

고속 리버스 자켓 변환과 그의 응용

정회원 이승래*, 성평모**

Fast Reverse Jacket Transform and Its Application

Seung-Rae Lee*, Koeng-Mo Sung** *Regular Members*

요 약

개선된 리버스 자켓 행렬 (Reverse Jacket matrix)의 정의와 함께 그의 역행렬을 소개한다. 새로이 정의된 리버스 자켓 행렬은 실베스터 타입의 하다마드 행렬을 이용하여 더욱 일반화되었다. 이 논문에서는 고속 리버스 자켓 변환 (fast Reverse Jacket transform)을 제시하며 또한 이 알고리즘이 4 점 이산 푸리에 변환으로 응용이 됨을 보여준다.

ABSTRACT

With revised definition of the Reverse Jacket matrix (RJM) the inverse of the RJM is presented. This new defined RJM is more generalized using the Hadamard matrix of Sylvester type. In this paper, we propose a fast Reverse Jacket transform (FRJT) and also show its application to 4-point discrete Fourier transform (DFT).

1. 서 론

참고문헌 [10]에서 최초로 소개된 리버스 자켓 행렬은 ± 1 만을 행렬요소로 갖는 하다마드 행렬과는 달리, 일부 혹은 모든 요소들에 대해 임의의 실수 또는 복소수를 사용할 수 있다는 장점을 갖는다. 이 중 특히, 중앙의 요소만을 정수로 대체한 중앙가중치 하다마드 변환(Center-weighted Hadamard transform, 간략히, CWHT)이 소개되었다^[8].

리버스 자켓 행렬의 특이한 현상은 그의 역행렬을 구할 때 본 행렬 (original matrix)의 요소들이 그룹을 지어 위치들이 이동한 형상으로 나타나 마치 등산복을 뒤집어 입은 것처럼 보인다 하여 그의 이름을 Reverse Jacket 행렬이라 칭하였다. 이 논문에서는, [10]에서 리버스 자켓 행렬의 정의가 단지 이 행렬의 부분적 성질만을 내포한 관계로 새롭게 정의하였으며 더욱, [8], [9]에서 명확히 밝히지 못했던 역행렬을 구하는 문제를 제2장에서 정리를 통해 소개하였다. 이 역행렬은 그 구조가 처음 행렬을 구성할 때와 동일한 구조를 갖는 장점을 갖고 있어

매우 쉽게 구할 수 있다. 더욱 주지할 점은 리버스 자켓 행렬을 처음 구성하는데 사용된 기본행렬 (basic matrix)이 그간 대칭행렬에 국한된 이전과는 달리 이번에는 비대칭 행렬을 기본행렬으로 사용할 수 있어, 교대대칭 (skew-symmetric)인 행렬도 갖는 하다마드 행렬의 전 클래스를 포함한다는 것이다. 하다마드 변환은 직교변환 (orthogonal transform)이어서 이미 여러 분야, 예를 들어, 정보이론, 영상 또는 신호처리 등에서 응용되고 있다^[1, 2, 5, 6, 16].

비정현 (non-sinusoidal) 직교 (orthogonal) 변환인 월쉬-하다마드 변환 (Walsh-Hadamard transform)은 신호처리와 부호이론에 사용되어 왔으며^{[3],[11]}, 최근 복소수 리버스 자켓 변환을 쌍극 시퀀스 (bipolar sequence)에 활용한 바 있다^[12].

리버스 자켓 행렬을 분해 (decomposition)하여 개발한 고속 리버스 자켓 변환 (FRJT)은 중앙가중치 하다마드 변환보다 훨씬 더 빠른 알고리즘이다. 리버스 자켓 행렬은 하다마드 행렬을 일반화된 형태를 갖고 있어 고속 푸리에 변환과도 밀접한 관계가 있다. 이러한 관계를 한눈에 보여주는 예제를 제시

* 서울대학교 뉴미디어통신공동연구소 (srlee@acoustics.snu.ac.kr),
논문번호 : K01002-0105, 접수일자 : 2001년 1월 5일

** 서울대학교 전기공학부 (kmsung@acoustics.snu.ac.kr)

하여 시각화하고자 한다.

본 논문은 다음과 같이 구성되었다. 제 2장에서는 개선된 리버스 자켓 행렬의 정의와 역행렬을 구하는 문제를 해결한 정리 그리고 고속 푸리에 변환과의 관계를 나타내는 정리를 보여준다. 제3장에서는 이 행렬을 분해하여 얻은 고속 리버스 자켓 변환과 그의 응용을 소개하며 제4장에서는 끝으로 결론과 향후 전망에 대하여 논의한다.

