한국통신학회 1987년도 춘계학술발표회 논문집(1987. 5. 23) 87439

FFT 와 FHT 의 ALGORITHM 및 수행 시간 비교에 관한 연구

°김진탁 성상기 김중태 이진이 양승인
숭실대학교대학원 전자공학과

A STUDY COMPARE WITH THE ALGORITHM AND THE RUN TIME OF THE FFT AND FHT·

JIN TARG KIM. SANG GI SUNG. JUNG TAE KIM. JIN REE LEE. SEUNG IN YANG.
Dept. of Electronic Eng. , Sung Sil University.

ABSTRACT

A fast algorithm has been worked out for performing the DFT(discrete Fourier transform) and the DHT(discrete Hartley transform) of a data sequence of N elements.

This paper deals with the algorithm of the FFT and the FHT, and compares the arithmetic numbers with the run time of the FFT and the FHT in the signal process.

In conclusion, the FHT is as fast as. or faster than the FFT.

1. 서 론

어떤 DIGITAL 신호 를 처리 하는 경우 그 신호의 특성을 분석하기 위하여 FOURIER 변환을 이용하게 된다.

종래의 경우는 DFT 방식으로 DIGITAL 신호를 처리 할경우 복소수 연산을 이용하면 N 번 곱셈과 N(N-1) 번의덧셈이 요구되었다.

계산을 하는데 있어서 연산 과정을 고속 처리하기 위하여 새로운 ALGORITHM 이 제안 되었는데 이러한 처리 방식이 FFT 와 FHT 방식인 고속 변환 ALGORITHM방식이다.

DFT 의 복수 연산에서 대칭성 과 주기성을 가지므로이러한 성질을 이용하여 N점 DFT를 N점 보다 작은부분으로 분할하여 계산 되어진다.

고속 처리 ALGORITHM은 Nlog2N의 시간에 비례하는 N개의 DATA를 FFT 와 FHT 방식으로 처리 할 수 있다.[1]

FFT 방식은 실수 부분을 EVEN, 허수 부분을 ODD로 이용한 복수 연산 방식이며, FHT 방식은 실수 부분을EVEN, ODD 로 분할 하여 계산 되어지는 방식이다. [2]

여기서 제안된 고속 처리 ALGORITHM 인 FFT 와 FHT 방식은 SPECTRUM 분석, DIGITAL SIGNAL PROCESSING, CONVOLUTION 등에 이용 할 수 있다.

이 논문에서는 고속 FOURIER 변환 방식인 FFT와 FHT의 ALGORITHM을 제시하였으며 FORTRAN PROGRAM 으로MICRO-VAX SYSTEM 을 사용하여 FFT와 FHT 방식에서각 SAMPLE 수에 대한 DIGITAL 신호 처리에 걸린 시간을 비교 검토 하였다.

2. 고속 변환 algorithm

2-1.FFT ALGORITHM

유한 수열 x(n) 의 DFT X(K) 는

$$X(K)=\sum_{n=0}^{N-1} x(n)\, W_N^{Kn} \qquad K=0,1,2,\ldots,N-1 \text{ --- } (1)$$

$$W = e^{-J\frac{2\pi}{N}}$$

X(K) 의 역변환(IDFT) 는

$$x(n)=1/N \sum_{K=0}^{N-1} X(K)\, W_N^{-kn} \qquad n=0,1,2,\ldots,N-1 \text{ --- } (2)$$

가 된다.[3]

식(1) 의 DFT에서 N점을 보다 작게 분할하여 계산과정을 줄일 수 있는 ALGORITHM이 FFT 이다.DFT의 연산과정을 줄일 수 있는 기본성질은 w 의 주기성과 대칭성을 이용한다.

$$X(K)= \sum_{n\ \mathrm{even}} x(n)\, W_N^{nk} + \sum_{n\ \mathrm{odd}} x(n)\, W_N^{nk} \text{ ----- } (3)$$

이때 even=2r, odd=2r+1 로 가정하면

$$X(K)=\sum_{r=0}^{(N/2)-1}x(2r)\,W_N^{2rk}+\sum_{r=0}^{(N/2)-1}x(2r+1)\,W_N^{2rk}$$
$$=\sum_{r=0}^{(N/2)-1}x(2r)\,W_N^{2rk}+\sum_{r=0}^{(N/2)-1}x(2r+1)\,W_N^{2rk} \text{----} \quad (4)$$

N점 DFT X(K)를 구할때 x(n)을 n이 짝수 와 홀수인 것의 두 부수열(subsequence)로 나누어 N/2 점 DFT를 산출한다. 여기서 N점 DFT를 N/2,N/4,N/8,...점 DFT로 변환 시켜 마지막에 2점 DFT가 될때까지 N=2 이므로 q=log2 N 번 의 분할 단계가 필요하다.

FFT ALGORITHM을 이용하여 N번 분할할 경우 각 분할 단계마다 N번의 덧셈과 곱셈을 연산되는 수는 N log2 N 이 된다.

