고속하다마드변환을 위한 치환기법

남지탁*, 박진배*, 최윤호**, 주영훈***
*연세대학교 전기공학과, **경기대학교 전자공학과, ***군산대학교 제어계측공학과

Permutation Algorithm For Fast Hadamard Transform

Ji-Tak Nam*, Jin-Bae Park*,Yun-Ho Choi**,Young-Hoon Joo***
*Yonsei University, **Kyunggi University, ***Kunsan University.

Abstract - The spectrum-recovery scheme in Hadamard transform spectroscopy is commonly implemented with a fast Hadamard transform (FHT). When the Hadamard or simplex matrix corresponding to the mask does not have the same ordering as the Hadamard matrix corresponding to the FHT, a modification is required. When the two Hadamard matrices are in the same equivalence class, this modification can be implemented as a permutation scheme. This paper investigates permutation schemes for this application. This paper is to relieve the confusion about the applicability of existing techniques ; reveals a new, more efficient method; and leads to an extension that allows a permutation scheme to be applied to any Hadamard or simplex matrix in the appropriate equivalence class.

1. 서 론

분광계측기(spectrometer)란 전자기적 방사광선을 서로 다른 주파수로 분리하여 각 주파수에서의 에너지를 측정하는 계측기이다. 하다마드분광계측기(Hadamard transform spectrometer, HTS)는 초창기에 사용된 다검파기 분광계측기 (multi-detector spectrometer) 와 단일파장 분광계측기(mono-chromator)의 신호 대 잡음비가 낮은 단점을 해결하고자 다중송신 (multiplexing) 기법을 이용한 분광계측기이다. 이 다중송신기법에서는 광원(source)으로부터 방사된 광선을 여러 주파수대역으로 분리하여 투과시키는 마스크(mask)가 사용된다. 처음에는 마스크 방식으로서 기계식 이동마스크(movable mask)가 사용되었으나, 재밍(jamming)과 조정불량(misalignment)의 기계적 문제점을 갖고 있다. 때문에 하다마드 변환 분광계측은 보편적인 관심을 끌지 못했다. 이런 단점을 보완하고자 고정형 전자광학 마스크(stationary electro-optical mask)가 소개되어 HTS는 푸리어 변환에 기초한 방법의 대체 방법으로 다시 관심의 대상이 되고 있다. HTS에서 여러다른 n개의 파장의 스펙트럼의 측정은 n개의 측정값을 필요로 한다. 이 측정값들은 n×1 행렬 η 로 표현되어 진다. 결과적으로 ϕ 가 실제 스펙트럼 성분을 나타내는 열벡터라면 측정은 다음의 식으로 표현된다.

$$\eta = H\,\phi \tag{1}$$

그리고 스펙트럼 회복은

$$\hat{\phi} = H^{-1}\,\eta = \frac{1}{n}\,H^{T}\,\eta \tag{2}$$

와 같이 표현된다.
검파기의 마스크가 하다마드행렬이 아니고 단순행렬(S-matrix)로 나타내어진다면 스펙트럼 분석은 다음과 같이 얻어진다.

$$\hat{\phi} = S^{-1}\,\eta \tag{3}$$

하다마드 행렬이 Sylvester-type 하다마드 행렬(H_s)일 때 스펙트럼 회복수식은 고속하다마드변환을 통해 효율적으로 계산된다. 마찬가지로 단순행렬도 Sylvester-type 하다마드 행렬로 부터 형성된 것이라면 부분적인 수정을 거쳐 고속하다마드변환을 스펙트럼 회복과정에 적용할 수 있다. 단순행렬은 하다마드 행렬의 첫 번째 행과 열을 생략한 다음 +1을 0으로 -1을 +1로 대치함으로써 형성된다. 하다마드 행렬이나 단순행렬이 Sylvester-type 하다마드행렬에 대응되지 않더라도 스펙트럼 회복과정에 2번의 치환과정을 추가해줌으로써 고속하다마드변환을 적용할 수 있다. 그러나 이것은 언제나 가능한 것은 아니고 하다마드행렬이 동일한 부류(equivalence class)에 속해 있어야 한다. 본 논문은 이러한 치환기법이 어떻게 이루어지는지를 소개하고자 한다.

