

DFT 및 FFT 에 있어서의 Redundancies 와 그의 除去에 의한 Fourier 變換高速化 (Improvement in computing times by the elimination of redundancies in existing DFT and FFT)

安 秀 桔*

(Ann, Sou guil)

要 約

DFT 및 FFT 의 計算過程에 포함된 redundancy 를 分析하고 이 redundancy 를 最大限으로 감소시키는 方法을 提案하였다. 이로서 FFT 計算 爲을 위한 計算機時間이 大幅減縮됨을 實證하였다.

T.D.C.F 와 T.D.F.T 등을 Discrete 한 경우에 확장하여 Digital Signal Processing 의 여러 過程에 있어서 더 깊은 物理的 뜻을 부여할 수가 있었다.

Abstract

Redundancies in the Calculation of DFT and FFT are analyzed and new algorithms are proposed which are capable of reducing the machine time by a considerable amount.

New extensions of T.D.C.F. and T.D.F. T. are given for the discrete case which permit a deeper insights for the techniques of digital signal processing i.e. Discrete Fourier Transform, Convolution Sum and Correlation sequences.

1. 序 論

그 概念이 導入된 當時에는 該當된 信號의 스펙트럼 計算 程度도 받아들여졌던 푸리에 級數는 解를 거듭함에 따라 回路 또는 한 시스템에 있어서 解析方法으로 不可缺한 道具가 되었고 다시 푸리에積分을 통하여 週期性이 없는 一般波形에까지 活用되게 되었다.

주파수영역에 線스펙트럼을 보이는 것이 時間領域에 있어서 週期性을 갖었음을 나타내는 것과 같이 時間領域에 있어서 discrete 한 波形에 對應하는 스펙트럼은 周波數域에 있어서 無限히 週期的으로 되풀이되고 비록 높은 곳에서나마 이를 限定하면 時間域에 있어서의 波

形은 理想的인 Delta 函數로부터 離脫한다. DFT (Discrete Fourier Transform)에 있어서는 Sampling 된 곳에서만 어떠한 값을 갖고 其他 時點에서는 零이라고 생각하는 아날로그 信號에 되도록이면 近接시키고자 하는 것을 止揚하고 單位時間의 정수배인 곳에서만 定義되는 數列로서 생각하여 시간(몇번제인가를 나타내는 순저장의 數이지 이미 시간으로 限定되는 것은 아니지만)과 주파수 양쪽에서 다 週期性을 띄고 모두 다 Discrete 한 分布를 갖는 한쌍의 函數를 定義하여 DFT pair 로 생각하게 될 정도로 추상화되었다. S.J. Mason 은 對象函數가 週期性을 갖거나 해서 그 Correlation 積分이 發散하게 되는 경우에 그 時間平均을 取扱하여 그 積分이 有限할 경우와 구별해서 정리하였는데⁽¹⁾ 이러한 power signal 의 경우나 積分이 有限한 Energy Signal 의 경우에 積分時間을 限定하고 그 時間內의 平均値를 取扱하여 中心時間 t , 平均區間 T , 信號間의 時間差 τ 등의 函數로 取扱하여 Correlation

* 正會員, 서울大學校 工大 電子工學科
(College of Engineering, Seoul National University)

接受日字: 1977年 12月 24日

의 기록을 볼 수 있게 하는 方法이 提案되었는데⁽²⁾ 그 T.D.C.F.(Time Dependant Correlation Fuction)은

$$R(\tau, t, T) = \frac{1}{T} \begin{cases} t + \frac{T}{2} \\ u(t)v^*(t+\tau)dt \dots\dots\dots(1) \\ t - \frac{T}{2} \end{cases}$$

와 같이 定義되고 $u(t)$ 를 電壓 또는 電流波形으로 잡고 $v(t)$ 를 phasor $e^{j\omega_0 t}$, $\tau=0$ 로 잡음으로 해서 $u(t)$ 가 $e^{j\omega_0 t}$ 와 사이에 授受할 수 있는 電力 또는 energy로서 $u(t)$ 內에 들어있는 W_0 成分을 求하는 것이 Fourier Series 또는 Fourier Transform 計算過程의 物理的 뜻이라는 것이 提案⁽³⁾되었다.

