = \begin{bmatrix} 1 & -2 & -3 & -4 \\ 2 & 1 & 4 & -3 \\ 3 & -4 & 1 & 2 \\ 4 & 3 & -2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 & 4 \\ -2 & 1 & -4 & 3 \\ -3 & 4 & 1 & -2 \\ -4 & -3 & 2 & 1 \end{bmatrix} = (1^2 + 2^2 + 3^2 + 4^2) I_{4\times 4}$$

# Index mapping

$$C_{Almouti} = \begin{bmatrix} x_1 & x_2 \\ -x_2* & x_1* \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 2 \\ -2j & j \end{bmatrix}$$

$$\begin{bmatrix} j & -2 \\ 2j & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ -2j & j \end{bmatrix} = \begin{bmatrix} j+4j & 0 \\ 0 & 4j+j \end{bmatrix} = (j+4j)[I]_2$$

Same as the STBC construction criterion

$$C_{Quasi} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ -x_2* & x_1* & -x_4* & x_3* \\ -x_3* & -x_4* & x_1* & x_2* \\ x_4 & -x_3 & -x_2 & x_1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 2 & 3 & 4 \\ -2j & j & -4j & 3j \\ -3j & -4j & j & 2j \\ 4 & -3 & -2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} j & -2 & -3 & 4j \\ -2j & 1 & -4 & -3j \\ 3j & -4 & 1 & -2j \\ 4j & 3 & 2 & j \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 & 4 \\ -2j & j & -4j & 3j \\ -3j & -4j & j & 2j \\ 4 & -3 & -2 & 1 \end{bmatrix} = \begin{bmatrix} j+4j+9j+16j & 0 & 0 & 8j-12j \\ 0 & j+4j+9j+16j & -8j-12j & 0 \\ 0 & -8j-12j & j+4j+9j+16j & 0 \\ 8j-12j & 0 & 0 & j+4j+9j+16j \end{bmatrix}$$

# Pilot Channel Estimation

$$r_1 = S_1\alpha_1 - S_2^*\alpha_2 + n_1 \quad \text{at time} \quad t = T$$

$$r_2 = S_2\alpha_1 + S_1^*\alpha_2 + n_2 \quad \text{at time} \quad t = 2T$$

Pilot patterns : (A, A) for antenna 1

(A, − A) for antenna 2 where A is real number

$$p_1 = A\alpha_1 + A\alpha_2 + n_3$$

$$p_2 = A\alpha_1 - A\alpha_2 + n_4$$

Channel estimates

$$\hat{\alpha}_1 = \frac{p_1 + p_2}{2} \quad \text{and} \quad \hat{\alpha}_2 = \frac{p_1 - p_2}{2}$$

Symbol estimates

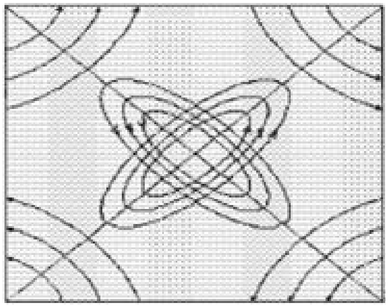
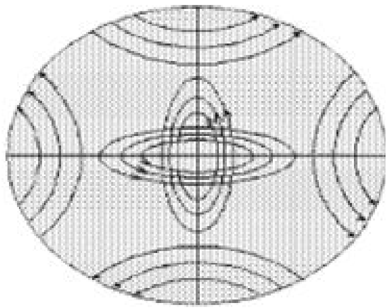
$$\hat{s}_1 = r_1\hat{\alpha}_1^* + r_2^*\hat{\alpha}_2$$

$$\hat{s}_2 = r_2\hat{\alpha}_1^* - r_1^*\hat{\alpha}_2$$

If channel estimates are perfect, diversity can be achieved

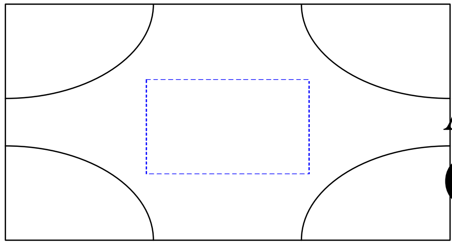
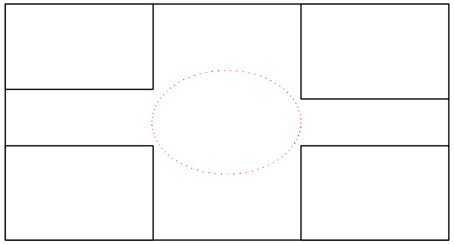
$$\hat{s}_1 = \left(|\alpha_1|^2 + |\alpha_2|^2\right)s_1 + \text{noise terms}$$

# Jacket Matrix

| Walsh Hadamard Matrix | | Jacket Matrix |
|---|---|---|
| $[H]_{k+1}=[H]_k \otimes [H]_1, \; k \geq 1.$ | | $[J]_{k+1}=[J]_k \otimes [H]_1, \; k \geq 2 .$ |
| 4*4 case $\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}=\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}\begin{bmatrix}4&0&0&0\\0&4&0&0\\0&0&4&0\\0&0&0&4\end{bmatrix}$ ⬚ : Fixed weighted factor as 1 | | 4*4 case $\begin{bmatrix}1&1&1&1\\1&-2&2&-1\\1&2&-2&-1\\1&-1&-1&1\end{bmatrix}=\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{bmatrix}\begin{bmatrix}4&0&0&0\\0&6&-2&0\\0&-2&6&0\\0&0&0&4\end{bmatrix}$ ⬚ : Variable weighted factor as 2 or j |
| ▭ : Structure of the sparse matrix  | | ▭ : Structure of the sparse matrix  |
| Space Time Codes | Real Orthogonal ST Designs | Quasi-orthogonal ST Designs |

# Weighted and Quasi Orthogonal

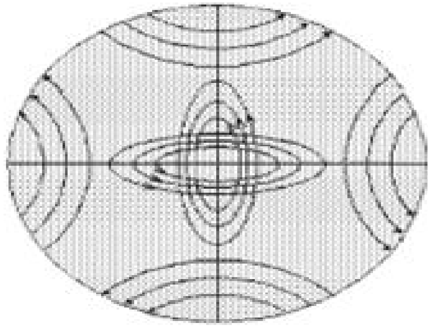
Weighted factor

| | Forward | Reverse |
|---|---|---|
| Jacket matrices | $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -2 & 2 & -1 \\ 1 & 2 & -2 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$ | $\begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & -1 & 1 & -2 \\ 2 & 1 & -1 & -2 \\ 2 & -2 & -2 & 2 \end{bmatrix}$ |
| Graphic Pattern |  |  |
| QOD Pattern | $\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ -x_2{}^* & x_1{}^* & -x_4{}^* & x_3{}^* \\ -x_3{}^* & -x_4{}^* & x_1{}^* & x_2{}^* \\ x_4 & -x_3 & -x_2 & x_1 \end{bmatrix}$ | $\begin{bmatrix} x_1 & x_4 & x_2 & x_3 \\ x_4 & x_1 & x_3 & x_2 \\ -x_2{}^* & -x_3{}^* & x_1{}^* & x_4{}^* \\ -x_3{}^* & -x_2{}^* & x_4{}^* & x_1{}^* \end{bmatrix}$ |
| | denotes the nonorthogonal | denotes the orthogonal |

**Alamouti (Orthogonal)**

Non Alamouti

# Feature selection in sparse matrices

| Jacket Sparse Matrix | |
|---|---|
| $\begin{bmatrix} a & 0 & 0 & d \\ 0 & b & c & 0 \\ 0 & c & b & 0 \\ d & 0 & 0 & a \end{bmatrix}$ <br> $a, b, c, d$ Variable values |  |
| **Jafarkhani Quasi-Orthogonal Design** | **TBH Quasi Orthogonal Design** |
| $S^H S = \begin{bmatrix} a & 0 & 0 & b \\ 0 & a & -b & 0 \\ 0 & -b & a & 0 \\ b & 0 & 0 & a \end{bmatrix}$ | $S^H S = \begin{bmatrix} a & 0 & b & 0 \\ 0 & a & 0 & b \\ b & 0 & a & 0 \\ 0 & b & 0 & a \end{bmatrix}$ |
| $a = \|x_1\|^2 + \|x_2\|^2 + \|x_3\|^2 + \|x_4\|^2$ <br> $b = x_1 x_4 * + x_4 x_1 * - x_2 x_3 * - x_3 x_2 *$ | $a = \|x_1\|^2 + \|x_2\|^2 + \|x_3\|^2 + \|x_4\|^2$ <br> $b = x_1 x_3 * + x_3 x_1 * - x_2 x_4 * - x_4 x_2 *$ |

# Analysis of Existing Designs

- Jacket Unitary pattern from generalizing Walsh and weighted Hadamard.
- Quasi orthogonal after extending the orthogonal cases.
- Analysis the existing patterns to find the unitary properties.