2. 본 론

2.1 동일 부류의 하마다드 행렬

임의의 하다마드 행렬 H가 하다마드 행렬 H'의

치환에 의해 생성된 것이라면 H와 H'는 동일 부류(equivalence class)라고 말한다. 모든 하다마드 행렬은 표준화(normalization)된 하다마드 행렬(첫 번째 행과 열이 모두 +1로 된 하다마드 행렬)과 동일 부류이다. 또한 작은 차원의 하다마드 행렬(n=2,4,8,12)은 모두 동일부류이다. 따라서 어떤 하다마드 행렬도 Sylvester-type 하다마드 행렬로 치환될 수 있다. 반면 큰 차원의 행렬은 몇개의 부류로 나뉘는지 알려져있지 않고 단지 하나의 동일 부류에 속하지 않는 것만은 확실하다. Sylvester-type 하다마드 행렬이 아닌 경우 식(2)의 스펙트럼 회복과정에 고속하다마드변환(FHT)를 적용할 때 그 하다마드 행렬이 Sylvester-type 하다마드 행렬과 동일 부류라면 치환기법에 의해 Sylvester-type 하다마드 행렬로 변화시킬 수 있다.

$$H = P_a H_s P_b \tag{4}$$

와 같이 표현될 수 있다. 여기서 P_a 와 P_b 는 각각 행과 렬을 치환시키는 치환 행렬이다.

P_a에 의해 H_s의 행이 치환되고 P_b 에 의해 H_s의 렬이 치환된다. 치환행렬의 각각의 행과 렬은 +1 이나 -1의 0이 아닌 값을 하나 가진다. H_s는 symmetric 하기 때문에

$$H^T = P_b^T H_a P_a^T = P_2 H_s P_1 \tag{5}$$

식(5)을 식(2)에 대입하면

$$\hat{\psi} = \frac{1}{n} P_2 H_s P_1 \eta \tag{6}$$

스펙트럼회복 과정은 식(6)으로 표현된다.

단일 검파기 시스템의 경우 위와 유사한 회복과정이 적용된다. 2개의 표준화된 n+1차원의 하다마드 행렬을 가정하자. 하나는 Sylvester-type 하다마드 행렬(H_s)이고 나머지 하나(H_1이라 함)는 Sylvester-type 하다마드 행렬은 아니나 동일 부류에 있다. 다음과 같이 쓸 수 있다.

$$H_s = \begin{bmatrix} 1 & 1^T \\ 1 & G_s \end{bmatrix} \quad H_1 = \begin{bmatrix} 1 & 1^T \\ 1 & G_1 \end{bmatrix} \tag{7}$$

여기서 1 행렬은 모든 원소가 1인 n×1 벡터이다. 0행렬은 모든 원소가 0인 n×1벡터라 한다. 그러면 식(4)는 다음과 같다.

$$\begin{bmatrix} 1 & 1^T \\ 1 & G_s \end{bmatrix} = \begin{bmatrix} 1 & 0^T \\ 0 & p_a \end{bmatrix} \begin{bmatrix} 1 & 1^T \\ 1 & G_s \end{bmatrix} \begin{bmatrix} 1 & 0^T \\ 0 & p_b \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1^T \\ 1 & p_a G_s p_b \end{bmatrix} \tag{8}$$

따라서

$$G_1^T = p_a G_s p_b \tag{9}$$

또는

$$G_1^T = p_b^T G_s p_a^T = p_2 G_s p_1 \tag{10}$$

식 (10)은 G와 S의 행렬의 특성으로 다음과 같이 쓰여진다.

$$[J-2S_1]^T = p_2[J-2S_s]p_1 \text{ 또는 } S_1^T = p_2 S_s p_1$$

여기서 J행렬은 모든 원소가 1인 n정방행렬이다. 위의 결과로 부터 동일 부류의 H행렬 특성이 S행렬에도 적용되는 것을 알 수 있다.

$S^{-1} = \frac{-2}{n+1} G^T$ 의 관계를 식(3)에 대입하면

$$\hat{\psi} = \frac{-2}{n+1} G^T \eta = \frac{-2}{n+1} p_2 G_s p_1 \eta \tag{11}$$

단, G와 G_s는 동일 부류에 있다.

