

복소수의 대수적 정수환 근사화를 이용한 이산 후리에 변환

(The Discrete Fourier Transform Using the Complex Approximations of the Ring of Algebraic Integer)

金德顯*, 金在功**

(Duck Hyun Kim and Jae Kong Kim)

要約

본 논문은 대수적 정수 환으로 근사화 된 복소 입력을 곱셈 없이 순회 이동과 덧셈만으로 DFT 가 가능한 방법을 제안 하였다. 곱셈 없이 DFT 가 가능함으로 연산 속도를 개선할 수 있었고 레인지 증가를 줄일 수 있었다. 또한 뒤들 팩터의 양자화 오차를 없앨 수 있어 정밀한 계산이 가능하였다. 제안된 방법의 타당성을 예제를 통해 입증하였으며, 응용 가능한 크기는 16 이었다.

Abstract

This paper presents a multiplier free technique for the complex DFT by rotations and additions based on the complex approximation of the ring of algebraic integers. Speeding-up the computation time and reducing the dynamic range growth has been achieved by the elimination of multiplication. Moreover the DFT of no twiddle factor quantization errors is possible. Numerical examples are given to prove the algorithm and the applicable size of the DFT is 16 has been concluded.

1. 서론

DFT (Discrete Fourier Transform) 는 신호 처리의 기본이며 프로세서의 크기와 계산 속도는 연산량에 좌우한다. 신호처리 기술 발달로 인해 DFT 프로세서의 크기와 계산 속도가 개선되어 오고 있지만, 많은 데이터의 실시간 처리에는 여전히 연산 속도의 개선 및 하드웨어 감소를 필요로 하고 있다.^[1] 연산 속도를 개선하려면 곱셈수가 적어야 한다. 곱셈 수를

줄이려면 뒤들 팩터를 단순한 형태로 분해한 후 복소 연산이나, 유한 체 연산 특성을 이용하여, DFT 핵 (Kernel)내의 복소 곱셈을 없애므로서 전체 곱셈 수를 줄이는 시도가 계속되고 있다.^[2] DFT 하드웨어의 구현은 구성 방식에 따라 직렬과 병렬 방식으로 나뉘어 진다. 입력수가 $N=d^m$ 이면, 직렬 구성의 경우 계산 복잡도는 $O(N)$ 이고, 병렬 구성에서는 $O(m)$ 이다. 어느 경우이던 소프트웨어적인 $O(Nm)$ 보다 속도가 개선된다.^[3] 단위 DFT 입력이 d 점이면 필요한 총 입력력 횟수는 $O(Nm/d)$ 이다. 따라서 속도를 향상시키려면 d 을 크게 택해야 한다. 즉 I/O 대역폭은 처리속도에 비해 매우 늦으므로 가능한 한 액세스 횟수가 적어야 한다. 다시 말해 d 를 크게 택해야 한다. 그러나 DFT 단위 블록을 하나의 칩에 집적하기 위해서는 d 는 무조건 크게 할 수 없다. 또한

* 正會員, 光州大學校 電子工學科
(Dept. of Elec. Eng., Kwangju Univ.)

** 正會員, 東國大學校 電子工學科
(Dept. of Elec. Eng., Dongguk Univ.)
接受H字: 1993年 2月 10日

DFT 단위 블록을 크게 집적하기 위해서는 곱셈기의 수가 적어야 한다. 왜냐하면 칩구현에서 곱셈기는 비교적 넓은 면적을 차지하기 때문이다. [1] DFT 프로세서의 연산은 구성에 따라 실수와 정수 연산으로 나눌 수 있으며, 연산의 정밀도와 속도는 서로 상반 관계에 있다. 즉 일반적으로 실수 연산은 정밀도가 높으나 속도가 느리고 하드웨어가 복잡하며 정수 연산은 그 반대이다. 정수 연산에 의한 프로세서는 실수보다 약 10 여배 빨라 [4] 속도를 필요로 하는 목적에 이용되고 있다. 정수 연산에 의한 DFT 계산에서 오차를 줄이기 위해서는 워드 길이(Word length)를 크게 해야하고, 이 경우 양자화 및 스케일링 오차는 줄어드나 하드웨어 및 연산 시간이 증가한다. [4] 이같은 정수 DFT 계산에서 다이내믹 레인지의 증가를 억제하기 위한 스케일링은 오차의 원인이 된다. Games [5-6] 는 레인지의 증가를 줄이기 위해 스케일링 대신 복소수의 대수적 정수 근사화를 이용하였고, Cozzens 와 Finkelstein [7-8] 은 Games 의 이론에 RNS (Residue Number System)를 응용, 레인지의 증가가 적은 고속 DFT 의 가능성을 시사하였다. 본 논문에서는, RNS 없이 대수적 정수가 지닌 레인지 증가 억제의 장점을 살린, 워드 단위 순회 이동 (Rotation)과 덧셈만의 복소 DFT 방법을 제안한다. II 장에서는 복소수의 대수적 정수 연산 특성 및 상호 변환을 요약하며 II 장 이후 복소수는 대수적 정수로 근사화 되었다고 전체 한다. III 장에서는 DFT 및 IDFT 를, 그리고 IV 장에서는 이론의 타당성을 예제를 통해 입증한다. V 장에 연산수 및 레인지를 종래의 정수 DFT 와 비교하고, 결론을 VI 장에 남긴다.