II. 개선된 리버스 자켓 행렬

이 장에서는 역행렬이 뒤집어지는 기하학적 구조를 갖는 리버스 자켓 행렬을 구성한다. 이 행렬은 하다마드 행렬의 요소들을 일반화한다. 안팎으로 뒤집어 입을 수 있는 등산복처럼 리버스 자켓의 역행렬을 구할 때 적어도 두 개의 행렬요소의 위치가 바뀐다. 예를 들어, 중앙 원으로부터 바깥쪽으로, 그리고 바깥쪽에서 안쪽으로 뒤바뀌는 형태가 된다. 이러한 기이한 현상으로 말미암아 이 행렬을 리버스 자켓 행렬이라 칭하였다^[10].

주어진 2×2 기본행렬 $R_2 \in R^{2 \times 2}$ 에 대하여 가역 (invertible)인 4×4 행렬을 구성한다.

I_{2^j} 와 O_{2^j} , $j \in N \cup 0$ 은 각각 $(2^j \times 2^j)$ 단위행렬과 영 행렬이며, R, N 은 각각 실수와 자연수의 집합을 칭한다.

정의 1: a, b, c, d 를 영이 아닌 실수 또는 복소수라 하자. 리버스 자켓 행렬은 귀납적으로 (recursively) 다음과 같이 정의된다.

$$R_{2^{k+1}} = \begin{bmatrix} R_{2^k} & Z_{2^k} R_{2^k} S_{2^k} \\ S_{2^k} R_{2^k} Z_{2^k} & -J_{2^k} R_{2^k} J_{2^k} \end{bmatrix}.$$

여기서 초기조건은 $k \geq 1$ 에 대하여

$$R_2 := \begin{bmatrix} a & b \\ c & -d \end{bmatrix}, Z_{2^k} := \begin{bmatrix} I_{2^{\alpha-1}} & O_{2^{\alpha-1}} \\ O_{2^{\alpha-1}} & -I_{2^{\alpha-1}} \end{bmatrix},$$

$$S_{2^k} := \begin{bmatrix} O_{2^{\alpha-1}} & I_{2^{\alpha-1}} \\ I_{2^{\alpha-1}} & O_{2^{\alpha-1}} \end{bmatrix}$$

$$\text{그리고 } J_{2^k} := \begin{bmatrix} O_{2^{\alpha-1}} & I_{2^{\alpha-1}} \\ -I_{2^{\alpha-1}} & O_{2^{\alpha-1}} \end{bmatrix} \text{이다.}$$

Remark 2: 우리는 여기서

$$S_2 S_2 = Z_2 Z_2 = I_2, J_2 S_2 = Z_2 \text{ 그리고}$$

$$S_2 J_2 = -Z_2 \text{ 임을 쉽게 알 수 있다.}$$

Remark 3: $A^{(k)} := R_{2^k}$, $k \geq 1$ 이라 하면 $k \geq 1$ 에 대하여 다음 등식이 성립한다.

$$\begin{aligned} A^{(k+1)} &= \begin{bmatrix} \overline{A^k} & \overline{B^k} \\ \overline{C^k} & \overline{-D^k} \end{bmatrix} \\ &= \begin{bmatrix} \overline{A^{(k-1)}} & \overline{B^{(k-1)}} & \overline{B^{(k-1)}} & \overline{A^{(k-1)}} \\ \overline{C^{(k-1)}} & \overline{-D^{(k-1)}} & \overline{D^{(k-1)}} & \overline{-C^{(k-1)}} \end{bmatrix} \\ &= \begin{bmatrix} \overline{C^{(k-1)}} & \overline{D^{(k-1)}} & \overline{-D^{(k-1)}} & \overline{-C^{(k-1)}} \\ \overline{A^{(k-1)}} & \overline{-B^{(k-1)}} & \overline{-B^{(k-1)}} & \overline{A^{(k-1)}} \end{bmatrix} \\ &= R_{2^{k+1}}. \end{aligned}$$

여기서 초기조건은

$$\begin{aligned} A^{(0)} &= a, B^{(0)} = b, C^{(0)} = c, D^{(0)} = d, \\ \text{즉, } A^{(1)} &:= \begin{bmatrix} a & b \\ c & -d \end{bmatrix}, B^{(k)} := Z_{2^k} A^{(k)} S_{2^k}, \\ C^{(k)} &:= S_{2^k} A^{(k)} Z_{2^k}, \\ \text{그리고 } B^{(k)} &:= -J_{2^k} A^{(k)} J_{2^k} \text{이다.} \end{aligned}$$

Remark 4: a, d 가 실수라 하면 기본행렬 R_2 에 대해 크게 5 가지 경우로 볼 수 있다.