# *Jafarkhani Quasi-Orthogonal*

Same as *Jacket* Matrix

$$C_J = \begin{bmatrix} A & B \\ -B^* & A^* \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ -x_2^* & x_1^* & -x_4^* & x_3^* \\ -x_3^* & -x_4^* & x_1^* & x_2^* \\ x_4 & -x_3 & -x_2 & x_1 \end{bmatrix}$$

$$C_J^H C_J = \begin{bmatrix} a & 0 & 0 & b_J \\ 0 & a & -b_J & 0 \\ 0 & -b_J & a & 0 \\ b_J & 0 & 0 & a \end{bmatrix}$$

Diversity gain: $a = |x_1|^2 + |x_2|^2 + |x_3|^2 + |x_4|^2$

Correlation: $b_J = x_1 x_4^* + x_4 x_1^* - x_2 x_3^* - x_3 x_2^*$

H. JAFARKHANI, IEEE Trans. Communications, vol. 49, No.1, Jan 2001, pp. 1-4

# TBH Quasi-Orthogonal

$$C_T = \begin{bmatrix} A & B \\ B & A \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ -x_2^* & x_1^* & -x_4^* & x_3^* \\ x_3 & x_4 & x_1 & x_2 \\ -x_4^* & x_3^* & -x_2^* & x_1^* \end{bmatrix} \quad C_T{}^H C_T = \begin{bmatrix} a & 0 & b_T & 0 \\ 0 & a & 0 & b_T \\ b_T & 0 & a & 0 \\ 0 & b_T & 0 & a \end{bmatrix}$$

Diversity gain: $a = |x_1|^2 + |x_2|^2 + |x_3|^2 + |x_4|^2$

Correlation: $b_T = x_1 x_3{}^* + x_3 x_1{}^* - x_2 x_4{}^* - x_4 x_2{}^*$

Tirkknen, Beariv, Hottinen, IEEE 6th Int. Sym. SSTA, 2000, New Jersey, USA, pp.429-432

# Measurement of the matrices

Diversity Product:

$$\lambda = \min_{\{C \neq \tilde{C}\}} \left| \det \left[ (C - \tilde{C})^H (C - \tilde{C}) \right] \right|^{\frac{1}{2n}}$$

Jafarkhani case

$$\lambda_J = \min_{\{C \neq \tilde{C}\}} \left| \det\left( \begin{bmatrix} \hat{a} & 0 & 0 & \hat{b}_J \\ 0 & \hat{a} & -\hat{b}_J & 0 \\ 0 & -\hat{b}_J & \hat{a} & 0 \\ \hat{b}_J & 0 & 0 & \hat{a} \end{bmatrix} \right) \right|^{\frac{1}{2n}}$$

$$= \min_{\{C \neq \tilde{C}\}} \left| ((\hat{a})^2 - (\hat{b}_J)^2)^2 \right|^{\frac{1}{2n}}$$

Similar scheme TBH has

$$\lambda_T = \min_{\{C \neq \tilde{C}\}} \left| ((\hat{a})^2 - (\hat{b}_T)^2)^2 \right|^{\frac{1}{2n}}$$

# Unitary of TBH and Jafarkhani

$\lambda$

- The same determinant of the matrices shows TBH and Jafarkhani are unitary.
- We can derive the some patterns from unitary transform.
- From investigation, the distribution of the conjugates is major point.

# *Jafarkhani with TBH Sparse Matrices*

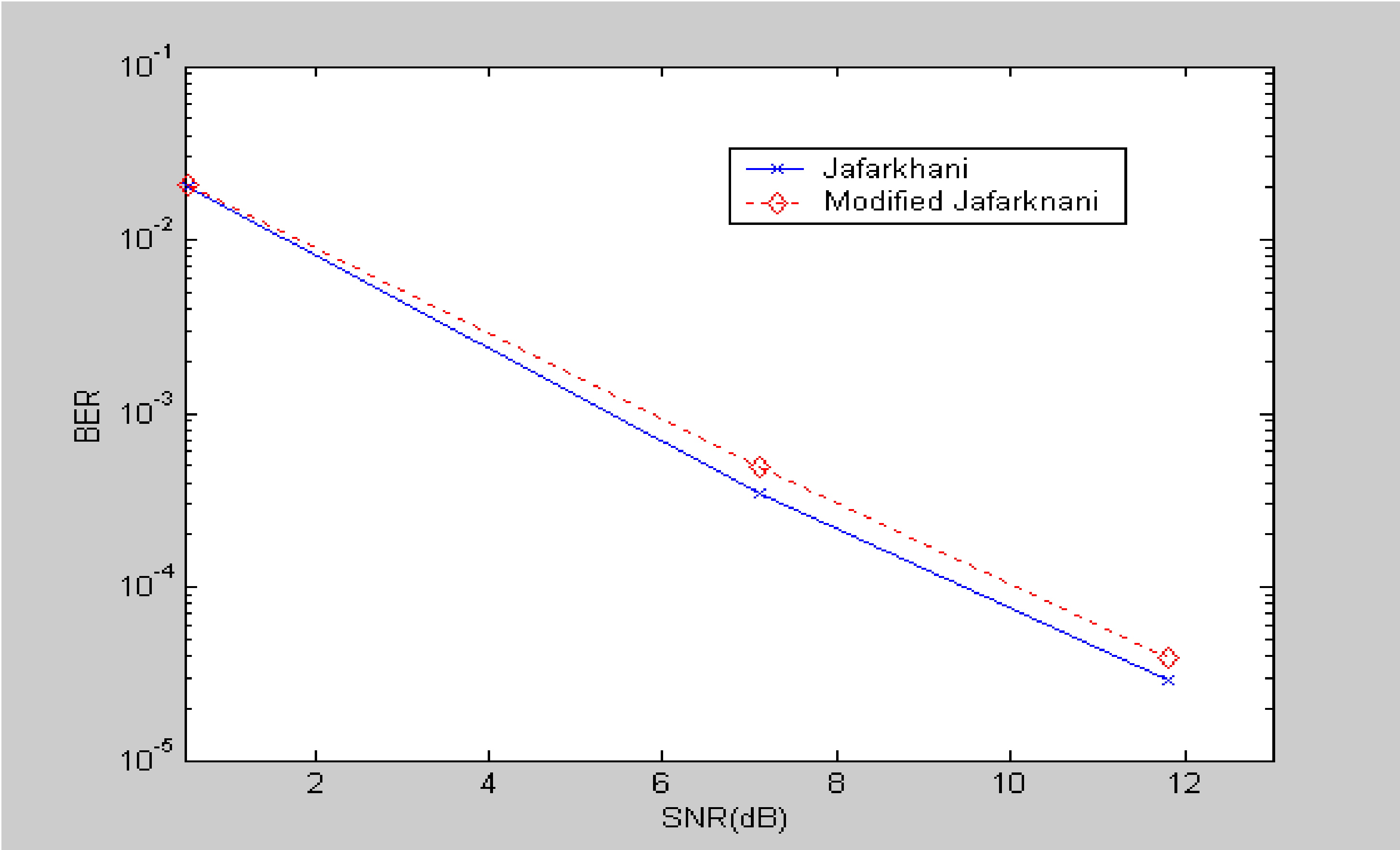
Original Jafarkhani

Let $$C_{JT}^{H} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ -x_2^* & x_1^* & -x_4^* & x_3^* \\ x_4 & -x_3 & -x_2 & x_1 \\ -x_3^* & -x_4^* & x_1^* & x_2^* \end{bmatrix} \longleftarrow \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ -x_2^* & x_1^* & -x_4^* & x_3^* \\ -x_3^* & -x_4^* & x_1^* & x_2^* \\ x_4 & -x_3 & -x_2 & x_1 \end{bmatrix}$$

We have $$C_{JT}^{H} C_{JT} = \begin{bmatrix} a & 0 & -b & 0 \\ 0 & a & 0 & b \\ -b & 0 & a & 0 \\ 0 & b & 0 & a \end{bmatrix} \longleftarrow \begin{bmatrix} a & 0 & 0 & b \\ 0 & a & -b & 0 \\ 0 & -b & a & 0 \\ b & 0 & 0 & a \end{bmatrix} = C_J^H C_J$$

*b: correlations position is moving*

# *Performances of Modified Jafarkhani*



# *TBH with Jafarkhani Sparse Matrices*

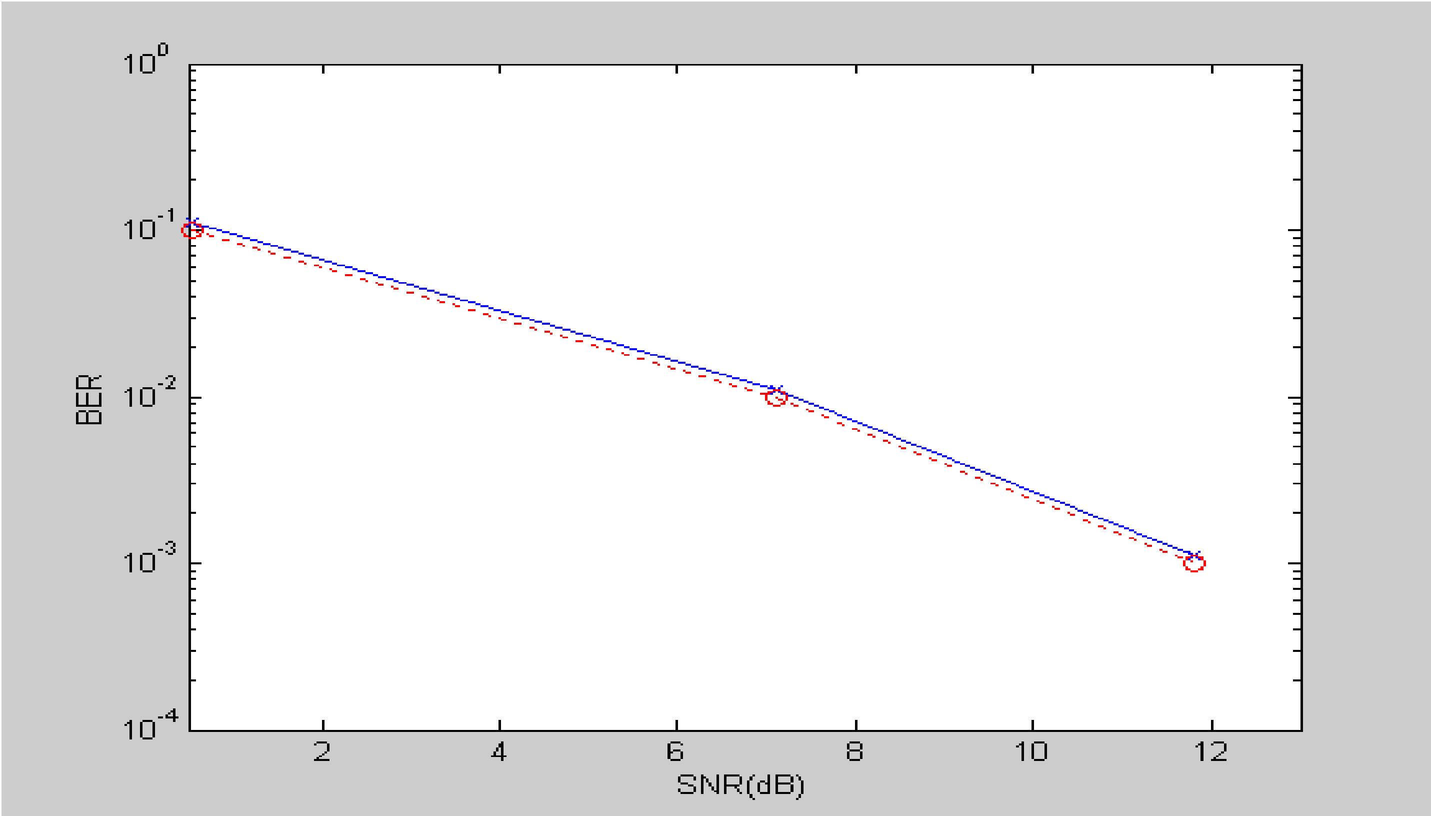
Original TBH

Let $C_{TJ}^{H} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ -x_2* & x_1* & -x_4* & x_3* \\ -x_4* & x_3* & -x_2* & x_1* \\ x_3 & x_4 & x_1 & x_2 \end{bmatrix} \longleftarrow \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ -x_2* & x_1* & -x_4* & x_3* \\ x_3 & x_4 & x_1 & x_2 \\ -x_4* & x_3* & -x_2* & x_1* \end{bmatrix}$