II. 대수적 정수와 변환

1. 대수적 정수와 연산 특성 [5, 7]

먼저 대수적 정수와 연산특성을 간단히 요약한다. 이 내용은 차후 응용된다. 대수적 정수란 최대 차수의 계수가 1 인 다항식의 근이다. 즉

$$U(x) = x^n + u_{n-1}x^{n-1} + \dots + u_kx^k + \dots + u_1x^1 + u_0x^0, u_k \in Z \quad (1)$$

의 근이다. 여기서 n 은 차수이다. 예를 들어 $\sqrt{2}$ 는 다항식 $x^2-2=0$ 을 만족하므로 대수적 정수이다. 만일 n 이 $n > 3$ 인 2 의 배수이면 식(1) 은

$$U(\omega) = u_{\phi(n)-1}\omega^{\phi(n)-1} + \dots + u_k\omega^k + \dots + u_1\omega^1 \\ = u_{n/2-1}\omega^{n/2-1} + \dots + u_k\omega^k + \dots + u_1\omega^1 \quad (2)$$

여기서 $\phi(n)$ 은 오일러의 파이 함수(Euler's phi function) 이고, $u_k \in Z$, $\omega = e^{2\pi i/n}$, $i^2=-1$ 이다. 즉 u_k 는 정수이고 ω 는 1의 n 차 원시근 이다. 따라서 U (ω) 는 복소수 체 C 의 조밀 집합(Dense Subset) 이며, $\omega^{\phi(n)}=1$ 이므로 덧셈과 곱셈에 대하여 닫혀 있다. 즉 U(ω) 는 환을 형성하고 이같은 대수적 정수 환을 Z [ω] 로 표시한다. Z [ω] 는 정수 Z 에 대해 $\phi(n)$ 차원 벡터 공간을 이루며, 특히 $n=4$ 이면 가우시안 정수 이다. 임의의 실계수 복소수를 유한 차수의 대수적 정수로 표현하면

$$u = x + yi = [u_0, u_1, \dots, u_{\phi(n)-1}] \\ = u_0 + u_1\omega^1 + \dots + u_{\phi(n)-1}\omega^{\phi(n)-1} \quad (3) \\ = \sum_{k=0}^{\phi(n)-1} u_k \omega^k \\ = UB$$

여기서 $x, y \in R$, $U=(u_0, u_1, \dots, u_{\phi(n)-1})$ 이고, $B=[\omega^0, \omega^1, \dots, \omega^k, \dots, \omega^{n/2-1}]^T$ 는 기저 벡터이다. 따라서 u는 좌표 (x, y) 또는 $(u_0, u_1, \dots, u_{\phi(n)-1})$ 로 표시되는 정수 집합 Z [ω] 로 나타낼 수 있다. 유한 구간 $[-r/2, r/2]$ 에서 u의 대수적 정수환을 Z [ω] 라 하면

$$Z[\omega] = \{[u_0, u_1, \dots, u_k, \dots, u_{\phi(n)-1}] \in Z[\omega], -r/2 \leq r/2\} \quad (4)$$

Z [ω] 내의 두 요소 U 및 V 가 다음과 같다면

$$U = [u_0, u_1, \dots, u_{\phi(n)-1}] \quad (5) \\ V = [v_0, v_1, \dots, v_{\phi(n)-1}]$$

이들의 덧셈, 스칼라 곱셈, 벡터 곱셈, 및 순회 이동은 다음과 같이 정의 된다. [5]

1) 덧셈

$$U + V = [u_0 + v_0, u_1 + v_1, \dots, u_{\phi(n)-1} + v_{\phi(n)-1}] \quad (6)$$

2) 스칼라 곱셈

$s \in Z$ 이면

$$sU = [su_0, su_1, \dots, su_{\phi(n)-1}] \quad (7)$$

3) 벡터 곱셈

$$U \times V = [c_0, c_1, c_2, \dots, c_{\phi(n)-1}] \quad (8)$$

여기서

$$c_k = \sum_{j=0}^k u_{k-j} v_j - \sum_{j=k+1}^{\phi(n)-1} u_j v_{n-(j+k)} \quad k = 0, 1, \dots, \phi(n)-1 \quad (9)$$

즉 두 대수적 정수의 곱은 컨볼루션과 유사하게 표현되고 $\phi(n)$ 보다 높은 차수 항은 관계식 $\omega^{\phi(n)} = -1$ 때문에 $\phi(n)$ 차 이상부터는 차수가 감소되며 부호가 반전 된다.

4) 순회 이동

n 이 $n \geq 4$ 인 2 의 배수이면 $\phi(n) = n/2$, $\omega = e^{2\pi i/n}$ 이고 만일 $U = \{u_0, u_1, \dots, u_{(n/2)-1}\} \in Z[\omega]$ 이면, $\omega U \in Z[\omega]$ 이고

$$\omega U = [-u_{n/2-1}, u_0, u_1, \dots, u_{n/2-2}] \quad (10)$$

즉 U 가 $Z[\omega]$ 의 요소이면 ωU 또한 $Z[\omega]$ 의 요소이고 U 에 ω 를 곱하면 각 요소는 한자리씩 좌측에서 우측으로 자리 이동 된다. 이때 관계식 $\omega^{\phi(n)} = -1$ 때문에 $\phi(n)$ 보다 높은 차수항은 차수가 감소되며 부호가 반전 된다.