2) Time Series의 energy 또는 電力, Convolution 및 Correlation

N 個의 Element로 되어있는 Discrete 한 time series $\bar{x}(n)$ 의 DFT는

$$X(k) = \sum_{n=0}^{N-1} \bar{x}(n) e^{-j\left(\frac{2\pi}{N}\right)nk} \quad (2)$$

$k=0, 1, \dots, N-1$

로서 定義되고 그 逆變換 역시 N 個의 時點에서

$$\bar{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\left(\frac{2\pi}{N}\right)nk} \quad (3)$$

$n=0, 1, \dots, N-1$

로서 定義된다⁽⁴⁾

$f(t)$ 의 Fourier Transform에서 $f(t)$ 를 電壓: 힘 등 原因函數로 보고 phasor $e^{-j\omega_0 t}$ 를 電流(變位) 등 結果函數의 Conjugate로 보며 이 둘사이의 Correlation을 電力(또는 電力量)으로 생각하여 $f(t)$ 에 포함되어 있는 ω_0 成分을 求하는 T.D.F.T.를 Discrete 경우에 확장하기 위해서 $e^{j\left(\frac{2\pi}{N}\right)nk}$ (但, n 은 임의의 定數)를 discrete motion phasor로 이름 붙여 그 單位回轉角도를

$$W_N^{-k} = e^{j\left(\frac{2\pi}{N}\right)nk} \quad (4)$$

라고 할 때 (2) 式은

$$X(k) = \sum_{n=1}^{N-1} \bar{x}(n) W_N^{-nk} \quad (5)$$

가 된다.

두 Time Series $x_1(i)$ 및 $x_2(j)$ 가 各各 N 및 M 個의 Element로서 이루어졌을 때 Convolution은 한 Series의 순서를 逆轉하여 相對的變位를 시켜가며 얻어진 積을 計算한 것이기 때문에 서로 접촉하기 시작한 點으로부터 各各 反對端에 이르러 서로 離脫할 때까지 $M+N-1$ 의 點에서의 non-zero 값을 생각할 수 있다. 길이가 各各 N 인 두 Time Series $x_1(n)$ 와 $x_2(n)$ 사이의 Convolution $x_3(n)$ 은

$$\bar{X}_1(k) = \sum_{n=0}^{2N-2} \bar{x}_1(n) W_{2N-1}^{-nk} \quad (6)$$

$$\bar{X}_2(k) = \sum_{n=0}^{2N-2} \bar{x}_2(n) W_{2N-1}^{-nk} \quad (7)$$

일대

$$x_3(n) = \frac{1}{2n-1} \left(\sum_{k=0}^{2N-2} [\bar{X}_1(k) \bar{X}_2(k)] W_{2N-1}^{-nk} \right)$$

로 計算되지만

$x_1(n)$ 와 $x_2(n)$ 가 N 를 주기로 한 週期函數로서 convolution(circular) 역시 N 週期函數가 될 때에는 그 一週期단의 函數 $x_3(n)$ 은 다음 式으로 주어진다.

$$x_3(n) = \left(\sum_{m=0}^{N-1} x_1(m)x_2(n-m) \right) R_N(n) \quad (9)$$

但 $R_N(n)$ 은 $n=0$ 에서 $n=N-1$ 까지의 計算值만 남기고 그밖을 소멸시키는 케이트 函數이다.

各各의 길이가 N 및 M 인 두개의 Time Series $x(n)$ 및 $y(n)$ 間의 Correlation 函數 $C_{xy}(k)$ 는 $x(n)$ 와 $y(n)$ 가 똑같이 $n=0$ 에서 始作(non zero 值)하는 實函數이고 $N \geq M$ (이것 때문에 一般性을 喪失하지는 않음)이라면

$$C_{xy}(k) = \frac{1}{M-|k|} \sum_{n=0}^{M-|k|} x(n)y(n+k) \quad (10)$$

이지만 一般的으로

$$C_{xy}(k) \neq C_{yx}(k) \quad (11)$$

이므로 다음式으로 정의되는⁽⁵⁾

$$R_{xx}(k) = \frac{1}{N-|k|} \sum_{n=0}^{N-|k|-1} x(n)x(n+k) \quad (12)$$

을 活用하는 事例가 있을 따름이다.

電力의 形成여하여 따라 Correlation의 有無를 생각하는 T.D.F.T.를 Discrete Series에 확장할때에는 $k=0$ 인 경우만 생각하면 되기 때문에

$$R_{xx}(0) = \frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2 \quad (13)$$

이며 다시 一般的인 경우에 확장하여 週期가 同一하게 N 인 Time Series $\bar{x}(n)$ 와 $\bar{y}(n)$ 間의 zero delay correlation function $C_{xy}(0)$ 는

$$C_{xy}(0) = \frac{1}{N} \sum_{n=0}^{N-1} \bar{x}(n) \bar{y}^*(n) \quad (14)$$

로 주어진다.