- i): $c = b \in R \cup C$, 즉, R_2 는 대칭행렬이다.
- ii): $c = -b \in R \cup C$.
- iii): $c = \bar{b} \in C$, 즉, R_2 는 허미치안 (Hermitian) 이다.
- iv): $c = -\bar{b} \in C$.

여기서 C 는 복소수 집합을 칭한다.

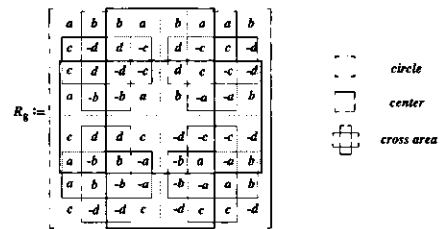
주어진 기본행렬

$$\begin{aligned} R_2 &:= \begin{bmatrix} a & b \\ c & -d \end{bmatrix} = \begin{bmatrix} \overline{A^{(0)}} & \overline{B^{(0)}} \\ \overline{C^{(0)}} & \overline{-D^{(0)}} \end{bmatrix} \text{에 대해} \\ &= A^{(1)} \end{aligned}$$

여 정의에 따라서 귀납적으로 확장하면 다음 그림 Fig. 1, Fig. 2와 같다.

$$R_4 := \begin{bmatrix} a & b & b & a \\ c & d & d & -c \\ c & d & -d & -c \\ a & -b & -b & a \end{bmatrix} =: \begin{bmatrix} A^{(1)} & B^{(1)} \\ C^{(1)} & -D^{(1)} \end{bmatrix} =: A^{(2)}$$

Fig. 1



$$=: \begin{bmatrix} \overline{A^{(1)}} & \overline{B^{(1)}} & \overline{B^{(1)}} & \overline{A^{(1)}} \\ \overline{C^{(1)}} & \overline{D^{(1)}} & \overline{D^{(1)}} & \overline{-C^{(1)}} \\ \overline{C^{(1)}} & \overline{D^{(1)}} & \overline{-D^{(1)}} & \overline{-C^{(1)}} \\ \overline{A^{(1)}} & \overline{-B^{(1)}} & \overline{-B^{(1)}} & \overline{A^{(1)}} \end{bmatrix} =: \begin{bmatrix} \overline{A^{(2)}} & \overline{B^{(2)}} \\ \overline{C^{(2)}} & \overline{-D^{(2)}} \end{bmatrix} =: A^{(3)}$$

Fig. 2

보조정리 5: 리버스 자켓 행렬 $R_{2^k} \in R_{2^k}$, $k \geq 1$ 은 가역 (invertible)이다.

증명: 치환행렬 $L_4 := \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ 와 상삼각전치행

렬 $M_4 := \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ 를 정의하면

$$N_4 := L_4 R_4 M_4 = \begin{bmatrix} c & -d & 0 & 0 \\ c & d & 0 & 0 \\ a & b & 2b & 2a \\ a & -b & -2b & 2a \end{bmatrix} \\ \equiv \begin{bmatrix} N_{11} & 0 \\ N_{21} & N_{22} \end{bmatrix}$$

이 성립한다. $\det(L_4)$ 와 $\det(M_4)$ 는 모두 1이므로 다음 식이 성립한다.

$$\det(R_4) = \det(L_4) \cdot \det(R_4) \cdot \det(M_4) \\ = \det(N_4) = \det(N_{11}) \cdot \det(N_{22})$$

리버스 자켓 행렬 정의에서 $a, b, c, d \neq 0$ 이므로 $\det(N_{11})$ 과 $\det(N_{22})$ 도 모두 0

이 아니다. 그러므로

$$L_{2^k} = L_{2^{k-1}} \otimes I_2, M_{2^k} = M_{2^{k-1}} \otimes I_2, k \geq 2 \text{ 일 때}$$

$$\det(R_{2^k}) = \det(L_{2^k}) \cdot \det(R_{2^k}) \cdot \det(M_{2^k}) \\ = \det(N_{2^k}) \neq 0$$

립함을 쉽게 확인할 수 있다. 따라서 R_{2^k} , $k \geq 2$ 는 정칙행렬이다.

다음의 두 정리와 따름정리를 증명하기 위하여 잘 알려진 보조정리를 소개한다.