Then we have $C_{TJ}^{H} C_{TJ} = \begin{bmatrix} a & 0 & 0 & b \\ 0 & a & b & 0 \\ 0 & b & a & 0 \\ b & 0 & 0 & a \end{bmatrix} \longleftarrow \begin{bmatrix} a & 0 & b & 0 \\ 0 & a & 0 & b \\ b & 0 & a & 0 \\ 0 & b & 0 & a \end{bmatrix}$

Correlated positions changed

# Performances of Modified TBH




Red line is modified TBH and Blue line is TBH

# Distribution of conjugates on the bottom (1)

- $A'_{12} = \begin{bmatrix} x_1 & x_2 \\ x_2 & x_1 \end{bmatrix}$

$$\Rightarrow C_{N1} = \begin{bmatrix} A_{12}' & A_{34}' \\ -(A_{34}')^* & (A_{12}')^* \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ x_2 & x_1 & x_4 & x_3 \\ -x_3{}^* & -x_4{}^* & x_1{}^* & x_2{}^* \\ -x_4{}^* & -x_3{}^* & x_2{}^* & x_1{}^* \end{bmatrix}$$

$$\Rightarrow C_{N1}{}^H C_{N2} = \begin{bmatrix} a & b_{N1} & 0 & 0 \\ b_{N1} & a & 0 & 0 \\ 0 & 0 & a & b_{N1} \\ 0 & 0 & b_{N1} & a \end{bmatrix}$$

$$b_{N1} = x_1 x_2{}^* + x_1{}^* x_2 + x_3 x_4{}^* + x_4 x_3{}^*$$

The diversity product is same as before:

$$\lambda_{N1} = \min_{\{C \neq \tilde{C}\}} \left| \det(B_{N1}^H B_{N1}) \right|^{\frac{1}{2n}} = \min_{\{C \neq \tilde{C}\}} \left| (\hat{a}^2 - \hat{b}_{N1}^2)^2 \right|^{\frac{1}{2n}}$$



Jia Hou, and Moon Ho Lee, "Matrices analysis of quasi orthogonal STBC" IEEE Comm. June, 2003.

# Distribution of conjugates on the bottom (1)

- $$A'_{12} = \begin{bmatrix} x_1 & x_2 \\ x_2 & x_1 \end{bmatrix}$$

$$\Rightarrow C_{N1} = \begin{bmatrix} A_{12}' & A_{34}' \\ -(A_{34}')^* & (A_{12}')^* \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ x_2 & x_1 & x_4 & x_3 \\ -x_3^* & -x_4^* & x_1^* & x_2^* \\ -x_4^* & -x_3^* & x_2^* & x_1^* \end{bmatrix}$$

$$\Rightarrow C_{N1}{}^H C_{N2} = \begin{bmatrix} a & b_{N1} & 0 & 0 \\ b_{N1} & a & 0 & 0 \\ 0 & 0 & a & b_{N1} \\ 0 & 0 & b_{N1} & a \end{bmatrix}$$

$$b_{N1} = x_1 x_2{}^* + x_1{}^* x_2 + x_3 x_4{}^* + x_4 x_3{}^*$$

The diversity product is same as before:

$$\lambda_{N1} = \min_{\{C \neq \tilde{C}\}} \left| \det(B_{N1}^H B_{N1}) \right|^{\frac{1}{2n}} = \min_{\{C \neq \tilde{C}\}} \left| (\hat{a}^2 - \hat{b}_{N1}^2)^2 \right|^{\frac{1}{2n}}$$

Jia Hou, and Moon Ho Lee, "Matrices analysis of quasi orthogonal STBC" IEEE Comm. June, 2003.

# Matrices Analysis of Quasi-Orthogonal Space-Time Block Codes

Jia Hou, Moon Ho Lee, *Senior Member, IEEE*, and Ju Yong Park

***Abstract*—In this letter, according to the analysis of existing transmission matrices of quasi-orthogonal space-time block codes (STBC), we generalize some of their characters and derive several new patterns to enrich the family of quasi-orthogonal STBC.**

***Index Terms*—Quasi-orthogonal, STBC, unitary pattern.**

## I. Introduction

IN ADDRESSING the issue of decoding complexity, STBC can efficiently achieve the transmit diversity to combat fading. By using the orthogonality of the transmitted symbols, Alamouti [3] first defined a space time transmission matrix as

$$A_{12} = \begin{bmatrix} x_1 & x_2 \\ -x_2^* & x_1^* \end{bmatrix} \quad (1)$$

where the subscript *12* indicates the indeterminates $x_1$ and $x_2$ existing in the transmission matrix. Based on Alamouti orthogonal STBC, Jafarkhani [2] gave a quasi-orthogonal STBC form for four transmit antennas as

$$C_J = \begin{bmatrix} A_{12} & A_{34} \\ -A_{34}^* & A_{12}^* \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ -x_2^* & x_1^* & -x_4^* & x_3^* \\ -x_3^* & -x_4^* & x_1^* & x_2^* \\ x_4 & -x_3 & -x_2 & x_1 \end{bmatrix} \quad (2)$$

where $A_{12}$, $A_{34}$ are Alamouti codes. Its character matrix has similar fashion as the sparse matrix pattern in [5], and we can write it as

$$C_J^H C_J = \begin{bmatrix} a & 0 & 0 & b_J \\ 0 & a & -b_J & 0 \\ 0 & -b_J & a & 0 \\ b_J & 0 & 0 & a \end{bmatrix} \quad (3)$$

where $C^H$ is the Hermitian of matrix $C$, $a = \sum_{i=0}^{4} |x_i|^2$, and the correlated vale $b_J = (x_1 x_4^* + x_1^* x_4) - (x_2 x_3^* + x_3 x_2^*)$ is a real number. Further, different from Jafarkhani scheme, the TBH case [4] has

$$C_T = \begin{bmatrix} A_{12} & A_{34} \\ A_{34} & A_{12} \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ -x_2^* & x_1^* & -x_4^* & x_3^* \\ x_3 & x_4 & x_1 & x_2 \\ -x_4^* & x_3^* & -x_2^* & x_1^* \end{bmatrix} \quad (4)$$

$$C_T^H C_T = \begin{bmatrix} a & 0 & b_T & 0 \\ 0 & a & 0 & b_T \\ b_T & 0 & a & 0 \\ 0 & b_T & 0 & a \end{bmatrix} \quad (5)$$

where the correlated value $b_T = x_1 x_3^* + x_1^* x_3 + x_2 x_4^* + x_4 x_2^*$. In this letter, we introduce some new designs through analyzing the character matrices of existing schemes. Moreover, one of the new designs can improve the performances by reducing the interferences of adjacent symbols.

## II. Analysis and Proposed New Matrices

Using a unitary pattern idea introduced in [5] to investigate the distribution of conjugates in the transmission matrices, we find that it is related to the positions of correlated values. By changing the distribution of conjugates, we can obtain matrices with different positions of correlated values.

### *A. Jafarkhani Case With TBH Correlated Positions*

We change the conjugates' distribution of Jafarkhani matrix, and let

$$C_{JT}^H = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ -x_2^* & x_1^* & -x_4^* & x_3^* \\ x_4 & -x_3 & -x_2 & x_1 \\ -x_3^* & -x_4^* & x_1^* & x_2^* \end{bmatrix}. \quad (6)$$

By exchanging the last row and the third row from (2), we may get the character matrix as

$$C_{JT}^H C_{JT} = \begin{bmatrix} a & 0 & b_{JT} & 0 \\ 0 & a & 0 & -b_{JT} \\ b_{JT} & 0 & a & 0 \\ 0 & -b_{JT} & 0 & a \end{bmatrix} \quad (7)$$

where the correlated value $b_{JT} = b_J$, but the positions of correlated values are the same as the TBH case.

In addition, the measurement of error probability is from calculating the diversity product $\lambda$[1]. We show it as

$$\lambda = \min_{\{C \neq \hat{C}\}} \left| \det\left[ (C - \hat{C})^H (C - \hat{C}) \right] \right|^{1/2n} \quad (8)$$

where $n$ is the number of transmit antennas, and $\hat{C}$ is the error code words from $C$. Assuming $B_J = (C_J - \hat{C}_J)$, $B_{JT} = (C_{JT} - \hat{C}_{JT})$, and $\tilde{x}_i = x_i - \hat{x}_i$, $i \in \{1, 2, 3, 4\}$ as elements of error matrix $B$, the diversity products from the Jafarkhani and its modified case are

$$\lambda_J = \min_{\{C \neq \hat{C}\}} \left| \det(B_J^H B_J) \right|^{1/2n}$$

Manuscript received November 26, 2002. The associate editor coordinating the review of this letter and approving it for publication was Prof. A. Annamalai. This work was supported by KOSEF 303-15-2 and NGVTEK.

