REAL SEQUENCE 의 CYCLIC CONVOLUTION 을 이용한 FFT 와 FHT 의 비교에 관한 연구

°성 상기, 김 진탁, 김 수일, 이 진이, 양 승인.
숭실대학교 전자공학과

A STUDY ON COMPARISION OF THE FFT AND FHT CYCLIC CONVOLUTION OF REAL SEQUENCE

Sang Ki Sung, Jin Targ Kim, Su Il Kim, Jin Lee Lee, Seung In Yang.
Dept. of Electronic Eng. Sung Sil University.

ABSTRACT

Recently, new fast transform (such as discrete *Hartley Tranasform) have been proposed which are* best suited for the computation of real sequence. Two approaches using Fourier or Hartley transform are first compared.

This paper is treated real sequence, compared number of addition of *cyclic convolution with us*ing the FFT and FHT, the convolution technique is defined as a separating system impulse response to the given input and output of the system.

1. 서 론

Radix-2 Fast Fourier Transform algorithm 은 1965년에 Cooley 와 Tukey 에 의해 처음으로 제안되었다. FFT 변환에서는 실수부분을 even, 허수부분을 odd 로 한 복소수 연산이다.

길이가 2^n 인 실수 또는 길이가 2^(n-1) 인 복소수 FFT 의 이용과 덧셈을 부가하고 길이가 2^n 인 FFT 하나를 이용한 시간에서 실수FFT 두개를 선택한다.

최근에 곱셈과 덧셈 둘 다 최소의 알고있는 연산수에 따른 2^n 인 FFT algorithm 은 실수함수 영역에서 제안되었다. 동시에 Bracewell 은 FHT algorithm 을 제시하였다.

본 논문에서는 fast transform algorithm 을 이용하여 길이가 2^n 인 실수함수의 cyclic convolution 을 계산하는데 그 목적을 둔다. 연산수는 세 번의 실수 곱셈과 세 번의 덧셈을 필요로 한다.

2. FFT 및 FHT 를 이용한 CONVOLUTION

1) FFT 를 이용한 CONVOLUTION

DFT는 convolution 성질을 갖고 있으며 이것은 그림.1 에 나타나 있다.[1]
초기 sequence x(n) 과 h(n) 이 실수 일때, X(h) 와 H(h) Y(h) 는 Hermittian symmetry 를 갖는다.
즉, $Y(h) = Y^*(-k)$ 이다. 그리고 DFT는 실수 함수에서 수행되고, IDFT는 hermittian symmetry 에 따라 실수 함수에서 수행된다.
이것은 다음과 같은 연산수로 나타내어진다.

$$M(n)=(2^{(n-1)})(2n-3)+3 \tag{1}$$

$$A(n)=(2^{(n-1)})(6n-7)+5 \tag{2}$$

n은 길이이다.
ex) $N=2^n$ $N=8,\ n=3$
이 연산수에서 X(k) 와 H(k) 는 둘 다 symmetry 이고 Y(k) 는 두 번의 실수곱셈(k=0, 과 k=N/2) 과 (N/2)-1 번의 복소 곱셈(k=1, ... ,N/2-1) 이다. 본 논문은 세 번의 곱셈과 세 번의 실수 덧셈 에 대하여 고찰했다.

2) FHT 를 이용한 CONVOLUTION

DHT는 Fourier Transform 의 실수부와 허수부의 합으로 정의된다.[4,5,7]

$$X(k)= (1/N)\sum_{n=0}^{N-1}x(n)(\cos(2\pi nk/N)+\sin(2\pi nk/N)) \quad k=0,\dots,N-1$$
$$= \sum_{n=0}^{N-1}x(n)\ \mathrm{cas}(2\pi nk/N) \tag{3}$$

여기서 $\mathrm{cas}(x)=\cos(x) + \sin(x)$
Hartly domain 에 따르는 연산에 대응하여 Frequency domain 에서,

$$Y(k)=X(k)He(k)+X(-k)Ho(k) \tag{4}$$

여기서 He(k) 는 h(n) 의 Hartley Transform 의 even 이고, Ho(k) 는 h(n) 의 Hartley Transform 의 odd 이다.

$$\begin{bmatrix} Y(k) \\ Y(-k) \end{bmatrix} = \begin{bmatrix} He(k) & Ho(k) \\ -Ho(k) & He(k) \end{bmatrix} \begin{bmatrix} X(k) \\ X(-k) \end{bmatrix} \tag{5}$$

두 번의 실수 곱셈을 할때, 디지탈 filtering 에서 odd=0 이므로

k=0 → 식(4)

$$Y(0) =X(0)He(0) + X(0)Ho(0) \tag{6.1}$$

여기서 odd part는 0이므로

$$Y(0)=X(0)He(0) \quad (6.2)$$

$k=N/2 \rightarrow$ 식(4)

$$Y(N/2)=X(N/2)He(N/2) \quad (7)$$

Hartley domain에서 곱셈의 수는 Fourier domain에서 와 마찬가지로 똑같은 연산수를 필요로 한다.[4, 7] radix - 2 FHT algorithm은 같은 곱셈의 수와 실수합수에서 이에 대응하는 FFT algorithm보다 (N-2)번의 곱셈을 더 행해야한다. [5]

$$M(n)=(2^{(n-1)})(2n-3)+3 \quad (8)$$

$$A(n)=(2^{(n-1)})(6n-17/3)+5-(4/3)(-1)^{(n-1)} \quad (9)$$

FHT를 이용한 convolution은 간단한 구조를 갖는데, 이는 Hartley transform의 self-inverse에 기인한다.