보조정리 6: (Matrix inversion lemma)^[4] 소행렬 (submatrix) M_{11} 과 M_{22} 가 각각 $n \times n$ 그리고 $m \times m$ 정칙행렬인 다음과 같이 행렬

$$M := \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

에서 소행렬

$$(M_{11} - M_{12} M_{22}^{-1} M_{21}) \text{과 } (M_{22} - M_{21} M_{11}^{-1} M_{12})$$

또한 정칙행렬이라면 행렬 M 은 정칙행렬이고 그 역행렬 M^{-1} 은 다음과 같다.

$$M^{-1} = \begin{bmatrix} M_{11}^{-1} + M_{11}^{-1} M_{12} (M_{22} - M_{21} M_{11}^{-1} M_{12})^{-1} M_{21} M_{11}^{-1} & -M_{11}^{-1} M_{12} (M_{22} - M_{21} M_{11}^{-1} M_{12})^{-1} \\ -M_{21} M_{11}^{-1} M_{12} (M_{22} - M_{21} M_{11}^{-1} M_{12})^{-1} & (M_{22} - M_{21} M_{11}^{-1} M_{12})^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} (M_{11} - M_{12} M_{22}^{-1} M_{21})^{-1} & -(M_{11} - M_{12} M_{22}^{-1} M_{21})^{-1} M_{12} M_{22}^{-1} \\ -M_{22}^{-1} M_{21} (M_{11} - M_{12} M_{22}^{-1} M_{21})^{-1} & M_{22}^{-1} + M_{22}^{-1} M_{21} (M_{11} - M_{12} M_{22}^{-1} M_{21})^{-1} M_{12} M_{22}^{-1} \end{bmatrix}$$

정리 7: R_{2^k} 가 정칙행렬이라 하자. 리버스 자켓 행렬

$$R_{2^{k+1}} = \begin{bmatrix} R_{2^k} & Z_{2^k} R_{2^k} S_{2^k} \\ S_{2^k} R_{2^k} Z_{2^k} & -J_{2^k} R_{2^k} J_{2^k} \end{bmatrix}$$

이면 $R_{2^{k+1}}$ 는 정칙행렬이며

$$R_{2^{k+1}}^{-1} =$$

$$\begin{bmatrix} \check{R}_{2^k}^{-1} & Z_{2^k} \check{R}_{2^k}^{-1} S_{2^k} \\ S_{2^k} \check{R}_{2^k}^{-1} Z_{2^k} & -J_{2^k} \check{R}_{2^k}^{-1} J_{2^k} \end{bmatrix}$$

이다. 이때

$$\check{R}_{2^k} = R_{2^k} + Z_{2^k} R_{2^k} S_{2^k} J_{2^k} R_{2^k}^{-1} J_{2^k} S_{2^k} R_{2^k} Z_{2^k}, k \geq 1.$$

증명: 부록을 참조

정리 8: R_{2^k} 가 정칙행렬이라 하자. 다음 행렬

$$W_{2^{k+1}} = \begin{bmatrix} R_{2^k} & Z_{2^k} R_{2^k} \\ R_{2^k} Z_{2^k} & Z_{2^k} R_{2^k} Z_{2^k} \end{bmatrix}$$

이면 $W_{2^{k+1}}$ 는 정칙행렬이며

$$W_{2^{k+1}}^{-1} = \begin{bmatrix} \hat{R}_{2^k}^{-1} & Z_{2^k} \hat{R}_{2^k}^{-1} \\ \hat{R}_{2^k}^{-1} Z_{2^k} & Z_{2^k} \hat{R}_{2^k}^{-1} Z_{2^k} \end{bmatrix}$$

이다. 이때

$$\hat{R}_{2^k} = R_{2^k} - Z_{2^k} R_{2^k} Z_{2^k} R_{2^k}^{-1} Z_{2^k} R_{2^k} Z_{2^k}, k \geq 1.$$

증명: 정리 7의 증명을 참조

따름정리 9: $k \geq 1$ 에 대하여 직교행렬 $U_{2^{k+1}}$ 이 존재하면

$$R_{2^{k+1}} = U_{2^{k+1}} W_{2^{k+1}} U_{2^{k+1}}^{-1}$$

증명: $U_{2^{k+1}} := \begin{bmatrix} I_{2^k} & O_{2^k} \\ O_{2^k} & S_{2^k} \end{bmatrix}$ 라 하자. 이 때

I_{2^k} , O_{2^k} 는 각각 차원이 2^k , $k \geq 1$ 인 단위행렬과 영행렬이다. 먼저 차원이 4인 경우를 생각해 보고 고 차원일 경우는 유사하게 증명할 수 있다. 이 경우, Remark 2를 참조하면 $U_4^T = U_4 = U_4^{-1}$ 임을 쉽게 확인할 수 있다. 여기서 T는 전치행렬을 나타낸다.