J. Hou and M. H. Lee are with the Institute of Information and Communication, Chonbuk National University, Chonju, 561-756, Korea (e-mail: houjiastock@hotmail.com; moonho@chonbuk.ac.kr).

J. Y. Park is with the Department of Electronic Engineering, Seonam University, 590-711 Seonam, Korea (e-mail: pjyhj@kornet.net).

Digital Object Identifier 10.1109/LCOMM.2003.814706

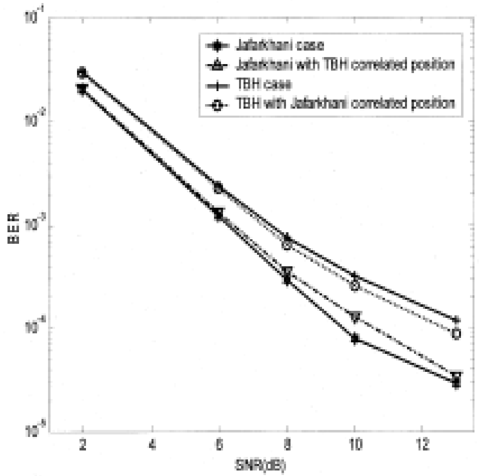



Fig. 1. Performances of the modified cases with different correlated positions using four transmit antennas and one receive antenna over flat fading channel.

$$= \min_{\{C \neq \hat{C}\}} \left| \det \begin{bmatrix} \hat{a} & 0 & 0 & \hat{b}_J \\ 0 & \hat{a} & -\hat{b}_J & 0 \\ 0 & -\hat{b}_J & \hat{a} & 0 \\ \hat{b}_J & 0 & 0 & \hat{a} \end{bmatrix} \right|^{1/2n} \quad (9)$$

$$\lambda_{JT} = \min_{\{C \neq \hat{C}\}} \left| \det(B_{JT}^H B_{JT}) \right|^{1/2n}$$

$$= \min_{\{C \neq \hat{C}\}} \left| \det \begin{bmatrix} \hat{a} & 0 & \hat{b}_{JT} & 0 \\ 0 & \hat{a} & 0 & -\hat{b}_{JT} \\ \hat{b}_{JT} & 0 & \hat{a} & 0 \\ 0 & -\hat{b}_{JT} & 0 & \hat{a} \end{bmatrix} \right|^{1/2n} \quad (10)$$

where $\hat{a} = \sum_{i=1}^{4} |\hat{x}_i|^2$ and $\hat{b}_J = \hat{b}_{JT} = (\hat{x}_1\hat{x}_4^* + \hat{x}_1^*\hat{x}_4) - (\hat{x}_2\hat{x}_3^* + \hat{x}_2^*\hat{x}_3)$. Thus we have $\lambda_J = \lambda_{JT}$, because

$$\det(B_J^H B_J) = \det(B_{JT}^H B_{JT}) = (\hat{a}^2 - \hat{b}_J^2)^2. \quad (11)$$

Based on the same diversity product, these two cases have similar performances by using the same maximum ratio combing (MRC) decoding algorithm proposed in [1], [2]. The numerical results are shown in Fig. 1.

### B. TBH Case With Jafarkhani-Correlated Positions

Similar to the above modification, we exchange the last row and the third row from (4) and let

$$C_{TJ}^H = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ x_2^* & -x_1^* & x_4^* & -x_3^* \\ x_4^* & -x_3^* & x_2^* & -x_1^* \\ x_3 & x_4 & x_1 & x_2 \end{bmatrix} \quad (12)$$

thus we obtain the character matrix as

$$C_{TJ}^H C_{TJ} = \begin{bmatrix} a & 0 & 0 & b_{TJ} \\ 0 & a & b_{TJ} & 0 \\ 0 & b_{TJ} & a & 0 \\ b_{TJ} & 0 & 0 & a \end{bmatrix} \quad (13)$$

where $b_{TJ} = b_T$ and the positions of correlated values are the same as the Jafarkhani case. Similar to the above analysis, we have $\lambda_T = \lambda_{TJ}$, because

$$\det(B_T^H B_T) = \det(B_{TJ}^H B_{TJ}) = (\hat{a}^2 - \hat{b}_T^2)^2 \quad (14)$$

where $\hat{b}_T = \hat{b}_{TJ} = (\hat{x}_1\hat{x}_3^* + \hat{x}_1^*\hat{x}_3 + \hat{x}_2\hat{x}_4^* + \hat{x}_2^*\hat{x}_4)$. By using the MRC decoding algorithm, these two cases present similar performances, and the simulations are shown in Fig. 1. Therefore, generalizing the above two modified cases, the different distribution of the conjugates in transmission matrix can lead to different positions of correlated values and the positions of correlated values in the character matrix are not directly corresponding to the performances of quasi-orthogonal STBC.

### C. Designs With New Positions of Correlated Values

According to the above analysis, we know the positions of correlated values do not affect the BER. Therefore, now we derive some new matrices with different positions of correlated values from the distribution of conjugates in the bottom of transmission matrices.

*Case 1:* Let the submatrix be denoted as

$$A'_{12} = \begin{bmatrix} x_1 & x_2 \\ x_2 & x_1 \end{bmatrix}. \quad (15)$$

Then we design a new form as

$$C_{N1} = \begin{bmatrix} A'_{12} & A'_{34} \\ -(A'_{34})^* & (A'_{12})^* \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ x_2 & x_1 & x_4 & x_3 \\ -x_3^* & -x_4^* & x_1^* & x_2^* \\ -x_4^* & -x_3^* & x_2^* & x_1^* \end{bmatrix} \quad (16)$$

where $A'_{12}$, $A'_{34}$ are from (15). Thus we can write

$$C_{N1}^H C_{N1} = \begin{bmatrix} a & b_{N1} & 0 & 0 \\ b_{N1} & a & 0 & 0 \\ 0 & 0 & a & b_{N1} \\ 0 & 0 & b_{N1} & a \end{bmatrix} \quad (17)$$

where $b_{N1} = x_1x_2^* + x_1^*x_2 + x_3x_4^* + x_3^*x_4$ is a real number. Based on the same MRC decoding algorithm, the performance of the new design case 1(*ND1*) is very close to that of TBH case, as shown in Fig. 2. The diversity product of *ND1* also is similar to TBH case as

$$\lambda_{N1} = \min_{\{C \neq \hat{C}\}} \left| \det(B_{N1}^H B_{N1}) \right|^{1/2n} = \min_{\{C \neq \hat{C}\}} \left| (\hat{a}^2 - \hat{b}_{N1}^2)^2 \right|^{1/2n}. \quad (18)$$

*Case 2:* Let the submatrix be denoted as

$$A''_{12} = \begin{bmatrix} x_1 & x_2 \\ -x_2 & x_1 \end{bmatrix}. \quad (19)$$

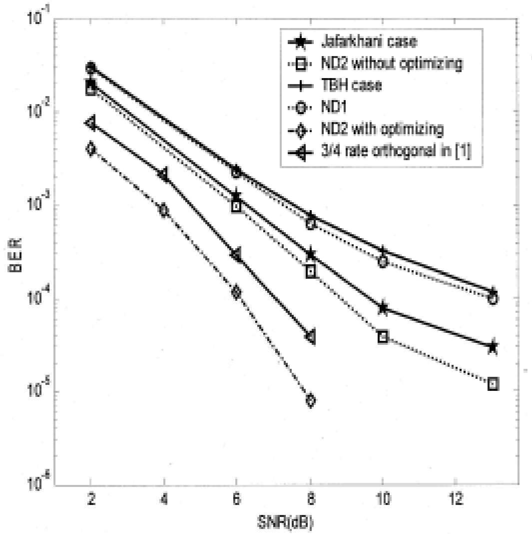



Fig. 2. Performances of the new design cases using four transmit antennas and one receive antenna over flat fading channel.

Then we define the new form as

$$C_{N2} = \begin{bmatrix} A''_{12} & A''_{34} \\ -(A''_{34})^H & (A''_{12})^H \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ -x_2 & x_1 & -x_4 & x_3 \\ -x_3^* & x_4^* & x_1^* & -x_2^* \\ -x_4^* & -x_3^* & x_2^* & x_1^* \end{bmatrix} \quad (20)$$

and its character matrix is

$$C_{N2}^H C_{N2} = \begin{bmatrix} a & b_{N2} & 0 & 0 \\ -b_{N2} & a & 0 & 0 \\ 0 & 0 & a & b_{N2} \\ 0 & 0 & -b_{N2} & a \end{bmatrix} \quad (21)$$

where $b_{N2} = x_1^* x_2 - x_1 x_2^* + x_3^* x_4 - x_3 x_4^*$ is a imaginary number. And the diversity product of the new design case 2 (*ND2*) is

$$\lambda_{N2} = \min_{\{C \neq \tilde{C}\}} \left| \det(B_{N2}^H B_{N2}) \right|^{1/2n} = \min_{\{C \neq \tilde{C}\}} \left| (\hat{a}^2 + \hat{b}_{N2}^2)^2 \right|^{1/2n}. \quad (22)$$

Based on the pairs of transmit symbols, the Jafarkhani correlated functions are $f(x_1, x_4)$ and $f(x_2, x_3)$[2]. However, in the new design cases, the correlations are only from the interferences of adjacent symbols $I(x_{2k-1}, x_{2k})$. In the *ND2*, we have

$$I(x_{2k-1}, x_{2k}) = x_{2k-1}^* x_{2k} - x_{2k-1} x_{2k}^*, \quad k = 1, 2, 3 \ldots . \quad (23)$$

By using a simple 3/4 rate modulation scheme, we can easily optimize the values from (23) and improve the performance of the *ND2*. The optimized modulation is given as

$$Q_{2k} = \frac{Q_{2k-1} I_{2k-1}}{I_{2k}} \quad (24)$$

where $I$ and $Q$ denote the in-phase and quadrature separately. From observation of Fig. 2, it is apparent that the *ND2* is not much better than the Jafarkhani case when the adjacent interferences exist but has remarkable enhancement from conventional quasi-orthogonal designs by using optimizing the interferences of adjacent symbols.