2. 복소수의 대수적 정수 근사화

복소수를 차수 n , 레인지 r 인 대수적 정수, $Z[\omega]$ 로 근사화 하는 문제를 먼저 생각하자. 입력과 거리가 가장 짧은 코드를 $(r+1)^{n/2}$ 경우에서 찾는 것과 같다. b 비트 2 진수가 $Z[\omega]$ 로 표현 되려면^[6]

$$r = \left\lceil (C_2 2^{2b-2})^{1/(n/2-2)} \right\rceil \quad (11)$$

여기서 $C_2 = n \sin(2\pi/n)/\pi (1 - \cos(2\pi/n))$ 이고, $\lceil \cdot \rceil$ 은 최대 정수를 의미한다. 윗식이 만족되면 2 진 복소수는 대수적 정수로 근사화 된다. 근사화 방법으로는 ROM 테이블 방법^[5-8] 과, 아날로그의 대수적 정수 직접 변환 방법^[10] 이 있으며, 전자는 실수부와 허수부의 아날로그 입력을 각각 A/D 변환하여 미리 계산된 ROM 테이블을 읽는다. 즉 b 비트 2 진수 2^b ($1 \leq j \leq b$)의 $Z[\omega]$ 를 계산하여 ROM 에 저장한 다음, 실수부와 허수부의 b 비트 2진 입력 $\sum_{j=0}^{b-1} u_j 2^j$ 에 해당하는 대수적 정수 $\sum_{j=0}^{b-1} \hat{u}_j 2^j \in Z[\omega]$ 구한다. 여기서 \hat{u}_j 는 근사화 값이다. 후자에서는 $[-2, 2]$ 의 레인지로 약 9.29 비트 해상도를 지닌 직접 변환기를 구성한 바 있다.

3. 대수적 정수의 복소수 역변환

대수적 정수 DFT 결과를 복소수로 다시 표현하려면 $Z[\omega]$ 요소와 기저 벡터를 곱해야 된다.^[7] 대수적 정수 $U = \{u_0, u_1, \dots, u_{n/2-1}\}$ 와 기저 벡터 B 의 곱

$$UB = \sum_{j=0}^{n/2-1} u_j \omega^j \quad (12)$$

식(12)에 필요한 복소 곱셈과 덧셈수는 각각 $n/2$, $n/2-1$ 이다. 그러나 u_n 는 정수값이므로 n 번의 실수 곱셈과 $n-2$ 번의 덧셈이 각각 필요하다. U 를 실수부와 허수부로 나누면

$$U = (R_r(U), I_m(U)) \quad (13)$$

여기서

$$R_r(U) = (U + \bar{U})/2 = \sum_{j=0}^{n/2-1} u_j \cos(2\pi j/n) \quad (14)$$

$$I_m(U) = (U - \bar{U})/2i = \sum_{j=0}^{n/2-1} u_j \sin(2\pi j/n) \quad (15)$$

이고 여기서 \bar{U} 는 U 의 공액이다. $\cos(\theta) = -\cos(\pi - \theta)$, $\sin(\theta) = \sin(\pi - \theta)$, $0 \leq \theta < \pi$ 의 관계를 이용, 식(14)-(15)을 다시 쓰면

$$R_r(U) = u_0 + \sum_{j=1}^{n/4-1} (u_j - u_{n-2-j}) \cos(2\pi j/n) \quad (16)$$

$$I_m(U) = u_{n/4} + \sum_{j=1}^{n/4-1} (u_j - u_{n-2-j}) \sin(2\pi j/n) \quad (17)$$

$n=16$ 이면 식(16)-(17)은

$$R_r(U) = u_0 + (u_1 - u_7) \cos(\pi/8) + (u_2 - u_6) \sqrt{2}/2 (u_3 - u_5) \sin(\pi/8) \quad (18)$$

$$I_m(U) = u_4 + (u_1 - u_7) \cos(\pi/8) + (u_3 + u_6) \sqrt{2}/2 (u_3 + u_5) \sin(\pi/8) \quad (19)$$

식(12)에 필요한 16 번의 곱셈 및 14 번의 덧셈 수는 식(18)-(19)을 이용하면 6 번의 곱셈과 12 번의 덧셈으로 줄어든다. 이같은 과정을 통해 대수적 정수는 역변환 되어 종래의 실수 계수의 복소수로 나타내진다. 이때 역변환 오차는 계수 값을 조정함으로써 근사화 오차 보다 적게 할 수 있으며 $\cos(\pi/8)$, $\sqrt{2}/2$, $\sin(\pi/8)$ 등의 계수 값은 기지이므로 ROM 테이블을 이용하면 곱셈 없이 수행 할 수 있다.^[7] 본 논문에서는 이같은 대수적 정수로의 근사화와 역변환을 전제로 한다.