T.D.F.T.의 경우도 마찬가지로 주기가 N 인 series. $x(n)$ 에 대하여

$$F(k) = \frac{1}{N} \sum_{n=0}^{N-1} \bar{x}(n) e^{-j\left(\frac{2\pi}{N}\right)nk} \quad (15)$$

로 된다. 이는 Series $x(n)$ 와 正方向(Counter clock wise)으로 單位角 $2\pi/N$ Rad.의 整數倍로 回轉하는 Discrete motion phasor와의 形成電力을 나타낸다.

(3) FFT Algorithm, DFT의 週期性和 Redundancy

FFT의 Algorithm은 DFT의 計算時間을 大幅 단축 시켜 주었는데⁽⁴⁾⁽⁵⁾ 그 原因은 한번 계산한 결과를 多重으로 利用하여 쓸데없이 번번이 새로 計算하는 것을 피한다는 데에 있다.

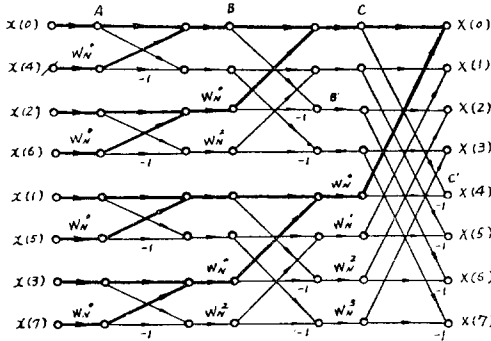
예를 들어 第1圖은 Butterfly Computation을 利用한 8-point DFT의 경우인데 $X(0)$ 值를 계산키 위해서 $x(0)$ 부터 $x(7)$ 까지가 두개씩 묶여서 計算되어

$$X_{m+1}(p) = X_m(p) + W_N^r X_m(q) \quad (16a)$$

$$X_{m+1}(q) = X_m(p) - W_N^r X_m(q) \quad (16b)$$

와 같이 이미 계산된 $X_m(p)$, $X_m(q)$ 량이 W_N^r 만을 새로 알음으로써 그 다음단계인 $X_{m+1}(p)$, $X_{m+1}(q)$ 를 計算하는데 使用되고 있음을 알수있고 第1圖에서 中間에 計算되었던 A,B,C點이 다시 A', B', C' 點등을 通해서 다시 利用되고 있음을 알 수 있다. 한마디로 해서 計算方法에 있어서의 Redundancy를 極少로 줄인 것이 FFT인 것이다.

그러나 計算過程에 있어서의 Redundancy는 없을망정 計算對象에 있어서 다음과 같은 두 가지 本質的인 Redundancy가 섞여 있어서 計算時間을 浪費하고 있다.



第1圖 8-point FFT

㉔ Discreteness에서 오는 Spectrum 중복(重疊)定理

(15) 式에서 計算되는 line spectrum 群은 N 을 週期로 하는 週期函數이다.

(證明)

$$F(k+mN) = \frac{1}{(m+1)N} \sum_{n=0}^{(m+1)N-1} \tilde{x}(n) e^{-j(2\pi/N)nk}$$

$$= \frac{1}{(m+1)N} (m+1) \sum_{n=0}^{N-1} x(n) e^{-j(2\pi/N)nk}$$

$$= F(k) \quad (17)$$

但 $x(n)$ 은 $\tilde{x}(n)$ 의 처음 한 週期만을 메어낸 函數이다.

$$\text{이식은 } e^{-j(2\pi/N)(n+mN)k} = e^{-j(2\pi/N)nk} e^{-j(2\pi/N)mNk} = e^{-j(2\pi/N)nk} e^{-j2\pi mk}$$

에서 phasor의 주기성 $e^{-j2\pi mk} = 1$ 때문에 成立하는 것이다. Q.E.D.

Collolary 1

(15) 式에서 $\tilde{x}(n)$ 가 實數列이라면 負周波數 $k(-N \leq k < 0)$ 에 對한 line spectrum은 redundancy로서

$$F(k) = F^*(-k) \text{이다.}$$

이는 (15)式에서 取扱한 series $\tilde{x}(n)$ 이 實函數임과 實函數의 Transform이 frequency domain에서 原點에 對해서 點對稱인 것에서 나온다.