## III. Conclusion

Generally, based on a unitary pattern idea from [5], this letter analyzes the character matrices of quasi-orthogonal STBC and derives several new codes to enrich their family. The new codes have different positions of correlated values in the character matrices by changing the distribution of conjugates in transmission matrices. In addition, some of them can improve BER with a simple optimizing scheme.

## References


[1] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space time block codes from orthogonal designs," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1456–1467, 1999.

[2] H. Jafarkhani, "A quasiorthogonal space time block codes," *IEEE Trans. Commun.*, vol. 49, pp. 1–4, 2001.

[3] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Selected Areas in Commun.*, vol. 16, pp. 1451–1458, 1998.

[4] O. Tirkkonen, A. Boariu, and A. Hottinen, "Minimal nonorthogonality rate one space time block codes for 3+ Tx antennas," *Proc. IEEE Int. Symp. on Spread Spectrum Techniques and Applications, ISSSTA 2000*, pp. 429–432, 2000.

[5] M. H. Lee, "A new reverse jacket transform and its fast algorithm," *IEEE Trans. Circuit and Syst. II*, vol. 47, pp. 39–47, Jan. 2000.

# DCT/DFT/Wavelet Hybrid Architecture Via Recursive Factorization

# Contents

- **Jacket Matrix(JM) Concept**
- **Properties of Jacket-like Sparse Matrix**
  – Element Inverse and Block Inverse
- **Recursive Factorization**
  – DCT-II matrix
  – DFT matrix
  – Wavelet based on Haar matrix
- **Dual Use of Cooley-Tukey Type** Data Flow Diagram
  – DFT data flow diagram
  – DCT-II data flow diagram
  – Wavelet data flow diagram
- **Discussions**
- **Conclusions**

# Jacket Matrix(JM) Concept

- General case :
  - A=QR factorization can be viewed as a Reverse Jacket Matrix representation.
  - Normally, matrix R is a sparse matrix and matrix Q is a unitary matrix.
- Special case : matrix Q can be fixed to a trigonometric transform matrix by a constraint.
  - A Reverse Jacket Matrix(RJM) is a generalized weighted Hadamard transform matrix [9] and refers to such a case.

    Here$[S]_m$ is a sparse matrix of $[J]_m$.

$$[J]_m = \frac{1}{m}[H]_m[S]_m, \; m = 2^{k+1}, k \in \{1,2,3,4,\ldots\} \; (1)$$

# Properties of Jacket-like Sparse Matrix

- Jacket matrix $[J]_m$ has an element inverse property.
- The inverse of $[J]_m$ is also a Jacket matrix.
- All trigonometric transform matrix(DFT, DCT, DST, WHT, Haar Transform, Hartley Transform, etc) can be represented as a Jacket-like sparse matrix. For example, DFT matrix for **N=8 case** can be represented as follows:

$$[F] = \sum_{m=0}^{N-1} \left( e^{-j\frac{2\pi}{N}} \right)^{nm}, \quad 0 \le n \le N-1$$

$$[F]_8 = \frac{1}{4}[H]_8 \begin{bmatrix} 4I_2 & 0 & 0 \\ 0 & 2(G_2 H_2)^h & 0 \\ 0 & 0 & (G_4 H_4)^h \end{bmatrix} [P]_8$$

$$= \frac{1}{4}[H]_8 \left[\hat{S}\right]_8 [P]_8 = \frac{1}{4}\left[\widetilde{F}\right]_8 [P]_8$$

# Recursive Factorization(1 ):DCT-II

- A typical forward DCT of type II of a sequence length N is given by:

DCT-II:

$$[C_N]_{m,n} = \sqrt{\frac{2}{N}} k_m \cos\frac{m(n+\frac{1}{2})\pi}{N}, \quad m,n = 0,1,\ldots,N-1$$

$$\text{where } k_j = \begin{cases} 1, & j = 1,2,\ldots,N-1 \\ \frac{1}{\sqrt{2}}, & j = 0, N \end{cases} \qquad (2)$$

# Recursive Factorization(2 ):DCT-II

- To generalize a recursive factorization of size N DCT-II , we start with N=2,4, and 8:

$$[C]_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_4^1 & C_4^3 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} C_1 & C_1 \\ B_1 & -B_1 \end{bmatrix}$$

$$[P_r]_4 [C]_4 [P_c]_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_8^1 & C_8^3 & C_8^5 & C_8^7 \\ C_8^2 & C_8^6 & C_8^6 & C_8^2 \\ C_8^3 & C_8^7 & C_8^1 & C_8^5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_8^2 & C_8^6 & C_8^6 & C_8^2 \\ C_8^1 & C_8^3 & C_8^5 & C_8^7 \\ C_8^3 & C_8^7 & C_8^1 & C_8^5 \end{bmatrix} = \begin{bmatrix} [C]_2 & [C]_2 \\ [B]_2 & -[B]_2 \end{bmatrix} = \begin{bmatrix} [C]_2 & 0 \\ 0 & [B]_2 \end{bmatrix} \begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix}$$

# Recursive Factorization(3 ):DCT-II

- Similarly, for N=8, we obtain the followings:

$$[P_r]_8[C_8][P_c]_8 = [\tilde{C}_8] = \begin{bmatrix} C_4 & C_4 \\ B_4 & -B_4 \end{bmatrix} = \begin{bmatrix} C_4 & 0 \\ 0 & B_4 \end{bmatrix} \begin{bmatrix} I_4 & I_4 \\ I_4 & -I_4 \end{bmatrix}$$

By induction,

$$[\tilde{C}]_N = [\mathrm{Pr}]_N[C]_N[Pc]_N = \begin{bmatrix} C_{N/2} & C_{N/2} \\ B_{N/2} & -B_{N/2} \end{bmatrix} = \left( \begin{bmatrix} C_{N/2} & 0 \\ 0 & B_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \right)$$

Where $[P_r]_N = BRO_N, [P_c]_N = \begin{bmatrix} I_{N/4} & 0 & 0 & 0 \\ 0 & I_{N/4} & 0 & 0 \\ 0 & 0 & 0 & I_{N/4} \\ 0 & 0 & I_{N/4} & 0 \end{bmatrix}$

# Recursive Factorization(4 ):DCT-II

- Top left block matrix $[C]_{N/2}$ of $[C]_N$ has a recursive factorization, but bottom right block matrix $[B]_{N/2}$ of $[C]_N$ does not.
- Many authors [Chen, Wang ] “High throughput VLSI architecures for the 1-D and 2-D discrete cosine transforms”, IEEE Trans. Circuits Syst. Video Technol., 1995, proposed a further decomposition algorithm to derive a fast implementation of $[B]_{N/2}$ computation, which normally requires (N/2 x N/2 ) real multiplication.
- Our proposed algorithm partitions $[B]_{N/2}$ into a recursive form using both generation matrix and the trigonometric identities and relations explained below:

$$\text{Generation matrix} : [B]_{N/2} = \left[ \left( C_{2N}^{f(m,n)} \right)_{m,n} \right]_{N/2} \quad (3)$$

# Recursive Factorization(5 ):DCT-II

- In case of NxN DCT-II matrix, $[C]_N$ can be represented using the form as:

$$
[C]_N = \begin{bmatrix}
\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\
C_{4N}^{2k_0\Phi_0} & C_{4N}^{2k_0\Phi_1} & C_{4N}^{2k_0\Phi_2} & \cdots & C_{4N}^{2k_0\Phi_{N-2}} & C_{4N}^{2k_0\Phi_{N-1}} \\
C_{4N}^{2k_1\Phi_0} & C_{4N}^{2k_1\Phi_1} & C_{4N}^{2k_1\Phi_2} & \cdots & C_{4N}^{2k_1\Phi_{N-2}} & C_{4N}^{2k_1\Phi_{N-1}} \\
C_{4N}^{2k_2\Phi_0} & C_{4N}^{2k_2\Phi_1} & C_{4N}^{2k_2\Phi_2} & \cdots & C_{4N}^{2k_2\Phi_{N-2}} & C_{4N}^{2k_2\Phi_{N-1}} \\
\vdots & & & \vdots & & \vdots \\
C_{4N}^{2k_{N-2}\Phi_0} & C_{4N}^{2k_{N-2}\Phi_1} & C_{4N}^{2k_{N-2}\Phi_2} & \cdots & C_{4N}^{2k_{N-2}\Phi_{N-2}} & C_{4N}^{2k_{N-2}\Phi_{N-1}}
\end{bmatrix},
$$

where $k_i = i + 1, i \in \{0,1,2,...\}$

# Recursive Factorization(6 ):DCT-II

- According to (3), a NxN matrix $[B]_N$ from $[C]_{2N}$ can be simply presented by:

$$[B]_N = \begin{bmatrix} C_{4N}^{\Phi_0} & C_{4N}^{\Phi_1} & C_{4N}^{\Phi_2} & \cdots & C_{4N}^{\Phi_{N-1}} \\ C_{4N}^{(2k_0+1)\Phi_0} & C_{4N}^{(2k_0+1)\Phi_1} & C_{4N}^{(2k_0+1)\Phi_2} & \cdots & C_{4N}^{(2k_0+1)\Phi_{N-1}} \\ C_{4N}^{(2k_1+1)\Phi_0} & C_{4N}^{(2k_1+1)\Phi_1} & C_{4N}^{(2k_1+1)\Phi_2} & \cdots & C_{4N}^{(2k_1+1)\Phi_{N-1}} \\ \vdots & & \vdots & & \vdots \\ C_{4N}^{(2k_{N-2}+1)\Phi_0} & C_{4N}^{(2k_{N-2}+1)\Phi_1} & C_{4N}^{(2k_{N-2}+1)\Phi_2} & \cdots & C_{4N}^{(2k_{N-2}+1)\Phi_{N-1}} \end{bmatrix}$$