III. DFT 와 IDFT

1. DFT

길이 N 인 시퀀스 $f(m)$ 의 DFT 는 다음과 같이 정의 된다.

$$\begin{bmatrix} F(0) \\ F(1) \\ \vdots \\ F(j) \\ \vdots \\ F(N-1) \end{bmatrix} = \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \dots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \dots & \omega^{N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^j & \omega^{2j} & \dots & \omega^{j(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(m) \\ \vdots \\ f(N-1) \end{bmatrix} \quad (20)$$

여기서 $w=e^{2\pi i/N}$ 이다. 식(20)에 필요한 복소 곱셈과 덧셈 수는 각각 $N^2(N-1)$ 이다. 대수적 정수의 차수 n 과 입력수 N 을 같게 택하면 $\phi(n) = N/2$ 이고, 복소 입력 $f(m)$ 이 식(3)에 의해 대수적 정수 근사화 되면

$$f(m) \cong \sum_{k=0}^{N/2-1} f_m \omega^k \quad (m = 0, 1, \dots, N-1) \quad (21)$$

식(21)를 행렬로 표시하면

$$[f(0), f(1), \dots, f(m), \dots, f(N-1)]^T = F_0 B \quad (22)$$

여기서

$$F_0 = \begin{bmatrix} f_{0,0} & f_{0,1} & \dots & f_{0,k} & \dots & f_{0,N/2-1} \\ f_{1,0} & f_{1,1} & \dots & f_{1,k} & \dots & f_{1,N/2-1} \\ \vdots & \vdots & & \vdots & & \vdots \\ f_{m,0} & f_{m,1} & \dots & f_{m,k} & \dots & f_{m,N/2-1} \\ \vdots & \vdots & & \vdots & & \vdots \\ f_{N-1,0} & f_{N-1,1} & \dots & f_{N-1,k} & \dots & f_{N-1,N/2-1} \end{bmatrix} \quad (23)$$

행렬 F_0 에서 행은 입력 시퀀스의 시간 지수이고 열은 대수적 정수의 차수이다. 식(22)를 식(20)에 대입하면

$$\begin{bmatrix} F(0) \\ F(1) \\ \vdots \\ F(j) \\ \vdots \\ F(N-1) \end{bmatrix} = \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \dots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \dots & \omega^{N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^j & \omega^{2j} & \dots & \omega^{j(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)(N-1)} \end{bmatrix} F_0 B \quad (24)$$

또는

$$F(j) = W_j F_0 B \quad (25)$$

여기서 $W_j = [\omega^0, \omega^j, \dots, \omega^{mj}, \dots, \omega^{j(N-1)}]$ 이다.

즉 식(25)에서 $F(0)$ 는 F_0 의 모든 행 요소와 ω^0 의 곱이며 $\omega^0=1$ 이므로 $F(0)$ 는 F_0 의 열 합과 같다. 즉 F_0 의 열 합을 구해 이를 대수적 정수로 역변환 하면 $F(0)$ 를 얻는다. $F(1) = W_1 F_0 B$ 이다. $W_1 F_0$ 의 곱

을 먼저 고려해보면 $w = e^{2\pi i/N} = \omega^1$ 이므로 W_1 의 순회 이동에 의해 F_0 의 m 행은 우측에서 좌측으로 m 자리 순회 이동 하며 차수가 $N/2$ 를 넘으면 부호가 반전된다. 즉 F_0 의 0 번째 행 요소는 변화가 없고, 1 번째 행 요소는 우측에서 좌측으로 1 자리 순회 이동 하며 1 열 0 행 요소는 $f_{1,1}$ 이 된다. 마찬가지로 m 행은 우측에서 좌측으로 m 자리 순회 이동 하고 차수 $N/2$ 를 넘을 때마다 부호가 반전된다. 이같은 결과를 F_1 이라 하면

$$F_1 = \begin{bmatrix} f_{0,0} & f_{0,1} & \dots & f_{0,m} & \dots & f_{0,N/2-1} \\ f_{1,1} & f_{1,2} & \dots & f_{1,m-1} & \dots & -f_{1,0} \\ \vdots & \vdots & & \vdots & & \vdots \\ -f_{N-1,N/2-1} & f_{N-1,0} & \dots & f_{m,N/2-m} & \dots & f_{N-1,N/2-2} \end{bmatrix} \quad (26)$$

따라서 식(26)의 열 합은 식(27)과 같다. 즉

$$F(1) = \sum_{k=0}^{N/2-1} f_1(k) \omega^k = [f_1(0), f_1(1), f_1(2), \dots, f_1(k), \dots, f_1(N/2-1)] B \quad (27)$$

여기서

$$f_1(k) = \sum_{m=0}^{N-1} F_1(m, k) \quad k = 0, 1, \dots, N/2-1 \quad (28)$$

이고, $F_1(m, k)$ 는 행렬 F_1 의 m 행 k 열을 의미한다.

마찬가지로 $F(2) = W_2 F_0 B$ 이며, 여기서 $W_2 F_0$ 의 곱셈을 먼저 고려해보자. 순회이동 정의에 의해 F_0 의 m 행은 좌측에서 우측으로 $2m$ 자리 순회 이동 하며 차수가 $N/2$ 를 넘으면 부호가 반전된다. 즉 F_0 의 0 번째 행 요소는 변화가 없고, 1 번째 행 요소는 좌측에서 우측으로 2 자리 순회 이동하면 0 열 1 행 요소는 $f_{2,1}$ 이 된다. 마찬가지로 m 행은 우측에서 좌측으로 $2m$ 자리 순회 이동 하고 차수 $N/2$ 를 넘을 때마다 부호가 반전된다. 이 결과를 F_2 라 하면

$$F_2 = \begin{bmatrix} f_{0,0} & f_{0,1} & \dots & f_{0,m} & \dots & f_{0,N/2-1} \\ f_{2,1} & f_{3,1} & \dots & f_{m-1,m} & \dots & -f_{1,1} \\ \vdots & \vdots & & \vdots & & \vdots \\ -f_{N-1,N/2-2} & f_{N-1,N/2-1} & \dots & f_{m,N/2-m} & \dots & f_{N-1,N/2-3} \end{bmatrix} \quad (29)$$

가 되고 이는 F_1 의 m 행을 m 자리 순회 이동 시킨 결과와 일치한다.