Collolary 2

(15)式에서 $\tilde{x}(n)$ 가 實數列이라면 $\frac{N}{2} < k \leq N$ 사이의 line spectrum은 Redundancy로서

$$F(k) = F^*(N-k) \text{이다. 이는 } F(k) = F^*(-k) = F^*$$

定理와 Collolary 1로서 다음과 같이 증명된다.

$$F(k) = F^*(-k) = F^*(N-k)$$

$k=0$ 로 부터 k 를 증가시키면서 line spectrum을 구하는 것은 phasor를 W_N^k 를 단위로하여 2倍 3倍 등으로 phasor의 回轉을 증가시켜 Series와의 사이의 Correlation을 計算하는 것인데 k 가 $\frac{N}{2}$ 을 초과하면 phasor의 回轉이 180° 를 넘어서서 反對回轉과 또는 $k+m$ 배의 全回轉과 구별될 수 없다는 物理的인 뜻을 갖는다.

㉕ Over frequent Sampling에서 오는 Redundancy 第1圖에서 $X(0)$ 라는 最低周波數 成分을 求할때에 특히 나타나는 Redundancy로서 Sampling이 $X(3)$ 의 計算을 爲해서 充分한 것이라면 그 보다 낮은 周波數成分 計算을 爲해서는 必然的으로 과잉한 것으로 Nyquist의 Sampling rate를 감안하여 $x(n)|_{n=0,1,\dots,7}$ 이 $X(3)$ 를 爲해서 充分한 rate이라면 $X(2)$ 를 爲해서는 $x(0), x(2), x(4), x(6), X(1)$ 을 爲해서는 $x(0) x(4)$ 만 이면 足하며 FFT의 경우에서는 어느 周波數成分을 爲해서나 $x(n)|_{n=0,\dots,7}$ 全體를 計算對象으로 삼고 있으니 計算時間의 浪費가 심하다.

(4) Redundancy의 除去에 依한 計算時間의 단축, DFT와 FFT

㉔ DFT의 경우

DFT의 경우는 前記 ㉔ 및 ㉕항의 redundancy가 모

두다 除去될 수가 있다. 즉 이는 N 이상의 周波數點에서는 phasor 回轉의 結果가 modulo N 로서 항상 $\pm N$ 이하의 回轉의 경우와 아무런 差를 나타내지 못하기 때문에 N 이상의 周波數點에서 計算할 必要가 전혀 없고 $\frac{N}{2}$ 이상 N 이하의 경우에 있어서도 phasor 回轉結果가 負의 回轉과 一致하여 버리기 때문에 實質上에 있어서 $-N/2 \sim N/2$ 만이 有效하여 이것마저 實函數의 Transform의 對稱性에 따라 絕對値는 線對稱, 角度는 原點에 對한 點對稱이므로 $0 \sim \frac{N}{2}$ 만의 계산단 하면된다.

또한 ④項에서 오는 Redundancy는 다음의 計算單을 行함으로써 除去될 수 있다.

即 $x(n)$ 로서 n 번째의 Sequence值 그리고 $X(m)$ 로서 m 번째의 line spectrum의 높이와 위상을 한꺼번에 ($X(m)$ 이 複數) 나타낸다고 하면

$$\begin{aligned} X(0) \text{ 계산에는 } & x(N+1), x(0) \\ X(1) \text{ " } & x(N+1), x(0), x(4) \\ X(2) \text{ " } & x(N+1), x(0), x(2), x(4), x(6) \\ X(3) \text{ " } & x(N+1), x(0), x(1), x(2), x(3) \end{aligned}$$

등만이 必要하게 되기 때문에 DFT 계산에서는 DO Loop 등을 걸쳐 階段을 2 배씩 증가시켜 주면서 計算을 行하면 절약된다.

⑤ FFT의 경우

計算에 있어서의 redundancy를 最大限으로 줄였기 때문에 한번 計算된 것은 다음에 多重으로 使用되어서 計算을 안하고 지날 수가 없다. 하지만 마지막 段에서는 出力 Data redundancy를 淸정하여 butterfly 계산에서 淸편을 제거할 수 있다. 마지막단이 가장 복잡하기 때문에 相當한 낭비를 막을 수 있고, 더군다나 printing 시간은 完全히 半減된다. 電算機는 入出力機器에서 時間을 많이 소비하는 것이고 보면 N 의 값이 상당히 큰 경우에는 機械時間은 거의 半減된다. 이를 위한 原 program과 modify된 program을 表 1에 보인다.