$$\begin{aligned}
 & U_4 \cdot R_4 \cdot U_4 \\
 = & \begin{bmatrix} I_2 & O_2 \\ O_2 & S_2 \end{bmatrix} \begin{bmatrix} R_2 & Z_2 R_2 S_2 \\ S_2 R_2 Z_2 & -J_2 R_2 J_2 \end{bmatrix} \begin{bmatrix} I_2 & O_2 \\ O_2 & S_2 \end{bmatrix} \\
 = & \begin{bmatrix} R_2 & Z_2 R_2 S_2 \\ S_2 S_2 R_2 Z_2 & -S_2 J_2 R_2 J_2 \end{bmatrix} \begin{bmatrix} I_2 & O_2 \\ O_2 & S_2 \end{bmatrix} \\
 = & \begin{bmatrix} R_2 & Z_2 R_2 S_2 \\ S_2 S_2 R_2 Z_2 & -S_2 J_2 R_2 J_2 S_2 \end{bmatrix} \\
 = & \begin{bmatrix} R_2 & Z_2 R_2 \\ R_2 Z_2 & Z_2 R_2 Z_2 \end{bmatrix} \\
 = & W_4.
 \end{aligned}$$

이로써 따름정리의 증명이 이루어졌다.

III. 고속 리버스 자켓 변환과 그의 응용

이 장에서는 고속 하다마드 변환이나 고속 중앙 가중치 하다마드 변환 [7]을 일반화한 알고리즘을 유도하기로 한다. 이 알고리즘을 위해 우선 두 개의 치환행렬을 정의하자.

$$\begin{aligned}
 P_{2^k} &:= P_4 \otimes I_{2^{k-2}}, \quad k \geq 2, \\
 P_4 &:= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},
 \end{aligned}$$

$$Q_{2^k} := [I_{2^{k-1}} \oplus (S_2 \otimes I_{2^{k-2}})].$$

여기서 \otimes 는 Kronecker product를 나타내며 \oplus 는 direct sum을 표시한다.

리버스 자켓 변환 알고리즘은 다음과 같은 수식으로 나타낸다.

$$\begin{aligned}
 R_{2^k} &= P_{2^k}^T \underbrace{(H_{2^{k-1}} \oplus H_{2^{k-1}})}_{\text{스태이지 3}} \underbrace{((U_A \otimes I_{2^{k-2}}) \oplus (U_B \otimes I_{2^{k-2}}))}_{\text{스태이지 2}} \\
 &\quad \underbrace{(H_2 \otimes I_{2^{k-1}})}_{\text{스태이지 1}} Q_{2^k}^T
 \end{aligned}$$

여기서

$$U_A := \text{diag}(a, b) \text{ 이고 } U_B := \text{diag}(c, d)$$

이다.

그러면 FRJT 알고리즘의 각 스테이지에서의 연산량을 조사해 보자. 우선 치환행렬은 연산량에 영향을 주지 않으므로 $P_{2^k}^T$ 과 $Q_{2^k}^T$ 의 연산량을 무시하기로 하자.

$N = 2^k$ 일 때 연산량을 계산한 도표 Tab. 1은 아래와 같으며 FRJT가 CWHT보다 훨씬 효율적이며 더욱 일반화 되었음을 한 눈에 알 수 있다.

따름 정리 9는 리버스 자켓 행렬과 4점 DFT 행렬과의 관계를 보여 주었다. 즉, 4점 FFT를 FRJT로 전환하여 연산할 수 있다. 다음의 예제는 FRJT가 4점 FFT에 응용될 수 있음을 잘 보여준다.

예제 10:

$$W_4 := \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

는 널리 알려진 4점 DFT 행렬이다.

$$R_2 := \begin{bmatrix} 1 & 1 \\ 1 & -j \end{bmatrix}, \quad U_4 := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

는 각각 기본행렬과 리버스 자켓 행렬을 위한 치환행렬이다. 따름정리 9에 의해

$$R_4 = U_4 \cdot W_4 \cdot U_4, \quad W_4 = U_4 \cdot R_4 \cdot U_4 \text{이다.}$$

따라서 FRJT 알고리즘을 적용하고

$$T_4 := U_4 \cdot P_4, \quad Q_4^T \cdot U_4 = (U_4)^2 = I_4 \text{이면}$$

W_4 와 R_4 를 다음과 같이 나타낸다.