따라서 식(29)의 열 합은 식(30)과 같이 된다. 즉

$$F(2) = \sum_{k=0}^{N/2-1} f_2(k) \omega^k = [f_2(0), f_2(1), f_2(2), \dots, f_2(k), \dots, f_2(N/2-1)] B \quad (30)$$

여기서

$$f_2(k) = \sum_{m=0}^{N-1} F_2(m, k) \quad k = 0, 1, \dots, N/2 - 1 \quad (31)$$

이고, $F_2(m, k)$ 는 행렬 F_2 의 m 행 k 열을 의미한다.

같은 방법으로 행렬 F_2 의 m 행을 m 번 순회 이동한 행렬 F_3 의 열 합 $f_3(k)$ 의 역변환은 $F(3)$ 이 된다. 이 과정을 N 번 반복한 N 제 계수행렬 f 의 열 합 $F(k)$ 을 구하고 이를 역변환하면 $F(j)$ 을 얻는다. 따라서 N 차 대수적 정수환 내에서 곱셈 없는 N 점 DFT가 가능하며 그 구성은 W 의 곱에 필요한 쉬프트 레지스터와 열 합에 필요한 가산기만으로 이루어 진다. 예로서 $Z(\omega)$ 상으로 근사화된 4 점 DFT 구조는 그림 1 과 같다.

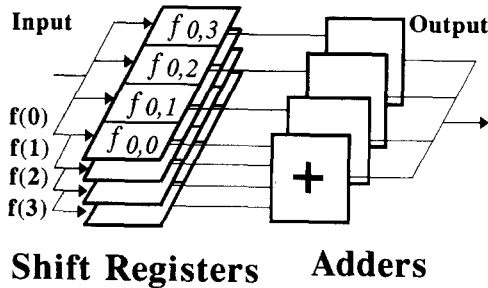


그림 1. $Z(\omega)$ 상의 4 점 DFT 블록도
Fig. 1. Block diagram for 4 point DFT in $Z(\omega)$.

N 점 DFT 프로세서는 각각 N 및 $N \log_2(N)$ 의 쉬프트 레지스터와 가산기로 구성 되고, 매 주파수 지수 마다 필요한 순회 이동과 덧셈 수는 $N-1$ 과 $(N-1) * N/2$ 이므로 DFT 에 필요한 총 순회 이동 및 덧셈 수는 $(N-1)^2$ 및 $N^2 * (N-1)/2$ 이다. 즉 제안된 방법의 계산 복잡도는 $O(N^3)$ 이다. 일반적으로 N 점 정수 FFT의 정확도는 \sqrt{N} 에 비례하는 것으로 알려져 있으며, 정밀도를 개선하기 위해 뒤들 팩터에 많은 비트를 할당한다. [11] 그러나 DFT 과정에서 값이 커지므로 적절한 스케일링이 요구되고 스케일 팩터가 크면 스케일 오차가 커진다. 제안된 DFT는 근사화 입력 계수의 순회 이동만으로 이루어 지므로 뒤들 팩터의 양자화 오차가 없고, 또한 곱셈이 없으므로 레인지의 증가를 억제할 수 있어 스케일 팩터를 작게 선택할 수 있어 DFT 계산의 정밀도를 높일 수 있다.

2. IDFT

길이 N 인 시퀀스 $F(j)$ 의 IDFT는 정의 식으로부터

$$\begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(m) \\ \vdots \\ f(N-1) \end{bmatrix} = \frac{1}{N} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \dots & \omega^0 \\ \omega^0 & \omega^{-1} & \omega^{-2} & \dots & \omega^{-(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{-j} & \omega^{-2j} & \dots & \omega^{-j(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{-(N-1)} & \omega^{-2(N-1)} & \dots & \omega^{-(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} F(0) \\ F(1) \\ \vdots \\ F(j) \\ \vdots \\ F(N-1) \end{bmatrix} \quad (32)$$

복소 입력 $F(j)$ 이 식(3)에 의해 대수적 정수로 근사화 되면

$$F(j) = \sum_{k=0}^{N/2-1} F_{j,k} \omega^k \quad j = 0, 1, \dots, N-1 \quad (33)$$

DFT와 마찬가지로 계수 $F_{j,k}$ 를 행렬로 배열하면

$$f_0 = \begin{bmatrix} F_{0,0} & F_{0,1} & \dots & F_{0,k} & \dots & F_{0,N/2-1} \\ F_{1,0} & F_{1,1} & \dots & F_{1,k} & \dots & F_{1,N/2-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ F_{j,0} & F_{j,1} & \dots & F_{j,k} & \dots & F_{j,N/2-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ F_{N-1,0} & F_{N-1,1} & \dots & F_{N-1,k} & \dots & F_{N-1,N/2-1} \end{bmatrix} \quad (34)$$

식(34)를 식(32)에 대입하면

$$\begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(m) \\ \vdots \\ f(N-1) \end{bmatrix} = \frac{1}{N} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \dots & \omega^0 \\ \omega^0 & \omega^{-1} & \omega^{-2} & \dots & \omega^{-(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{-j} & \omega^{-2j} & \dots & \omega^{-j(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{-(N-1)} & \omega^{-2(N-1)} & \dots & \omega^{-(N-1)(N-1)} \end{bmatrix} f_0 B \quad (35)$$

식 (35)로부터

$$f(m) = W_m f_0 B \quad (36)$$

여기서 $W_m = [\omega^0, \omega^{-m}, \dots, \omega^{-mj}, \dots, \omega^{-m(N-1)}]$ 이다.