3. F F T를 이용한 규칙적인 CONVOLUTION 도표

Hartley domain에서 convolution 성질은 식(4)에서 주어지고 X(K)와 H(K)의 익은

$$Y(K) = Xe(K)H(K) + Xo(K)H(-K) \quad (10)$$

여기서, Xe(K)는 x(n)의 Hartley transform의 우수부이고, x(n)의 DFT의 실수부이다.
Xo(K)는 x(n)의 DFT 허수부이다.

식(5)를 다시 쓰면,

$$\begin{bmatrix} Y(K) \\ Y(-K) \end{bmatrix} = \begin{bmatrix} H(K) & H(-K) \\ H(-K) & -H(K) \end{bmatrix} \begin{bmatrix} RE\ X(K) \\ IM\ X(K) \end{bmatrix} \quad (11)$$

이것은 또한 복소수 적과 같다.
DHT는 DFT의 실수부와 허수부의 합이고 그림.2에서 각 DFT성분의 실수부와 허수부의 합으로 바꿔어진다. 이 도표는 FFT를 기본으로 하는 일반적인 convolution도표로서 OPERATION이 정확하게 같은 수를 필요로 한다.

$$A(n)=2^{(n-1)}*(6n-5)+3 \quad (12)$$

4. 개선된 FHT algorithm

개선된 FHT는 Fourier Transform과 Hartley Transform 사이의 밀접한 관계를 이용하여 개선할 수 있다.[3,5]

Fourier domain에서 FFT는

$$X(k)=\sum_{n=0}^{N-1} x(n)[\cos(2\pi nk/N)-j\sin(2\pi nk/N)] \quad k=0,\ldots,N-1$$

$$=\sum_{n=0}^{N-1} x(n)\, W_N^{nk} \quad (13)$$

여기서, $W_N^{nk}= \exp(-j2\pi nk/N)$. x(n)의 DFT에 (1+j)를 곱하면 다음 식이 얻어진다.

$$(1+j)X(k)=[Re\ X(k)-Im\ X(k)]+j[Re\ X(k)-Im\ X(k)]$$

$$=X(k)+jX(-k) \quad (14)$$

DFT의 기본적인 Split-radix decimation-in-time FFT 분해를 식(9)에 적용하면[3,6]

$$X(k)=\sum_{\substack{n=0 \\ n\ \text{even}}}^{N/2-1} x(2n)W_N^{2nk} + \sum_{\substack{n=0 \\ n\ \text{odd}}}^{N/4-1} x(4n+1)W_N^{4nk} + \sum_{\substack{n=0 \\ n\ \text{odd}}}^{N/4-1} x(4n+3)W_N^{4nk} \quad (15)$$

식 (11)에서 (1+j)를 곱하면,

$$(1+j)X(k) = X(k)+jX(-k)$$

$$= (1+j)\sum_{n=0}^{N/2-1} x(2n)W_N^{2nk} + (1+j)W_N^{nk}\sum_{n=0}^{N/4-1} x(4n+1)W_N^{4nk}$$

$$+(1+j)\sum_{n=0}^{N/4-1} x(4n+3)W^{(4n+3)nk} \quad (16)$$

식(12)는 지금 개선된 FHT algorithm의 기본적인 순환이다. 이것은 길이가 N에서 길이가 N/2인 DHT를 변환 한다. 길이가 N/4인 두번의 DFT를 더하고, (1+j) W 또는 (1+j) W의 twiddle factor(j, w)에 의해 약간의 곱셈 을 나타낸다.
실수 합수 FFT의 경우처럼, twiddle factor에 따른 butter-fly(Decimation-in-time)은 twiddle addition 계산의 반복 real data에서 FFT에대해서 얻는 것보다 더 정확하게 같은 수가 주어진다.

이것은 다음 식들로 유도된다.

$$A(n) =(2^{(n-1)})(3*2^n-5)+6 \quad (17)$$

이 algorithm의 전체 convolution은

$$A(n) =(2^{(n-1)})(6n-7)+9 \quad (18)$$

곱셈의 수는 FFT와 FHT가 똑 같다. 이 새로운 Fast Hartley Transform(FHT)algorithm의 convolution은 그림.2에 나타나 있다. 이 algorithm에서는 program 의 길이가 증가하는 것을 감소 시키는데 그 목적이 있다. 새로운 FHT algorithm은 이에 해당하는 algorithm으로서 같은 구조를 갖는다.