Using the trigonometric identities and relations, we can have the following equation:

$$C_{4N}^{(2k_i+1)\Phi_m} = 2C_{4N}^{2k_i\Phi_m} C_{4N}^{\Phi_m} - C_{4N}^{(2k_i-1)\Phi_m} = -C_{4N}^{(2k_i-1)\Phi_m} + 2C_{4N}^{2k_i\Phi_m} C_{4N}^{\Phi_m},$$

$m \in \{0,1,2,\ldots.\}$ Example: N=2, k=1, $C_8^{(2+1)=3} = 2C_8^2 C_8^1 - C_8^{(2-1)=1}$

# Recursive Factorization(7 ):DCT-II

- Using the trigonometric identities and relations,$[B]_{N/2}$ can be expressed in terms of $[C]_{N/2}$ and a simplified recursive form for $[B]_{N/2}$ follows:

$$[K]_N [C]_N [D]_N$$

$$= \begin{bmatrix} \sqrt{2} & 0 & 0 & 0 & \cdots & 0 \\ -\sqrt{2} & 2 & 0 & 0 & & 0 \\ \sqrt{2} & -2 & 2 & 0 & & 0 \\ -\sqrt{2} & 2 & -2 & 2 & \cdots & \vdots \\ \sqrt{2} & -2 & 2 & -2 & 2 & \\ \vdots & & & & & \ddots \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ C_{4N}^{2k_0\Phi_0} & C_{4N}^{2k_0\Phi_1} & C_{4N}^{2k_0\Phi_2} & \cdots & C_{4N}^{2k_0\Phi_{N-2}} & C_{4N}^{2k_0\Phi_{N-1}} \\ C_{4N}^{2k_1\Phi_0} & C_{4N}^{2k_1\Phi_1} & C_{4N}^{2k_1\Phi_2} & \cdots & C_{4N}^{2k_1\Phi_{N-2}} & C_{4N}^{2k_1\Phi_{N-1}} \\ C_{4N}^{2k_2\Phi_0} & C_{4N}^{2k_2\Phi_1} & C_{4N}^{2k_2\Phi_2} & \cdots & C_{4N}^{2k_2\Phi_{N-2}} & C_{4N}^{2k_2\Phi_{N-1}} \\ \vdots & & & \vdots & & \vdots \\ C_{4N}^{2k_{N-2}\Phi_0} & C_{4N}^{2k_{N-2}\Phi_1} & C_{4N}^{2k_{N-2}\Phi_2} & \cdots & C_{4N}^{2k_{N-2}\Phi_{N-2}} & C_{4N}^{2k_{N-2}\Phi_{N-1}} \end{bmatrix}$$

$$\begin{bmatrix} C_{4N}^{\Phi_0} & 0 & \cdots & 0 \\ 0 & C_{4N}^{\Phi_1} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & C_{4N}^{\Phi_{N-1}} \end{bmatrix} = [B]_N$$

# Recursive Factorization(8):DCT-II

- By using the results obtained from the previous slides , we have a new form for DCT - II matrix

$$[C]_N = [P_r]_N^{-1} [\widetilde{C}]_N [P_c]_N^{-1} = [P_r]_N [\widetilde{C}]_N [P_c]_N$$

$$\left[\widetilde{C}\right]_N = [\mathrm{Pr}]_N [C]_N [Pc]_N$$

$$= \left( \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \begin{bmatrix} C_{N/2} & 0 \\ 0 & B_{N/2} \end{bmatrix} \right)^T = \begin{bmatrix} C_{N/2} & 0 \\ 0 & B_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}$$

$$= \begin{bmatrix} C_{N/2} & 0 \\ 0 & K_{N/2} C_{N/2} D_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}$$

$$= \begin{bmatrix} I_{N/2} & 0 \\ 0 & K_{N/2} \end{bmatrix} \begin{bmatrix} C_{N/2} & 0 \\ 0 & C_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & 0 \\ 0 & D_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \quad (4)$$

# Recursive Factorization(9):DCT-II

- Center diagonal matrix

$$\begin{bmatrix} C_{N/2} & 0 \\ 0 & C_{N/2} \end{bmatrix}$$

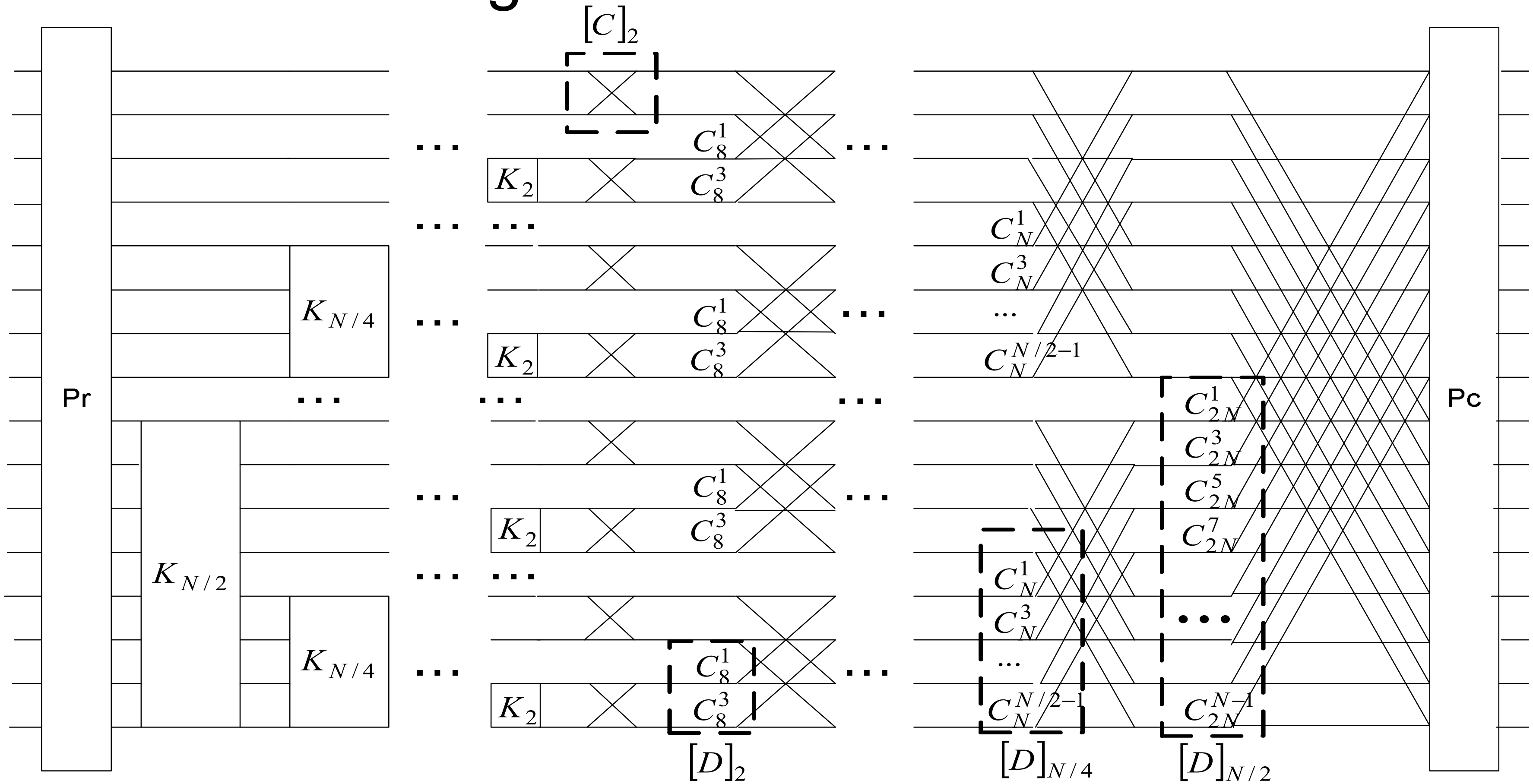
can now be factorized in a recursive manner as

$$\begin{bmatrix} C_{N/2} & 0 \\ 0 & C_{N/2} \end{bmatrix} = \begin{bmatrix} I_{N/4} & 0 & 0 & 0 \\ 0 & K_{N/4} & 0 & 0 \\ 0 & 0 & I_{N/4} & 0 \\ 0 & 0 & 0 & K_{N/4} \end{bmatrix} \begin{bmatrix} [C]_{N/4} & 0 & 0 & 0 \\ 0 & [C]_{N/4} & 0 & 0 \\ 0 & 0 & [C]_{N/4} & 0 \\ 0 & 0 & 0 & [C]_{N/4} \end{bmatrix} \cdot$$

$$\begin{bmatrix} I_{N/4} & 0 & 0 & 0 \\ 0 & D_{N/4} & 0 & 0 \\ 0 & 0 & I_{N/4} & 0 \\ 0 & 0 & 0 & D_{N/4} \end{bmatrix} \begin{bmatrix} I_{N/4} & I_{N/4} & 0 & 0 \\ I_{N/4} & -I_{N/4} & 0 & 0 \\ 0 & 0 & I_{N/4} & I_{N/4} \\ 0 & 0 & I_{N/4} & -I_{N/4} \end{bmatrix}$$

# Recursive Factorization(8):DCT-II

- Data Flow Diagram of DCT-II



# Recursive Factorization(1):DFT

- In a similar way, we can factorized a DFT matrix $[F]_N$ into a recursive form:

$$[\widetilde{F}]_N = [\Pr]_N [F]_N = \left( \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \begin{bmatrix} \widetilde{F}_{N/2} & 0 \\ 0 & E_{N/2} \end{bmatrix} \right)^T$$

$$= \begin{bmatrix} \widetilde{F}_{N/2} & 0 \\ 0 & E_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}$$

$$= \begin{bmatrix} \widetilde{F}_{N/2} & 0 \\ 0 & \Pr_{N/2} \widetilde{F}_{N/2} W_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}$$

$$= \begin{bmatrix} I_{N/2} & 0 \\ 0 & \Pr_{N/2} \end{bmatrix} \begin{bmatrix} \widetilde{F}_{N/2} & 0 \\ 0 & \widetilde{F}_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & 0 \\ 0 & W_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}$$