$$\begin{aligned}
 W_4 &= U_4 \cdot P_4^T (H_2 \oplus H_2) (U_A \oplus U_B) \\
 &\quad \cdot (H_2 \otimes I_2) Q_4^T \cdot U_4 \\
 &= T_4 (H_2 \oplus H_2) (U_A \oplus U_B) (H_2 \otimes I_2)
 \end{aligned}$$

Tab. 1

알고리즘 \ 스테이지	CWHT ($a = b = c = 1$)		FRJT ($a = b = c = 1$)		FRJT ($a, b, c, d \neq 1$)	
	덧셈연산	곱셈연산	덧셈연산	곱셈연산	덧셈연산	곱셈연산
첫번째	생 략		N		N	
두번째				$\frac{N}{4}$		N
세번째				$N \log_2 \frac{N}{2}$		$N \log_2 \frac{N}{2}$
총 연산량			$kN + \frac{N}{2}$	$\frac{3N}{2}$	$N \log_2 N$	$\frac{N}{4}$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & j \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

$$R_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & j & -1 \\ 1 & j & -j & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

행렬 형태의 이 알고리즘은 각 스테이지에서의 연산을 명백하게 해주며 8 번의 덧셈과 한 번의 복소수 곱셈을 필요로 한다. 그러나 8 점 이상의 DFT 행렬은 따름정리 9에 명시된 행렬 W_N , $N \geq 8$ 이 아님을 유의하기 바란다.

IV. 결론

리버스 자켓 행렬을 처음 구성하는데 사용된 기본행렬 (basic matrix)의 요소를 정수만이 아닌 실수와 복소수를 사용할 수 있는 일반화된 리버스 자켓 행렬을 정의함으로써 이 행렬의 더욱 좋은 성질들이 연구될 전망이다. 특히, 그간 대칭행렬에 국한된 이전 [9, 10]과는 달리 이번에는 비대칭 행렬을 기본행렬로 사용할 수 있어 교대대칭(skew-symmetric)인 행렬도 허용하여 모든 클래스의 하다마드 행렬을 포함하게 되었으며 그간 미해결의 역행렬을 구하는 문제를 해결하였다. 새로운 고속 리버스 자켓 변환 알고리즘을 소개하였고 아울러 고속 푸리에 변환과의 관계를 나타내는 정리와 예제를 보여주었다.

향후 리버스 자켓 행렬의 성질이 좀 더 연구되어야 하고, N-점 FRJT와 N-점 FFT와의 관계를 명확히 밝혀야 하는 문제가 남아 있으며 또한 역행렬을 양직 (explicit)으로 연산하는 연구가 요청된다.

앞으로, 리버스 자켓 행렬과 변환 알고리즘이 하다마드 행렬 [1, 12, 13, 16]과 더불어 신호처리, 영상처리 및 정보이론 [1, 2, 5, 6, 16], 부호론 [12, 13], 필터뱅크 [14, 15] 등의 분야에서 응용이 될 전망이다.

참고 문헌

[1] S.S. Aghaian, "Hadamard Matrices and Their Applications", *Lectures Notes in Mathematics*,

1168, 1985.

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

[3] K. G. Beauchamp, "Applications of Walsh and related functions", *Academic Press*, 1984.

[4] C. K. Chui and G. Chen, "Kalman Filtering with Real-Time Applications", *Springer Verlag*, 1991.

[5] D. F. Eallaot and K. R. Rao, "Fast Transforms, Algorithms, Applications", *New York: Academic*, 1982.

[6] O. Ersoy, "Fourier-Related Transforms, Fast Algorithms and Applications", *Prentice Hall International Editions*, 1997.

[7] C.-P. Fan and J.-F. Yang, "Fast Center-weighted Hadamard Transform Algorithms", *IEEE Trans. on Circuits and Systems-II*, Vol. 45, No. 3, pp. 429-432, 1998.

[8] M. H. Lee, "The Center-weighted Hadamard Transform", *IEEE Trans. on Circuits and Systems-II*, Vol. 36, No. 9, pp. 1247-1249, 1989.

[9] M. H. Lee, "A New Reverse Jacket Transform and Its Fast Algorithm", *IEEE Trans. on Circuits and Systems-II*, Vol. 47, No. 1, pp. 39-47, 2000.

[10] S. -R. Lee and M. H. Lee, "On the Reverse Jacket Matrix for Weighted Hadamard Transform", *IEEE Trans. on Circuits and Systems-II*, Vol. 45, No. 3, pp. 436-441, 1998.

[11] F. J. Macwilliams and N. J. Sloane, "The theory of error-correcting codes", *North-Holland Publ. Comp.*, 1992.