2-2. FHT 알고리즘.

DHT는 k=0,1,--- N-1 인 실함수인 경우 COSINE 과 SINE 함으로써 DHT H(k)는 다음과 같이 정의된다.

$$H(k)=\sum_{n=0}^{N-1}x(n)CAS[2\pi/N*kn] \quad k=0,1,2,\text{---}N-1 \text{ --}(5)$$

$$x(n)=1/n\sum_{K=0}^{N-1}H(k)CAS[2\pi/N*kn] \quad k=0,1,\text{---}N-1 \text{ --}(6)$$

여기서 CAS(x)=COS(x)+SIN(x) 이다.
DFT와의 차이점은 SINE항에 "-J"가 곱하여진다.
즉 DFT를 DHT로 표현하면 다음과 같다.[4]

$$DHT[X(n)]=\underset{\text{Even}}{Re\{DFTX(n)\}}-\underset{\text{Odd}}{Im\{DFTX(N)\}} \text{ ---- } (7)$$

$$Re\{DFTX(n)\}=1/2[DHT\{X(N-n)\}+DHT\{X(n)\}]\text{---}(8)$$

$$Im\{DFTX(n)\}=1/2[DHT\{X(N-n)\}-DHT\{X(n)\}]\text{---}(9)$$

여기서 X(n)=X(0)이며 DHT를 DFT로 쉽게 계산할 수 있다. DHT를 FHT로 변형하기 위하여 변형성 정리(Shift theorm)와 유사성 정리(Similarity theorm)가 적용된다.

변형성 정리(Shift theorem)에 의하여

$$f(n+a)=DHT\ H(k)COS(2\pi ak/N)-H(N-k)SIN(2\pi ak/N) \text{ ---- } (10)$$

로 된다.

유사성 정리(Similarity theorem)는 X(n)의 Data 사이에 zero 를 삽입하면 두배의 길이가 되어 처음 DHT 인자는 반복되어 다음과 같이 구할 수 있다.

[1,2,3,4,]=DHT[2.5,-1,-0.5,0]

[1,0,2,0,3,0,4,0]=DHT[2.5,-1,-0.5,0,2.5,-1,-0.5,0]

로 된다.

DHT H(k)={a1 a2 b1 b2 c1 c2-------}의 N개 인자가 있다면 그때 N/2인자의 a1 수열 {a1 b1 c1----}은 DHT{α_1 β_1 γ_1---}이 되고, a2 수열 {a2 b2 c2----}은 DHT{α_2 β_2 γ_2 ---}이 된다. 이러한 연산은 식(11)에서 설명된다.

$$\begin{aligned}
H(k)=&DHT\ of\{a1\ 0\ b1\ 0\ c1\ 0\ ....\}\\
&+DHT\ of\{0\ a2\ 0\ b2\ 0\ c2\ 0\ ....\}\\
=&\{\alpha_1\ \beta_1\ \gamma_1\cdots\alpha_1\ \beta_1\ \gamma_1\cdots\}\\
&+DHT\ of\ \{0\ 1\}*\{a2\ b2\ c2\ ....\}\\
=&\{\alpha_1\ \beta_1\ \gamma_1\cdots\alpha_1\ \beta_1\ \gamma_1\cdots\}\\
&+\{\alpha_2\quad \beta_2\cos(2\pi/N)\quad \gamma_2\cos(2\pi*2/N)...\\
&\quad -\alpha_2\ \beta_2\cos(2\pi(N/2+1)/N)\\
&\quad \gamma_2\cos(2\pi(N/2+2)/N)...\}\\
&+\{0...\ \gamma_2\sin(2\pi*2/N)\ \beta_2\sin(2\pi*3/N)\\
&\quad 0\ ...\ \gamma_2\sin(2\pi(N/2+2)/N)\\
&\quad \beta_2\sin(2\pi(N/2+3)/N)\}\qquad ....\quad (11)
\end{aligned}$$

식 (11)의 일반적인 공식은 식 (12)로 표현 할 수 있다.

$$H(k)=Ha1(k)+Ha2(k)COS(2\pi k/N)+Ha2(N-k)*SIN(2\pi k/N) \text{ ----- } (12)$$

여기서 Ha1(k)와 Ha2(k)는 1/2{α_1 β_1 γ_1…α_1 β_1 γ_1…}과 1/2{α_2 β_2 γ_2…α_2 β_2 γ_2…} 로 된다.

예 를 들면 N=8 인 경우 FIG.2 와 같이 H(k) ={1 2 3 4 5 6 7 8} 인 수열로 배열 할 수 있다. 이 때 변형은 P-1 step 으로 수행된다.

첫 번째 단계, 2 column 으로 {1,3,5,7}과 {2,4,6,8} 즉 odd 와 even 을 분리.

두 번째 단계, 고른의 마지막 단계로써 첫단계에서 odd 와 even 즉 {1,5},{3,7},{2,6},{4,8}로 구분.