즉 식(36)에서 $f(0)$ 는 f_0 의 모든 행 요소와 ω^0 의 곱이며 $\omega^0=1$ 이므로 $f(0)$ 는 f_0 의 열 합과 같다. 또한 $f(1)=W_1 f_0 B$ 이며 이때 $W_1 f_0$ 의 곱을 먼저 고려하면 $\omega^{-1} = e^{j2\pi/N} = \omega$ 이므로 순회 이동 정의에 의해 f_0 의 j 행은 좌측에서 우측으로 j 자리 순회 이동 하며 차수가 $N/2$ 를 넘으면 부호가 반전된다. 즉 DFT와 동일하며 회전 방향이 반대이다. 마찬가지로 j 행은 좌측에서 우측으로 j 자리 순회 이동되고 차수 $N/2$ 를 넘을 때마다 부호가 반전된다. 이 결과를 f_i 이라 하면

$$f_j = \begin{bmatrix} F_{j,0} & F_{j,1} & \dots & F_{j,j} & \dots & F_{j,N/2-1} \\ -F_{j,N/2-1} & F_{j,0} & \dots & F_{j,j+1} & \dots & F_{j,N/2-2} \\ \vdots & \vdots & & & & \\ -F_{j,N-1,N/2-1} & F_{j,N-1,0} & \dots & F_{j,N-1,N/2-j} & \dots & F_{j,N-1,N/2-2} \end{bmatrix} \quad (37)$$

따라서 식(37)의 열 합에 기저 벡터를 곱하면 f (1)이 된다. 이같은 과정은 DFT와 순회 이동 방향만 다를 뿐 계산 방법은 같다. 또한 데이터를 $N=2^v$ 로 취하면 $1/N$ 항은 하위 v 비트를 제거하므로 오버헤드 없는 나눗셈이 가능하다.

IV 계산 예

예제 1. 이론의 타당성을 증명하기 위해 DFT와 IDFT를 예시한다. 다음 복소 입력의 4점 DFT를 계산한다.

$$\begin{aligned} f(0) &= -0.5858 - 0.4142i \\ f(1) &= 0.0 + 0.5858i \\ f(2) &= -0.1213 - 0.2929i \\ f(3) &= -0.1716 + 0.1716i \end{aligned}$$

윗 복소수를 대수적 정수환 $Z[e^{2\pi i/8}]_B$ 으로 근사화 하면

$$\begin{aligned} f(0) &\approx (-2, 0, 1, -2) \\ f(1) &\approx (0, -1, 2, -1) \\ f(2) &\approx (2, -1, -1, 2) \\ f(3) &\approx (-3, 0, 3, -4) \end{aligned}$$

식(23)형태로 배열하면

$$F_0 = \begin{bmatrix} -2 & 0 & 1 & -2 \\ 0 & -1 & -2 & -1 \\ 2 & -1 & -1 & 2 \\ -3 & 0 & 3 & -4 \end{bmatrix}$$

식(25)에 의해

$$\begin{aligned} F(0) &= \sum_{k=0}^3 f_0(k) \omega^k \\ &= [-3, -2.5, -5]B = -0.8787 + 0.0503i \end{aligned}$$

또한

$$F(1) = \sum_{k=0}^3 f_1(k) \omega^k$$

여기서 $f_1(k) = W_8 F_0[k]$ 이고 $\omega = e^{2\pi i/8}$, $\omega = e^{2\pi i/4}$ 이므로 ω 를 곱하면 대수적 정수환 내에서는 각 자리수 마다 우측에서 좌측으로 2 자리씩 순회 이동된다. 즉

$$F_1 = \begin{bmatrix} -2 & 0 & 1 & -2 \\ 0 & -1 & 0 & 1 \\ -2 & 1 & 1 & -2 \\ -3 & 4 & -3 & 0 \end{bmatrix}$$

그러므로

$$F(1) = [-5 \ 4 \ -1 \ -3]B = 0.0503 - 0.2929i$$

마찬가지로

$$F_2 = \begin{bmatrix} -2 & 0 & 1 & 2 \\ 0 & 1 & -2 & 1 \\ 2 & -1 & -1 & 2 \\ 3 & 0 & -3 & 4 \end{bmatrix}$$

$$F(2) = [3 \ 0 \ -5 \ 5]B = -0.5355 - 1.4645i$$

마찬가지로

$$F_3 = \begin{bmatrix} -2 & 0 & 1 & -2 \\ -2 & 1 & 0 & -1 \\ -2 & 1 & 1 & -2 \\ 3 & -4 & 3 & 0 \end{bmatrix}$$

$$F(3) = [-3 \ -2 \ 5 \ -5]B = -0.8787 + 0.0503i$$

예제 2. 예제 1의 DFT 결과를 IDFT한다. 예제 1의 결과를 식(31)형태로 배열하면

$$F_0 = \begin{bmatrix} -3 & -2 & 5 & -5 \\ -5 & 4 & -1 & -3 \\ 3 & 0 & -5 & 5 \\ -3 & -2 & 5 & -5 \end{bmatrix}$$

식(36)에 의해 f_0 의 열을 합하면

$$F(0) = \frac{1}{4}[-8 \ 0 \ 4 \ 8]B = -0.5358 - 0.4142i$$

마찬가지로 f_0 의 순회이동한 행렬 f_1, f_2, f_3 로부터 다음을 얻는다.

$$F(1) = \frac{1}{4}[0 \ -4 \ 8 \ -4]B = -0.0 - 0.5858i$$

$$F(2) = \frac{1}{4}[8 \ -4 \ -4 \ 8]B = -0.1213 - 0.2929i$$

$$F(3) = \frac{1}{4}[-12 \ 0 \ 12 \ -16]B = -0.1716 - 0.1716i$$

이상으로부터 IDFT 결과는 예제 1의 DFT 입력과 일치함을 알 수 있다.