[12] M. G. Parker and M. H. Lee, "Optimal Bipolar Sequences for the Complex Reverse-Jacket Transform", *Int. Symp. on Information Theory and Its Appl.* Honolulu, Hawaii, U.S.A., Nov. 5-8, 2000.

[13] M. -H. Shin, H. -Y. Song and J. -S. No, "Generalized Sylvester Construction for Hadamard Matrices", *한국통신학회논문집 제25권 제3A호*, pp. 412-416, 2000.

[14] E. Viscito and P. Allebach, "The Analysis and Design of Multidimensional FIR Perfect

Reconstruction Filter Banks for Arbitrary Sampling Lattices”, *IEEE Trans. on Circuits and Systems*, Vol. 38, No. 1, pp. 29-41, 1991.

[15] P. P. Vaidyanathan, “Multirate Systems and Filter Banks”, *Prentice Hall Signal Processing Series*, 1993.

[16] R. K. R. Yarlagadda and J. E. Hershey, “Hadamard Matrix Analysis and Synthesis”, *Kluwer Academic Publishers*, 1997.

부 록

정리 7의 증명: 4×4 리버스 자켓 행렬의 역이 가능하면 정의 1과 보조정리 5에 따라 고차원의 리버스 자켓 행렬은 항상 정칙행렬이다. 따라서 4×4 행렬의 증명만을 다루고자

한다. 한편 보조정리 6의 가정과 같은 방식으로 두 행렬

$$N_1 := R_2 + Z_2 R_2 S_2 J_2 R_2^{-1} J_2 S_2 R_2 Z_2 \text{ 과}$$

$$Y_1 := -J_2 R_2 J_2 - S_2 R_2 Z_2 R_2^{-1} Z_2 R_2 S_2$$

이 모두 가역이면 $k \geq 2$ 에 대하여 두 행렬

$$N_k := R_{2^k} + Z_{2^k} R_{2^k} S_{2^k} J_{2^k} R_{2^k}^{-1} J_{2^k} S_{2^k} R_{2^k} Z_{2^k} \text{ 과}$$

$$Y_k := -J_{2^k} R_{2^k} J_{2^k} - S_{2^k} R_{2^k} Z_{2^k} R_{2^k}^{-1} Z_{2^k} R_{2^k} S_{2^k}$$

이 가역이다.

$a, b, c, d \neq 0$ 이고 $\det R_2 \neq 0$ 이면 $k \geq 1$ 에 대해 두 행렬 N_k, Y_k 가 가역임을 쉽게 확인할 수 있다. 첫 번째 이 조건은 바로 정의 1에서 충족된다. 조건에 따라

$$T_1 := -J_2 R_2 J_2 - S_2 R_2 Z_2 R_2^{-1} Z_2 R_2 S_2$$

로 놓자. 보조정리 6을 이용하면

$$R_4^{-1} = \begin{bmatrix} \check{R}_2^{-1} & -R_2^{-1} Z_2 R_2 S_2 T_1^{-1} \\ -T_1^{-1} S_2 R_2 Z_2 R_2^{-1} & T_1^{-1} \end{bmatrix}$$

이다. 그러므로 다음 세 가지 사항을 증명하면 된다.

- i) $-R_2^{-1} Z_2 R_2 S_2 T_1^{-1} = Z_2 \check{R}_2^{-1} S_2$,
- ii) $-T_1^{-1} S_2 R_2 Z_2 R_2^{-1} = S_2 \check{R}_2^{-1} Z_2$,
- iii) $T_1^{-1} = -J_2 \check{R}_2^{-1} J_2$.

Remark 2를 이용하면 다음 등식이 성립함을 알 수 있다.

$$-T_1 S_2 R_2^{-1} Z_2 R_2$$

$$= -(-J_2 R_2 J_2 - S_2 R_2 Z_2 R_2^{-1} Z_2 R_2 S_2) S_2 R_2^{-1} Z_2 R_2$$

$$= J_2 R_2 J_2 S_2 R_2^{-1} Z_2 R_2 + S_2 R_2 Z_2 R_2^{-1} Z_2 R_2 R_2^{-1} Z_2 R_2$$

$$= S_2 S_2 J_2 R_2 Z_2 R_2^{-1} Z_2 R_2 Z_2 + S_2 R_2 Z_2$$

$$= S_2 (R_2 + S_2 J_2 R_2 Z_2 R_2^{-1} Z_2 R_2 Z_2) Z_2$$

$$= S_2 \check{R}_2 Z_2.$$

따라서

$$-T_1 S_2 R_2^{-1} Z_2 R_2 = S_2 \check{R}_2 Z_2.$$

위 식의 양변의 역을 취하면

$$-R_2^{-1} Z_2 R_2 S_2 T_1^{-1} = Z_2 \check{R}_2^{-1} S_2.$$

이는 곧 증명해야 할 사항 i)이다.