{a,b} → 1/2{a+b,a-b}의 공식에 의하여
H1(k)={6,-4,10,-4,8,-4,12,-4} 으로 된다.
H2(k)는 한 칸 간격으로 덧셈과 뺄셈으로 수행.
H3(k)는 두 칸 간격으로 덧셈과 뺄셈으로 수행.
H3(k)에서는 8개가 고대로 Sine 과 Cosine 을 포함. 여기서 N=2 공식에서 FHT 고속 변환 ALGORITHM 에서 1/N 이므로 1/8을 나누어 얻은 값이 H(k)값이 된다.

즉 {1,2,3,4,5,6,7,8} 은
DHT{4.5,-1.7,-1,-0.7,-0.5,-0.3,0,0.7}
와 같은 결과를 얻을 수 있다.

위 와같은 계산에 의하여 FHT의 실제적인 연산 동작 과정은 Nlog2N-3N+4 의 곱셈과 (3Nlog2N-3N+4)/2 의 덧셈이 요구된다.

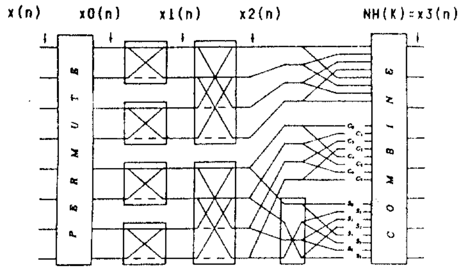

그림.1 DISCRETE HARTLEY TRANSFORM에 대한 FLOW DIAGRAM(N=8,q=3)

그림 2.NUMERICAL EXAMPLE OF A SHORT FHT WITH N=8,q=3

n	x(n)	π	x0(n)	x1(n)	x2(n)	x3(n)	H(k)	k
0	1	1	1	6	16	36	4.5	0
1	2	3	5	-4	-8	-13.6	-1.7	1
2	3	5	3	10	-4	-8	-1	2
3	4	7	7	-4	0	-5.6	-0.7	3
4	5	2	2	8	20	-4	-0.5	4
5	6	4	6	-4	-8	-2.4	-0.3	5
6	7	6	4	12	-4	0	0	6
7	8	8	8	-4	0	5.6	0.7	7

3. 결과 및 검토

표1은 DFT(N),FFT(Nlog2N),FHT(Nlog2N-3N+4)의 연산되는 수를 계산한 결과이다. 그리고 이와같이 연산수에 의하여 FFT와 FHT 처리 속도를 비교한 경우에 신호처리 시간을 표2에 재시 하였다.

표2 에 재시된것과 같이 FHT가 FFT보다 빠른 것을 얻었다. 표 2 에 나와있는 시간은 RUN TIME이다.

이 계산에 사용된 계산기는 숭실대학교 전자과에 설치되어있는MICRO-VAX이다.

표1.N에 따른 N(DFT) 과 N log2 N (FFT) 와 FHT 의 비교

N	N(DFT)	N log2 N(FFT)	N log2 N-3N+4(FHT)
8	64	24	4
16	256	64	20
32	1024	160	68
64	4096	384	196
128	16384	896	516
256	65536	2048	1284
512	262144	4608	3076
1024	1048576	10240	7172

표2.FFT/FHT FORTRAN PROGRAM에서의 수행시간

N	FFT	FHT
8	0.38초	0.28초
16	0.4초	0.32초
32	0.81초	0.53초
64	1.28초	0.81초
128	1.71초	1.08초
256	2.18초	1.32초
512	2.81초	1.64초
1024	3.64초	2.08초

4. 결론.

디지탈 신호를 컴퓨터에 의하여 해석할때 원래 신호를 even 과 odd로세분하여처리한다면 더욱 빠른계산을 할 수 있을것이다.
이러한 이유로 새로운 고속 Fourier 변환 알고리즘이 재안되어 DFT방식에서 복소수 연산을 even과 odd로구분한 FFT방식보다 실수 부분을even 혹은 odd로 구분한 FHT 방식이 빠른 결과를 얻었다.
앞으로 FHT 방식에서도 MULTI-RADIX방식을 사용한다면 계산시간을 더욱 감축 할 수 있을것이다 .

참 고 문 헌

1.H.RENGANATHAN AND K.M.M.PRABHU
"IMPROVE FFT EFFICIENCY IN HIGH SPEED APPLI - CATION"
NORTH-HOLLAND,DIGITAL SIGNAL PROCESSING
PP.101-104

2.R.N.BRACEWELL
"THE FAST HARTLEY TRANSFORM"
IEEE PROC.VOL.72,PP.1010-1018,
AUG.1984

3.A.V.OPPENHEIM AND R.W.SCHAFER
"DIGITAL SIGNAL PROCESSING"
CHAP.6
ENGLEWOOD CLIFFS.NJ,PRENTICE-HALL,1975

4.HENRIK V.SORENSEN
"ON COMPUTING THE DHT"
IEEE TRANS.ASSP,VOL.33,PP.1231-1238,
OCT.1985