G_s는 단지 H_s의 첫 행과 열을 제거한 행렬이기 때문에 식(11)은 고속하다마드변환에 의해 계산될 수 있다.

$$\hat{\psi} = \frac{-2}{n+1} p_2 H_s p_1 \begin{bmatrix} 0 \\ \eta \end{bmatrix}$$

$$= \frac{-2}{n+1} \begin{bmatrix} 1 & 1^T \\ 1 & p_2 G_s p_1 \end{bmatrix} \begin{bmatrix} 0 \\ \eta \end{bmatrix}$$

$$= \frac{-2}{n+1} \begin{bmatrix} 1^T \eta \\ p_2 G_s p_1 \eta \end{bmatrix} \tag{12}$$

따라서 단순행렬인 경우의 스펙트럼 회복과정은 위의 식으로 표현된다. 확실히 이러한 과정을 효과적으로 이용하는데 있어서의 핵심은 적절한 치환행렬 p_1와 p_2를 구하는 기술에 있다.

2.2 Maximal-Length Shift-Register Sequence에 의한 단순행렬의 경우

단일 검파기 하다마드 부호하는 단순행렬에 의해 주로 이루어진다. 그런데 이 단순행렬은 MLSRS(Maximal-Length Shift-Register Sequences)에 의해 생성되는 부류의 단순행렬이 널리 사용되어 왔다. 그 이유는 이 부류의 단순행렬은 순환적인(cyclic)형태로 표현되기 때문이다. 순환적인 형태란, 행렬의 각 행이 바로 그 위에 있는 행을 한 칸 이동시킴으로써 (cyclic shift)이루어진다는 것을 의미한다. 이런 특성(window property)은 부호화과정을 매우 단순화시킨다. 왜냐하면 적절히 구성된 2n-1원소 마스크의 경우 i+1번째 측정에 대한 마스크는 i번째 측정에 대한 마스크를 왼쪽으로 한 칸 이동시킴으로써 대치할 수 있기 때문이다. n×n MLSRS 단순행렬은 $n=2^m-1$ (여기서 m은 정수)에 대해 존재한다. 이런 단순행렬은 Sylvester-type 하다마드 행렬(H_s)로 부터 구해지는 단순행렬과 같은 부류이고 따라서 위에서 제시한 바대로 치환기법을 이용하여 고속하다마드변환을 적용할 수 있다. 치환행렬을 구하는 방법을 살펴보자.

1. 첫 번째 치환행렬 구하기 : 앞에서 제시한 스펙트럼 회복과정에서 첫 번째 치환이란 η의 행을 치환시키는 P_1를 말한다. 이 행렬을 π_1으로 나타낸다. $\pi(i)=j$는 i번째 행을 j번째행으로 치환하는 것을 의미한다. 치환행렬을 구하기위해 MLSRS에

의해 형성된 단순행렬의 처음 m개의 열을 n×m 행렬 R로 정의한다. 그러면 R의 행들은 최상위 비트가 왼쪽에 있는 m비트 unsigned binary number 라 할 수 있다. $\pi_1(1)$은 첫 번째 행의 십진수값이고, $\pi_1(2)$는 두 번째 행의 십진수값을 나타내는 형태로 이루어 진다. 바꾸어 말해서 π_1은 행의 십진수값이 증가하는 순서로 R을 치환하는 행렬이다.

2. 두번째 치환행렬 구하기 : 두번째 치환행렬 P_2는 π_2로 나타내어진다. 이것은 S^T행렬의 처음 m개의 열과 X행렬의 렬을 비교함으로써 얻어진다. X행렬은 S_s의 m개의 열로 $\pi_1(1), \pi_1(2), \cdots, \pi_1(m)$의 순서로 이루어진다. $\pi_2(i)$는 X의 i번째 행과 대응하는 S^T행의 색인을 나타낸다. HTS시스템이 단지 하나의 마스크 형태(하나의 단순행렬로 나타나짐)를 사용한다면 이런 치환를 계산하는데 소요되는 시간은 중요시되지 않는다. 왜냐하면 치환행렬은 한번 구해지면 반복적으로 이용되기 때문이다. 그러나 전자광학 마스크의 등장으로 하나의 고정된 형태가 아니라 마스크의 형태를 변화시키는 것이 용이하게 되었다. 이런 상황에서는 스펙트로미터가 초기화될 때마다 적당한 치환행렬을 구하는 것이 바람직하다. 결론적으로 π_2를 결정하기위해 큰 차원의 Sylvester-type 하다마드 행렬을 구축하는 것은 메모리나 속도에 있어서 바람직하지 않다. 또한 이 방법은 n의 크기가 커짐에 따라 필요한 비교의 횟수가 극도로 커지게 된다. 한편 순환적(cyclic)형태가 아닌 단순행렬에 이 방법을 적용하는 것은 많은 문제가 있다. 예를 들어서 다음의 cyclic MLSRS-S행렬을 생각해보자.