# Recursive Factorization(2):DFT

- Finally, based on the recursive form we have

$$[F]_N = ([\Pr]_N)^{-1}[\widetilde{F}]_N = ([\Pr]_N)^{-1}\begin{bmatrix} I_{N/2} & 0 \\ 0 & \Pr_{N/2} \end{bmatrix}\begin{bmatrix} \widetilde{F}_{N/2} & 0 \\ 0 & \widetilde{F}_{N/2} \end{bmatrix}.$$

$$\begin{bmatrix} I_{N/2} & 0 \\ 0 & W_{N/2} \end{bmatrix}\begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}$$

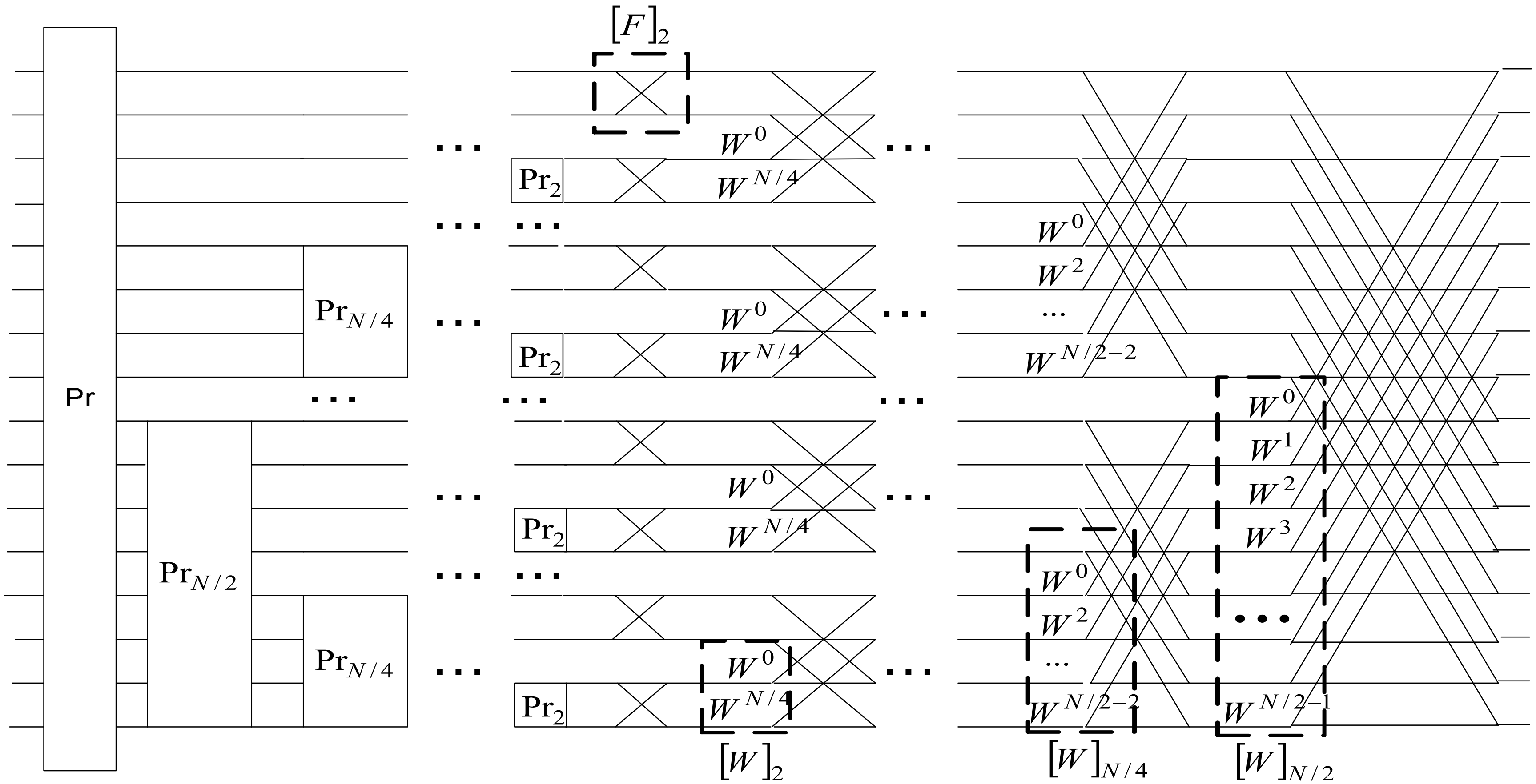
$$= ([\Pr]_N)^{-1}\begin{bmatrix} I_{N/2} & 0 \\ 0 & \Pr_{N/2} \end{bmatrix}\cdots\left[I_{N/4}\otimes\begin{bmatrix} I_2 & 0 \\ 0 & \Pr_2 \end{bmatrix}\right][I_{N/2}\otimes F_2]$$

$$\left[I_{N/4}\otimes\begin{bmatrix} I_2 & 0 \\ 0 & W_2 \end{bmatrix}\right]\left[I_{N/4}\otimes\begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix}\right]..\begin{bmatrix} I_{N/2} & 0 \\ 0 & W_{N/2} \end{bmatrix}\begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}$$

Where $[W]_N = \begin{bmatrix} W^0 & 0 & \cdots & 0 \\ 0 & W^1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & W^{N-1} \end{bmatrix}$

# Recursive Factorization(3):DFT

- Data Flow Diagram of DFT



# Recursive Factorization(1):Wavelet

- The discrete wavelet transform based on the Haar matrix (HWT). Analysis in discrete time: $b = Ax$ and

$$[A]_2 = \begin{bmatrix} r & r \\ r & -r \end{bmatrix} \qquad [A]_2^{-1} = \frac{1}{2}\begin{bmatrix} 1/r & 1/r \\ 1/r & -1/r \end{bmatrix}^T$$

$$\left[\tilde{A}\right]_4 = [Pi]_4 [A]_4 [Pj]_4$$

$$= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} r^2 & r^2 & r^2 & r^2 \\ r^2 & r^2 & -r^2 & -r^2 \\ r & -r & 0 & 0 \\ 0 & 0 & r & -r \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$= r\begin{bmatrix} I_2 & I_2 \\ A_2 & -A_2 \end{bmatrix} = r\left(\begin{bmatrix} I_2 & I_2 \\ I_2 & -I_2 \end{bmatrix}\begin{bmatrix} I_2 & 0 \\ 0 & A_2 \end{bmatrix}\right)^T$$

# Recursive Factorization(2):Wavelet

- In a similar way, we can factorized a Wavelet which based on Haar matrix $[A]_N$ into a recursive form:

$$\left[\tilde{A}\right]_N = \left[Pi\right]_N \left[A\right]_N \left[Pj\right]_N = r\left(\begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}\begin{bmatrix} I_{N/2} & 0 \\ 0 & A_{N/2} \end{bmatrix}\right)^T$$

$$= r\begin{bmatrix} I_{N/2} & 0 \\ 0 & Pi_{N/2}^{-1}\tilde{A}_{N/2}Pj_{N/2}^{-1} \end{bmatrix}\begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}$$

$$= r\begin{bmatrix} I_{N/2} & 0 \\ 0 & Pi_{N/2}^{-1} \end{bmatrix}\begin{bmatrix} I_{N/2} & 0 \\ 0 & \tilde{A}_{N/2} \end{bmatrix}\begin{bmatrix} I_{N/2} & 0 \\ 0 & Pj_{N/2}^{-1} \end{bmatrix}\begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix}$$

# Recursive Factorization(3):Wavelet

- Finally, based on the recursive form we can get the similar formula as following

$$\left[A\right]_N = \left[Pi\right]_N^{-1} \left[\tilde{A}\right]_N \left[Pj\right]_N^{-1}$$

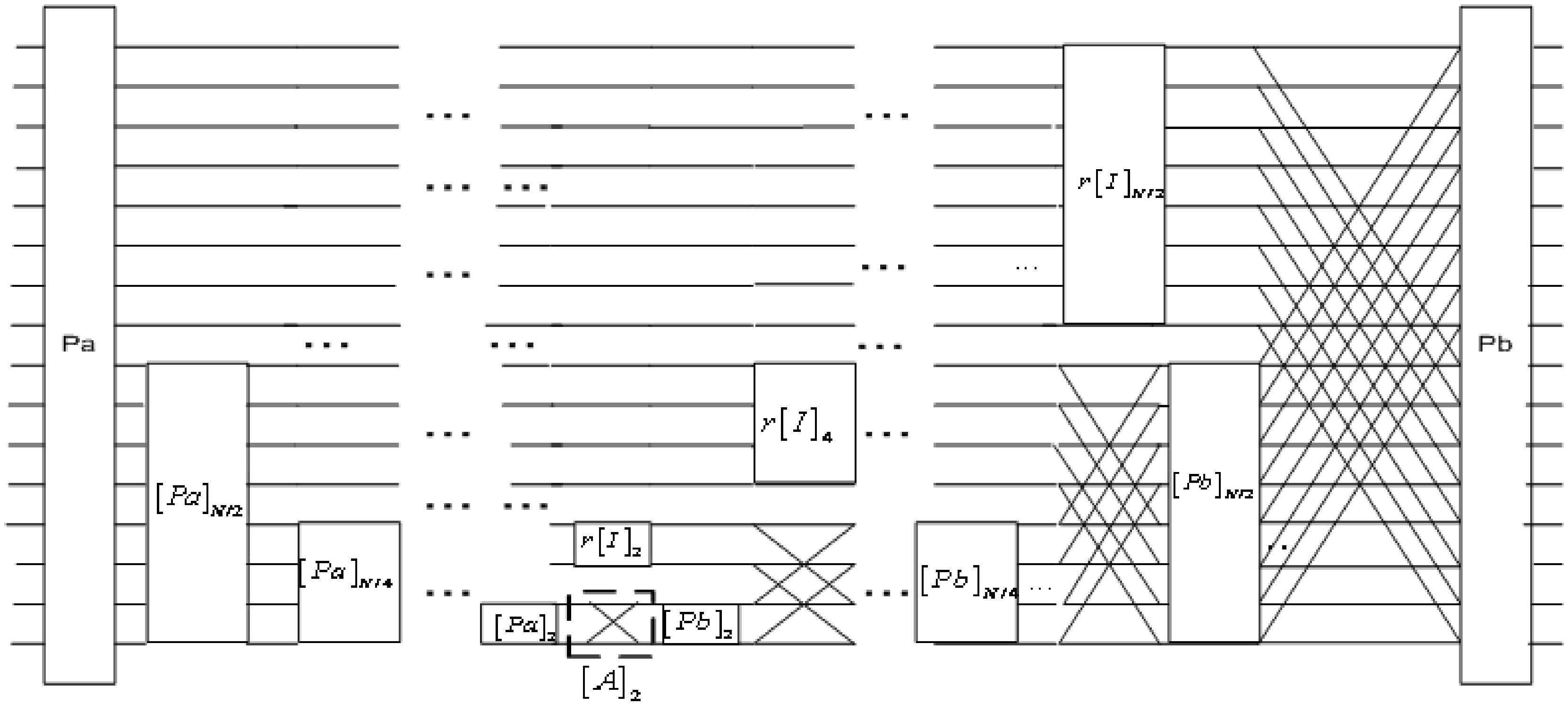
$$= r\left[Pi\right]_N^{-1} \begin{bmatrix} I_{N/2} & 0 \\ 0 & Pi_{N/2}^{-1} \end{bmatrix} \begin{bmatrix} I_{N/2} & 0 \\ 0 & \tilde{A}_{N/2} \end{bmatrix} \begin{bmatrix} I_{N/2} & 0 \\ 0 & Pj_{N/2}^{-1} \end{bmatrix} \begin{bmatrix} I_{N/2} & I_{N/2} \\ I_{N/2} & -I_{N/2} \end{bmatrix} \left[Pj\right]_N^{-1}$$