(5) 結果와 검토

이 새 Algorithm에 의한 計算時間을 在來의 FFT에 의한 경우와 비교하여 表 2에 보였다. 256點의 경우

表 2 Duration time 대비표

Sample數	종래의 FFT	본제의 FFT
64	1분 50초	1분 46초
128	2분 14초	2분 5초
256	3분 28초	3분 24초

두 방법 사이에는 1분 이상의 差가 나는 것을 볼 수 있다. 이 計算에 使用된 計算機는 서울大學校 本部에 설치되어 있는 IBM 360이었으며,

表 2에 나와있는 時間은 Duration time이다.

(6) 結 論

Continuous quantity를 sampling하여 discrete series를 얻어서 digital computer에 依해서 解析할 때 本質적으로 포함되는 redundancy는 discrete motion phasor $e^{j(2\pi/N)kn}$ 의 주기성으로 보아 本質적이며 불가피하다. 따라서 實數列의 경우라면 Frequency domain에 있어서 0부터 $N/2$ 까지 以外の 成分을 計算함은 不必要한 努力이다.

Discrete signal processing의 경우에 있어서도 Transform(spectrum), Convolution sum 및 Correlation Sequences는 T.D.C.F.의 경우에 準하여 時間密度로 取扱함이 그 物理的 뜻을 파악함에 있어서 더욱 自然스럽다. 특히 Correlation Sequences의 도입은 T.D.C.F.의 定義가 도약이 없고 더 큰 體系를 이룰 수 있다.

Redundancy를 最少로 줄이는 Algorithm을 만들어 FFT의 경우에 있어서도 計算機時間을 大幅감축시킬 수 있다.

IBM 360을 使用하여 實際적으로 立證한바 $N=256$ 의 경우에서 이미 3분 28秒였던 것이 2분 24秒로 減속되었고 이 경향은 N 의 증가에 따라 더욱 현저하다.

參 考 文 獻

- (1) Samuel J. Mason & Henry J. Zimmerman "Electronic Circuits, Signals and Systems" John Wiley & Sons INC. 1960.
- (2) 안수길 : "Time Dependent Correlation Function과 그의 應用에 관한 研究" 전자공학회지 제10권 6호 1973年 12月.
- (3) 안수길 : "Time Dependent Fourier Transform, Time, Dependent Spectrum Density 및 그의 應用" 전자공학 회지 제13권 4호 1976年 10月 :
- (4) Alan V. Oppenheim and Ronald W. Schaffer "Digital Signal Processing" Prentice-Hall 1975.
- (5) B. Gold and C.M. Rader "Digital Processing of Signals" McGraw-Hill Book Company, New York 1969.

表 1.

Modify된 FFT Program

DOS FORTRAN IV 360N-FO-479 3-6	FFT	DATE 02/25/77
0001	SUBROUTINE FFT(X,M)	
0002	COMPLEX X(128), U,W,T	
0003	N=2**M	
0004	NV2=N/2	
0005	NM1=N-1	
0006	J=1	
0007	DO 7 I=1, NM1	
0008	IF(I.GE.J) GO TO 5	
0009	T=X(J)	
0010	X(J)=X(I)	
0011	X(I)=T	
0012	5 K=NV2	
0013	6 IF(K.GE.J) GO TO 7	
0014	J=J-K	
0015	K=K/2	
0016	GO TO 6	
0017	7 J=J+K	
0018	PI=3.141592653	
0019	DO 20 L=1, M	
0020	LE=2**L	
0021	LE1=LE/2	
0022	ME1=LE1	
0023	ME=LE1/2+1	
0024	U=(1.0,0,0)	
0025	W=CMPLX(COS(PI/FLOAT(LE1)), -SIN(PI/FLOAT(LE1)))	
0026	IF(L.GE.M) ME1=ME	
0027	DO 20 J=1, ME 1	
0028	DO 10 I=J,N,LE	
0029	IP=I+LE1	
0030	IP1=LE1+2-I	
0031	T=X(IP)*U	
0032	X(IP)=X(I)-T	
0033	IF(L.GE.M) X(IP1)=X(IP)	
0034	10 X(I)=X(I)+T	
0035	20 U=U*W	
0036	RETURN	
0037	END	