$$S_7 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

$$S_7' = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

간단한 열치환 즉, 5열을 2열로, 7열을 3열로 치환하여 얻어진 또 다른 S 행렬을 생각해 보자. 이렇게 형성된 S_7'행렬을 고려해보자. 지금까지 언급해온 치환기법으로는 이 행렬에 대한 π_1행렬을 구하지 못한다. 각 행의 처음 3개의 원소는 3-비트 숫자로 해석되며 정수 1~7까지를 각 한번씩 나타내어야 한다. 그러나 위의 예는 {5,5,6,0,3,6,3}를 나타내어 이 조건을 만족시키지 못함을 알 수 있다. 이러한 문제점은 오직 cyclic MLSRS-S행렬을 사용하는 것에의해 해결될 수 있다. 이 경우에 MLSRS의 window property 에 의해 각각의 0이 아닌 m-비트 형태가 한 번 표현되어 S행렬의 처음 m렬에 단지 한번 나타내어 진다.

2.3 분해기법

π_2를 구하는데 있어 또 다른 방법을 소개한다. Sylvester-type 하다마드 행렬로 부터 얻어진 n차원의 S_s행렬은 다음과 같이 분해될 수 있다.

$$S_s = M^T M \quad \text{modulo 2} \tag{13}$$

여기서 M은 m×n 행렬이고 $n=2^m-1$

m-비트 2진수로 표현되는 H행렬의 열은 1에서 부터 2^m-1까지의 0이 아닌 정수를 형성한다. 그리고 이것은 순차적으로 그 정수값이 증가함을 보인다. 분해기법에 의해 π_2를 구하는 방법은 다음과 같다.

1. S행렬의 처음 m열로 이루어진 부분행렬에서 m-비트가 십진수 $2^0, 2^1, \cdots 2^{m-1}$에 대응하는 행을 찾는다. 2^{m-1}의 값과 대응되는 행의 S행이 m×n인 L행렬의 첫 번째 행이 된다. 마찬가지로 2^{m-1}의 값과 대응되는 행의 S행은 L행렬의 2번째 행이 된다. 이런 과정을 계속하여 2^0과 대응되는 행이 L의 m번째 행이 될 때까지 한다. 바꾸어 말해서, 2의 정수 급수가 나타나는 행을 π_1행렬에 사용한다.

2. L의 n렬을 m-bit 숫자로 해석한다. 최상위 비트가 맨 위에 온다고 생각한다. 그러면 1번째 열의 2진수와 대응하는 십진수 값을 π라 하면 $\pi_2(i) = 1$ 또 두번째와 대응되는 십진수 값을 j라 하면 $\pi_2(j)=2$ ... 등등. 이런 방법은 한 번 이상의 치환을 계산하는데 있어서 메모리와 계산시간을 절약할 수 있다는 이점을 가지고 있다. 게다가 이런 방법에서 요구되어지는 비교 과정이 단순한 찾기과정으로 대치됨으로써 보다 효율적인 실행이 가능하다.