V. 연산 수 및 레인지 비교

1. 연산 수의 비교

제안된 알고리즘과 정수 DFT의 연산 수를 비교한다. 입력이 N 일 때 DFT에 필요한 복소 곱셈 수

M_c 와 덧셈 수 A_c 는 각각 N^2 , $N(N-1)$ 이다. 1 번의 복소 곱셈은 4 번의 정수 곱셈 및 2 번의 정수 덧셈과 같고 1 번의 복소 덧셈은 2 번의 정수 덧셈과 같으므로 정수 곱셈 수 M_r 과 덧셈 수 A_r 는

$$M_r = 4 * N^2 \tag{38}$$

$$A_r = 2N^2 + 2N(N-1) \tag{39}$$

또한 제안된 방법은 하나의 주파수 요소에서, $(N-1)$ 번의 순회 이동과 $(N-1)$ 번의 정수 덧셈이 $N/2$ 번 필요하므로 총 순회 이동 수 R_n 과 덧셈 수 A_n 는

$$R_n = (N-1)^2 \tag{40}$$

$$A_n = N^2(N-1)/2 \tag{41}$$

본 알고리즘의 응용 가능한 최대 크기를 구하기 위해 다음 가정을 도입 한다. ^{[11]-[12]}

- (가) 1회의 정수 곱셈은 A_r 개의 정수 덧셈 시간이 소요된다.
- (나) 1회의 정수 덧셈에 필요한 클럭 수는 1이다.
- (다) 1회의 순회 이동에 필요한 클럭 수는 1이다.

정수 DFT 에 필요한 클럭 수는 가정 (가) 및 (나) 에 의해

$$C_r = A_r * M_r + A_n \tag{42}$$

또한 제안된 방법에 필요한 클럭 수는

$$C_s = (N-1)^2 + N^2(N-1)/2 \tag{43}$$

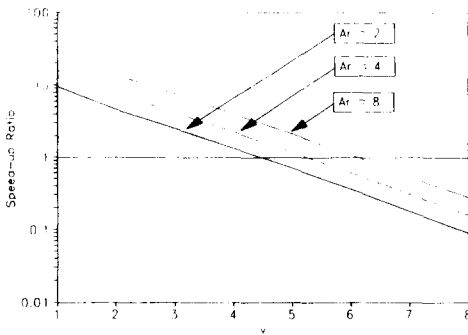


그림 2. 성능 개선비
Fig. 2. Speed-up ratio.

그림 2.는 $A_r = 2, 4, 8$ 일때 속도 개선비 C_r/C_s 를 $N = 2^v$ ($v = 1, 2, \dots, 8$) 에 대해 도사 하였다.

그림 2 를 보면 $v \leq 4$ 일때, 즉 $N \leq 16$ 에서 연산 시간은 정수 DFT 보다 적고 곱셈과 덧셈의 시간차가 클수록 속도는 더욱 향상 된다. 제안된 방법은 $O(N^4)$ 계산 복잡도를 가지므로, $O(N^3)$ 의 순회 이동 및 덧셈과 $O(N^2)$ 의 곱셈과의 관계에서 유용성을 찾을 수 있다. 또한 일반적으로 Radix-R FFT 알고리즘은 밀수가 4 보다 크면 곱셈수가 증가하나 ^[1] 제안된 방법에서 곱셈 수는 밀수 크기와 무관하다. 밀수가 크면 VLSI 화에 유리하다. ^[4]

2. 레인지 해석

b 비트 정수의 곱셈은 2b 비트, 덧셈은 b+1 비트의 결과를 생성한다. 이 결과를 유한 어장으로 표현하면 곱셈에서 반올림 오차, 덧셈에서 스케일링 오차가 발생 한다. 여기서는 b 비트 고정 소수점 DFT 에 필요한 전체 레인지를 비교한다. $N = 2^v$ 의 직접 DFT 에서 복소 곱셈의 실수부 및 허수부 계산은 다 같이 2b+1 비트 이고 덧셈은 1 비트 증가하므로 복소 DFT 에 필요한 총 비트 수 R_d 는

$$R_d = 2b + v + 1 \tag{44}$$

b 비트 레인지를 얻기 위한 대수적 정수의 레인지는 식(12)에 의해 $t = \log_2(r)$ 보다 커야 한다. 순회 이동은 레인지의 증가가 없고, 한번의 덧셈은 레인지를 1 비트 증가하므로 DFT 에 필요한 레인지 R_s 는

$$R_s = t + v \tag{45}$$

b = 8, 16 비트 일 때 레인지의 감소비 R_s/R_d 를 N 에 대해 도사하면 그림 3 과 같다.