5. 결론

본 논문에서 FFT나 FHT를 이용하여 실수 합 수 cyclic convolution에 두개의 도표를 제시 하였다.
FFT와 FHT를 이용하여 연산수를 제시하여 FFT convolution은 산술적으로 덜 복잡하지만 구조적으로 더 복잡하다. 두개의 도표를 이용하여 개선점을 찾을 수 있었다.

1)FFT 도표: 실수 합수 두 번의 forward transform을 이용.

2)FHT algorithm ; 덧셈수를 줄임.

모든 연산수는 표.1에 요약 하였다.

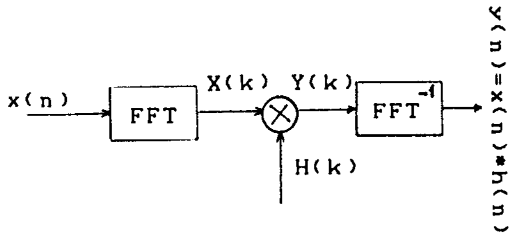

그림1. FFT를 이용한 일반적인 CONVOLUTION
Fig1. GENERAL CONVOLUTION USING FFT

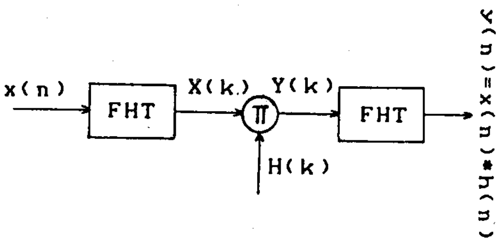

그림2. FHT를 기초로 한 CONVOLUTION
FIG2. CONVOLUTION BASED ON FHT

길이 n	convolution size (N)	곱셈수	덧셈수 초기 FFT	덧셈수 FORWARD FFT	덧셈수 초기 FHT	덧셈수 개선된 FHT
2	4	5	15	17	13	10
3	8	15	49	55	53	53
4	16	43	141	155	153	145
5	32	115	373	403	393	377
6	64	291	933	995	977	937
7	128	707	2245	2371	2329	2249
8	256	1667	5253	5507	5425	5257
9	512	3843	12037	12547	12377	12041
10	1024	8707	27141	28163	27825	27145
11	2048	19459	60421	62467	61785	60425

표1. REAL DATA에서 CYCLIC CONVOLUTION에 대한 연산 동작수
TABLE 1. NUMBER OF ARITHMETIC OPERATION FOR CYCLIC CONVOLUTION ON REAL DATA.

6. 참 고 문 헌

[1]J.W.Cooley, J.W.Tukey :"An algorithm for maç-hine computation of complex Fourier series". Math.comp.vol.19,pp.297-301.1965.

[2]P.Duhamel ,H.Hollman :"Implementation of "sp-lit radix" FFT algorithms for complex,real, and real symmetric data" : ICASSP'85,Tampa

[3]R.N.Bracewell :"The Fast Hartley Transform". proc.IEEE vol.22,no.8,pp.1010-1018.Aug.1984.

[4]H.V.Sorenson, C.S.Burrus, M.T.Heideman."On the computing the discrete Hartley Transform". IEEE Trans.Assp,vol.33,no.4,pp.1231-1238,Oct. 1985.

[5]김 진탁외 3인 : "FFT와 FHT의 algorithm및 수행시간 비교에 관한 연구" 통신학회 춘계학술발표회 논문집,: 제 6권 1호 ,5.1987

[6]Hong Hao and R.N.Bracewell:"A three-dimensi-onal DFT algorithm using the Fast Hartley Tr-ansform" proc.IEEE.vol.75,no.2,pp.264-266,Feb. 1987

[7].Nussbaumer; "Fast Fourier Transform and con-volution algorithm":Springer-verlag Berlin Hei-delberg,New York

부록. DISCRETE FOURIER TRANSFORM의 성질 및 DISCRETE HARTLEY TRANSFORM의 성질 (CONVOLUTION 중심으로)

1. DFT의 CONVOLUTION

(1) 일반적인 CONVOLUTION

$$y(k)=\Sigma x(n)h(k-n)=x(k)*h(k)$$

(2) TIME CONVOLUTION

$$y(k)*h(k) \rightarrow Y(n)H(n)$$

(3) FREQUENCY CONVOLUTION

$$y(k)h(k) \rightarrow (1/N)\,Y(n)*H(n)$$

2. DHT의 CONVOLUTION

(1) 일차원에서의 CONVOLUTION

$$F(n)=F1(n)*F2(n)= H1(k)H2e(k)+H1(-k)H2o(k)$$

F2(n)이 symmetric이면, H2(k)의 odd 부분은 0 이다.

그래서 이 식은 $H1(k)H2e(k)= H1(k)H2(K)$

여기서 , $H2(k)=H2e(k)+H2o(k)=H2e(k)$