이제 ii)를 증명하자.

$$-R_2 Z_2 R_2^{-1} S_2 T_1$$

$$= -R_2 Z_2 R_2^{-1} S_2 (-J_2 R_2 J_2 - S_2 R_2 Z_2 R_2^{-1} Z_2 R_2 S_2)$$

$$= R_2 Z_2 R_2^{-1} S_2 J_2 R_2 J_2 + R_2 Z_2 R_2^{-1} S_2 S_2 R_2 Z_2 R_2^{-1} Z_2 R_2 S_2$$

$$= Z_2 Z_2 R_2 Z_2 R_2^{-1} S_2 J_2 R_2 J_2 + R_2 Z_2 Z_2 R_2^{-1} Z_2 R_2 S_2$$

$$= Z_2 (Z_2 R_2 Z_2 R_2^{-1} S_2 J_2 R_2 J_2 + R_2 S_2)$$

$$= Z_2 (R_2 + Z_2 R_2 (-S_2 J_2) R_2^{-1} (-J_2 S_2) R_2 J_2 S_2) S_2$$

$$= Z_2 (R_2 + Z_2 R_2 (S_2 J_2) R_2^{-1} (J_2 S_2 R_2 Z_2) S_2)$$

$$= Z_2 \check{R}_2 S_2.$$

그러므로,

$$-R_2 Z_2 R_2^{-1} S_2 T_1 = Z_2 \check{R}_2 S_2.$$

위 식의 양변의 역을 취하면 곧 ii)이다.

$$-T_1^{-1} S_2 R_2 Z_2 R_2^{-1} = S_2 \check{R}_2^{-1} Z_2.$$

다시 Remark 2의 사항을 이용하면 유사한 방법으로 다음 등식을 얻는다.

$$-J_2 \check{R}_2 J_2$$

$$= -J_2 (R_2 + Z_2 R_2 S_2 J_2 R_2^{-1} J_2 S_2 R_2 Z_2) J_2$$

$$= -J_2 R_2 J_2 - J_2 Z_2 R_2 S_2 J_2 R_2^{-1} J_2 S_2 R_2 Z_2 J_2$$

$$= -J_2 R_2 J_2 - S_2 R_2 Z_2 R_2^{-1} Z_2 R_2 S_2$$

$$= T_1.$$

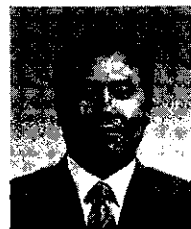
이는 곧 iii)식을 이끈다.

$$T_1^{-1} = -J_2 \check{R}_2^{-1} J_2.$$

이로써 이 정리의 증명을 완결한다.

이 승 래(Seung-Rae Lee)

정희원



1991년: 독일 아헨공대 수학과 학사 (VorDiplom)
 1993년: 독일 아헨공대 수학과 석사 (Diplom Mathematiker)
 1997년: 독일 아헨공대 수학과 이학박사 (Dr. rer. Nat.)
 1995년 7월~1996년 12월: 독일 뒤스부르크 대학

수학과 강사

1998년 9월~2000년 3월: 서울대학교 제어계측신기술연구센터 Post Doc. 연구원

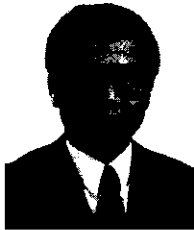
2000년 4월~2000년 11월: 상군관대학교 전기전자 및 컴퓨터공학부 연구조교수

2000년 12월~현재: 서울대학교 음향공학연구실 BK21 Post Doc. 연구원

<주관심 분야> 미분게임, 최적제어이론, 신호처리, 부호이론, 음향공학

성 평 모(Koeng-Mo Sung)

정회원



1971년: 서울대학교 공과대학
전자공학과 학사

1977년: 독일 아헨공대
전자공학과 석사
(Diplom Ingenieur)

1982년: 독일 아헨공대
전자공학과 공학박사
(Dr. Ing.)

1983년 7월~현재: 서울대학교 전기공학부 교수

1997년 7월~1998년 9월: 뉴미디어 통신공동연구소
장

1998년 9월~2000년 9월: 서울대학교 전기공학부
학부장

2000년 1월~현재: 한국음향학회 회장

<주관심 분야> ultrasonics, musical acoustics, under-
water acoustics, electro-acoustics, speech
recognition and synthesis