2.3.1 분해기법의 확장

위의 형태의 분해방법은 여전히 위에서 제시한 S_7'에 대해서 적용이 되지 않는다. 그러나 이 방법은 S_s와 동일한 부류에 있는 모든 S행렬에 적용될 수 있도록 수정될 수 있다. S의 처음 m열의 행이 0이 아닌 모든 m-비트 유형을 형성하지 않는다면 m열의 또 다른 조합을 선택하면 된다. m개의 적당한 열을 선택한 다음 π_1을 구하는데 필요한 R행렬을 그것들로 형성한다. π_2를 구하기 위해서는 S의 m열을 이용하기보다 R행렬을 이용하여 2의 정수급수에 대응하는 2진수를 찾으면 된다. S의 적절한 m열의 부분 행렬을 찾는 것이 체계적으로 수행된

다면 확실히 해를 구할 수 있다. 물론, 커다란 행렬에 대해서는 여러 가지 다른 조합이 이항분포계수 $\binom{n}{m}$에 의해 주어지고 이는 n과 m이 증가함에 따라 몹시 커진다. 다행히 처음 m개의 열로 이루어지지 않을 때 단지 2 번째에서 m+1까지의 렬이나 3 번째에서 m+2까지의 렬을 고려해봄으로써 조건을 만족시키는 부분행렬을 구할 수 있다. 예를 들어 분해기법을 위에서 제시한 S_7'에 적용해 본다. 주어진 S_7'의 처음 m=3개의 렬은 조건을 만족하는 3-비트 형태를 나타내고 있지 않다. 따라서 2,3,4열을 이용해야 한다.

1. π_1구하기 : 2,3,4열로 이루어진 행렬 R을 선택한다.

$$R = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$\pi_1(1)=2, \pi_1(2)=3, \pi_1(3)=4, \pi_1(4)=1,$

$\pi_1(5)=7, \pi_1(6)=5, \pi_1(7)=6$

2. π_2구하기: 앞에서 구한 행렬 R과 m=3이라는 사실로부터 행렬 L을 구할 수 있다.

$2^{3-1} = 4$ $\therefore$ S_7'의 3행이 L의 1행이 된다.

$2^{3-2} = 2$ $\therefore$ S_7'의 1행이 L의 2행이 된다.

$2^{3-3} = 1$ $\therefore$ S_7'의 4행이 L의 3행이 된다.

$\therefore$ S_7'의 1, 3, 4행으로 L행렬을 구한다.

$$L = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

L행렬 각각의 열을 3-비트 2진수로 생각하여 치환행렬을 구할 수 있다.

$\pi_2(6)=1, \quad \pi_2(4)=2, \quad \pi_2(2)=3, \quad \pi_2(1)=4,$

$\pi_2(5)=5, \quad \pi_2(3)=6, \quad \pi_2(7)=7$

이와 같은 치환을 행렬로 변형하기위해 π_1을 행렬 I에 적용하여 P_1을 구하고, π_2을 I에 적용하여 행렬 P_2을 구한다. 그러면 구하고자 하는 $(S_7')^T = p_2 S_s p_1$ 가 얻어진다.

3. 결 론

이 논문에서 스펙트럼 회복과정을 FHT를 사용하여 보다 손쉽게 하기위해 마스크를 나타내는 H행렬과 S행렬을 FHT를 사용할 수 있는 행렬로 치환하는 기법에 대해 알아 보았다. 위의 조사에서 이끌어진 주요한 결론은 다음과 같다.

1. 큰 차원의 S-행렬에 대해 하나 이상의 동일한 부류가 존재하기 때문에 FHT의 사용을 위한 치환기법이 있는 것을 확신하기 위해 S-행렬이 S_s행렬과 동일한 부류에 속한다는 것이 증명되어야 한다.
2. Cyclic MLSRS-S행렬을 사용함으로써 구하려는 치환의 계산을 최대한 단순화시킬 수 있다.
3. 치환기법을 한 번 이상 계산해야 한다면 분해기법에 의해 필요한 메모리와 계산과정을 줄일 수 있다.
4. Noncyclic MLSRS-S행렬을 사용해야만 할 때에도 FHT의 사용이 가능하다. 그러나 기존의 치환기법을 그대로 적용할 수 없다. 본 논문에서 제시한 분해기법의 확장을 통해 이러한 문제점을 해결할 수 있다.

[참 고 문 헌]

[1] M. Harwit and N. J. A. Sloane, Hadamard Transform Optics, Academic Press, 1979.

[2] F. J. MacWilliams and N. J. A. Sloane, The Theory of Error Correcting Codes, 1977.

[3] G. H. Golub and C. F. Van Loan, "Matrix Computations", The John Hopkins University Press, Baltimore, pp. 3, 1983.

[4] E. D. Nelson and M. L. Fredman, J. Opt. Soc. America 60, 1664, 1970.