where

$$\left[Pi\right]_N = \begin{bmatrix} 0 & I_{N/2} \\ I_{N/2} & 0 \end{bmatrix} \qquad \left[Pj\right]_N = \left[\Pr\right]_N \begin{bmatrix} I_{N/2} & 0 \\ 0 & -I_{N/2} \end{bmatrix}$$

# Recursive Factorization(4):Wavelet

- Data Flow Diagram of Wavelet



# Discussions : Dual Use of Cooley-Tukey Type Data Flow Diagram

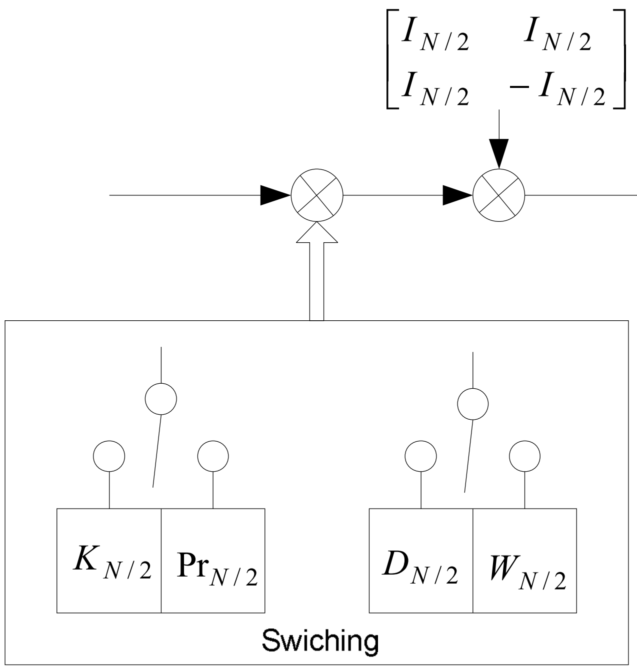
- Computational Complexity

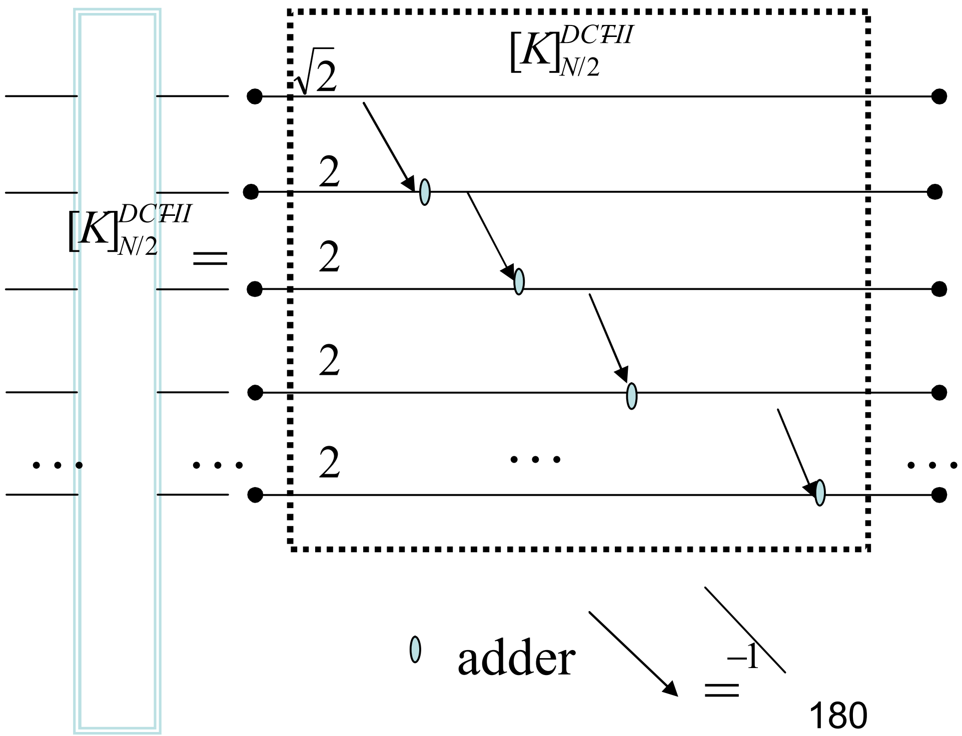
| | | |
|---|---|---|
| FFT | Complex additions : | $N \log_2 N$ |
| Wavelet based on Haar | Real additions : | $\frac{5}{2}N - 2\log_2 N - 2$ |
| DCT-II(Chen Wang) | Real additions : | $3N/2\ (\log_2 N - 1) + 2,\ M \geq 4$ |
| DCT-RF(proposed) | Real additions : | $N \log_2 N + N/2\ - 1$ |

[Chen, Wang ] "High throughput VLSI architecures for the 1-D and 2-D discrete cosine transforms", IEEE Trans. Circuits Syst. Video Technol., 1995

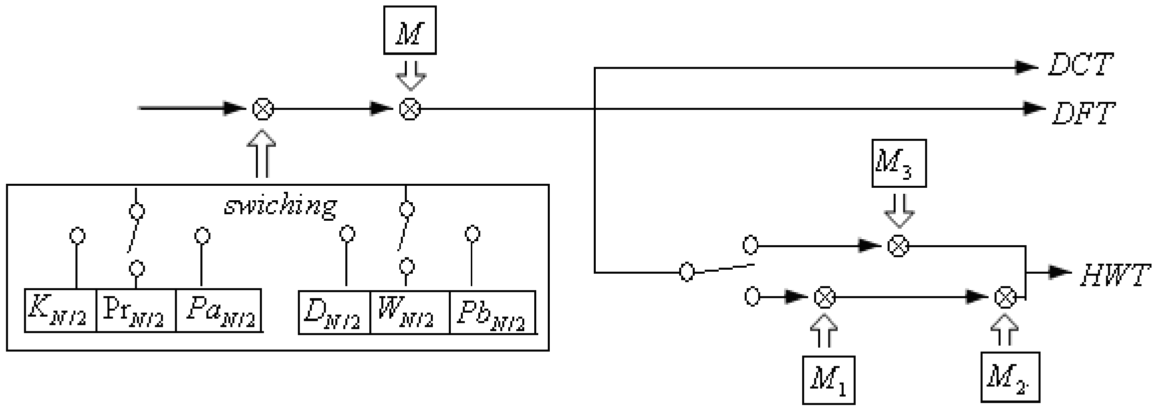
# Discussions : Dual Use of Cooley-Tukey Type Data Flow Diagram

- Hybrid Architecture(switching to DCT or DFT)



# Generalized DFT/DCT/Wavelet by using Jacket Pattern




The DFT and DCT Wavelet matrices can be simply generated by using the Element-wise Inverse Jacket Sparse matrices. Where:

$$[M_1] = [I]_{N/2^h - 1} \otimes \begin{bmatrix} I_{2^{h-1}} & I_{2^{h-1}} \\ I_{2^{h-1}} & -I_{2^{h-1}} \end{bmatrix} \oplus \left[ I_{2^h} \right] \qquad [M_2] = [I]_{N-2^{h+1}} \oplus r[I]_{2^h} \oplus [I]_{2^h}$$

$$[M_3] = [I]_{N/2^k - 1} \otimes \left[ I_{2^{k-1}} \oplus Pa^{-1}_{2^{k-1}} \right] \oplus [I]_{2^k}, (3 \le k \le h) \qquad [M] = [I]_{N/2^h} \otimes \begin{bmatrix} I_{2^{h-1}} & I_{2^{h-1}} \\ I_{2^{h-1}} & I_{2^{h-1}} \end{bmatrix}$$

# Conclusions

- Hybrid architecture achieves dual use of data flow structure in DCT and DFT/Wavelet computation.
- Computational complexity of the proposed scheme is comparable to FFT, Chen’s DCT, and Wang’s DCT algorithms.
- Further investigation is needed for dual use of FFT architecture in other trigonometric transform computation.

# THANK YOU FOR ATTENTION!