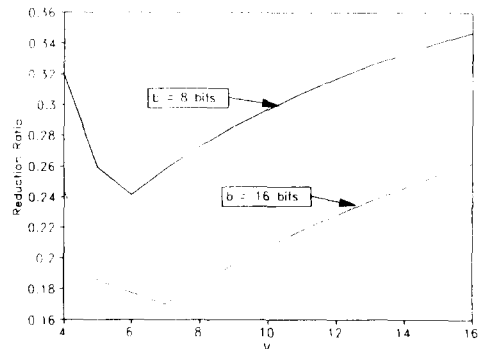


그림 3. DFT 에 필요한 레인지의 감소비
Fig. 3. Range Reduction Ratio for DFT.

그림 3. 을 보면 본 방법은 종래의 DFT 보다 다 이내믹 레인지가 적다. 이는 입력이 작은 정수로 근사화 되고 아울러 연산은 작은 정수 기리의 순회 이동 및 덧셈만으로 이루어지므로 다이내믹 레인지를 줄일 수 있기 때문이다. 대수적 정수 변환 및 역변환 과정 오차를 제외하면 제안된 방법은 DFT 계산에서 뒤들 팩터의 근사 오차가 없고 레인지의 증가가 적어 정밀 계산이 가능하다. 입력 수가 커지면 스칼라 곱셈 정리의 스케일링 기법을 적용하여 다이내믹 레인지의 증가를 억제 할 수 있다. 또한 그림 3 에서 $b = 8$ 비트일때 $N = 64$ 에서 레인지가 최소임을 알 수 있다. 그러나 그림 2 의 결과로부터 $N \leq 16$ 에서 속도가 종래의 DFT 보다 빠르므로, 두 조건을 동시에 만족하는 최대 값은 16 임을 알 수 있다.

VI. 결론

대수적 정수환 표현의 곱셈 없는 복소수 DFT 를 검토하였다. 뒤들 팩터의 근사화 오차 없이, 순회 이동과 덧셈만의 DFT 이므로 다이내믹 레인지의 축소가 가능하였고 연산 속도를 개선할 수 있었다. 그러나 제안된 방법의 계산 복잡도는 $O(N^3)$ 이고, 응용 가능한 최대 크기는 $N = 16$ 이었다. 상세한 프로세서의 설계 및 오차 해석은 차후 과제로 남긴다.

參考文獻

[1] S.Y. Kung, *VLSI Array Processors*, Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[2] L.R. Rabiner, G.D. Gold, *The Theory and Application of Digital Signal Processing*, Prentice Hall, Englewood Cliffs, New Jersey, 1975.

[3] G.D. Bergland, "Fast Fourier Transform Hardware Implementations - An overview," *IEEE Trans. on Audio and Electroacoustics*, vol. AU-17, pp. 104-108, June, 1969.

[4] H. Miyanaga, et al., "A Real-time 256x256 point Two-dimensional FFT Single-chip Processor," *Proceedings of ICASSP-91*, vol-2, pp. 1193-1196, April, 1990.

[5] R.A. Games, "Complex Approximations

Using Algebraic Integers," *IEEE Trans. on Information Theory*, vol. IT-31, No. 5, pp. 565-579, Sept. 1985.

[6] R.A. Games "An Algorithm for Complex Approximations in $Z [e^{j\omega}]$ " *IEEE Trans. on Information Theory*, vol. IT-32, no.4, pp. 603-607, July 1986.

[7] J.H. Cozzens, L.A. Finkelstein, "Computing the Discrete Fourier Transform using Residue Number Systems in a Ring of Algebraic Integers," *IEEE Trans. on Information Theory*, vol. IT-3, pp. 580- 587, Sept. 1985.

[8] J.H. Cozzens, L.A. Finkelstein, "Range and Error Analysis for a Fast Fourier Transform Computed over $Z [\omega]$," *IEEE Trans. on Information Theory*, vol. IT-33, no. 4, pp. 582-590, July, 1987.

[9] M.W. Marcellin, T.R. Fisher, "Encoding algorithms for Complex Approximations in $Z [e^{j\omega}]$," *IEEE Trans. on Information Theory*, vol. IT-35, no. 4, pp. 1133-1136, Sept. 1989.

[10] R.A. Games, et al "Algebraic Integer Quantization and Residue Number System Processing," *Proceedings of ICASSP-89 at Scotland*, March, 1989.

[11] P.D. Welch, "A Fixed-point Fast Fourier Transform Error Analysis," *IEEE Trans. Audio and Electroacoustics*, AU-21, no 2., pp. 151-157, June, 1969.

[12] E.E. Swartzlander, G. Hallnor, "High speed FFT Processor Implementation," *VLSI SignalProcessing*, pp. 27-34, IEEE Press, New York, 1984.

[13] W.A. Perera, "Architectures for Multiplierless Fast Fourier Transform Hardware Implementation in VLSI" *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP- 35, no. 12, Dec. 1987.

著者紹介



金 德 顯(正會員)

1958年 6月 13日生. 1982年 동국대학교 전자공학과 졸업(공학사). 1984년 동국대학교 대학원 전자공학과 졸업 (공학석사). 1985년 ~ 현재 동국대학교 전자공학과 박사과정 재학중. 1985년 ~ 현재 광주대학교 전자공학과 부교수. 주관심분야는 신호처리 알고리즘과 VLSI 구조 등임.

金 在 功(正會員) 第 27卷 第 8號 參照

현재 동국대학교 전자공학과 교수