

이산 푸리에 변환과 고속 푸리에 변환

이 책 ‘1절 이산 푸리에 변환’과 관련하여 기본 개념을 보충 설명하고, 행렬 표현 접근을 통해 DFT가 일종의 직교 변환임을 보였다. 또한 실제 응용에서 DFT에 의해 스펙트럼의 누설이 발생하는 데 대한 설명과 예를 추가하였다.

책의 ‘3절 원형 컨벌루션’과 관련하여 원형 컨벌루션과 선형 컨벌루션이 어떤 관계인지를 보충 설명하였다. 또한 원형 컨벌루션을 이용하여 긴 데이터의 컨벌루션을 계산하는 방법인 중복-더하기 방법 및 중복-저장 방법의 알고리즘을 정리하여 추가하였다.

책의 ‘4절 고속 푸리에 변환’에 대해서는 우선 DFT의 계산량에 대한 설명을 제시하였다. 그리고 신호 흐름 선도의 성질을 이용한 다양한 변형 FFT 선도를 소개하고, 역 FFT의 수식적 유도를 추가하였으며, 간단한 FFT 계산 예제를 보충하였다. 또한 밑수 2 FFT보다 일반화된 소인수 분해에 의한 FFT에 대해 비교적 상세하게 원리와 예를 소개하였다. 이 주제는 조금 더 고급의 FFT 학습이 필요한 사람들에게 유용할 것이다.

마지막으로 책의 내용과 심화학습 자료에서 새로이 제시된 내용과 관련된 연습 문제들을 몇 개 추가하여 실제적인 이해를 도울 수 있도록 하였다.

이상과 같은 보충 자료를 잘 학습하면 이 책 2부의 주제인 주파수 영역 해석의 실용적 방법인 DFT와 FFT에 대한 이해와 활용이 힘들지 않을 것이다.

11.1 이산 푸리에 변환(DFT)

11.1.1 이산 푸리에 변환의 정의

z 변환과 DFT의 관계

연속 신호와 시스템에 대한 라플라스 변환에 상응하는 이산 신호와 시스템의 주파수 영역 변환이 바로 다음 12장에서 다룰 z 변환인데, 라플라스 변환과 마찬가지로 z 변환도 푸리에 표현과 밀접한 관계가 있다. DFT는 DTFT를 샘플링한 것이고, DTFT는 z 평면의 단위 원 위에서 계산된 z 변환의 값에 해당하므로 DFT와 z 변환의 관계는 다음과 같이 된다.

$$\begin{aligned} X[k] &= \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} = \sum_{n=-\infty}^{\infty} x[n] \left(e^{j\frac{2\pi}{N}k} \right)^{-n} \\ &= \sum_{n=-\infty}^{\infty} x[n] z^{-n} \Big|_{z=e^{j\frac{2\pi}{N}k}} = X(z) \Big|_{z=e^{j\frac{2\pi}{N}k}} \end{aligned} \quad (C11.1)$$

즉 DFT는 z 평면상의 단위원 위에 등간격으로 위치한 N 개의 점에서 z 변환의 값을 구한 것과 같다.

DFT의 행렬 표현

(책)식 (11.3)의 DFT를 $k=0, 1, \dots, N-1$ 에 대해 풀어 쓰면 DFT는 결국 N 개의 미지수 $X[0], X[1], \dots, X[N-1]$ 에 대해 N 개의 연립방정식을 푸는 문제와 같게 된다.

DFT를 행렬 형태로 정리하여 나타내면 다음과 같다.

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ \vdots \\ X[N-1] \end{bmatrix} = \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \cdots & \cdots & W_N^0 \\ W_N^0 & W_N^1 & W_N^2 & \cdots & \cdots & W_N^{N-1} \\ W_N^0 & W_N^2 & W_N^4 & \cdots & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ W_N^0 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & \cdots & W_N^{(N-1)^2} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{bmatrix} \quad (C11.2a)$$

$$\mathbf{X} = \mathbf{W} \mathbf{x} \quad (C11.2b)$$

식 (C11.2)에서 회전 인자로 이루어진 계수 행렬 \mathbf{W} 를 DFT행렬이라고 한다. 회전 인자 $W_N^{kn} = W_N^{nk}$ 이므로 DFT행렬 \mathbf{W} 는 대칭 행렬이다. 즉 $\mathbf{W}^T = \mathbf{W}$ 이다.

한편 (책)식 (11.4)의 IDFT를 행렬 형태로 나타내면, $W_N^{-kn} = (W_N^{nk})^*$ 이므로 다음과 같다.

$$\begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{bmatrix} = \frac{1}{N} \begin{bmatrix} W_N^0 & W_N^0 & W_N^0 & \cdots & W_N^0 \\ W_N^0 & W_N^{-1} & W_N^{-2} & \cdots & W_N^{-(N-1)} \\ W_N^0 & W_N^{-2} & W_N^{-4} & \cdots & W_N^{-2(N-1)} \\ \vdots & \ddots & & \ddots & \vdots \\ W_N^0 & W_N^{-(N-1)} & W_N^{-2(N-1)} & \cdots & W_N^{-(N-1)^2} \end{bmatrix} \begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ \vdots \\ X[N-1] \end{bmatrix} \quad (\text{C11.3a})$$

$$\mathbf{x} = \frac{1}{N} \mathbf{W}^* \mathbf{X} \quad (\text{C11.3b})$$

그런데, 식(C11.2b)를 \mathbf{x} 에 대해서 풀면 $\mathbf{x} = \mathbf{W}^{-1} \mathbf{X}$ 가 되므로 다음이 성립한다.

$$\mathbf{W}^{-1} = \frac{1}{N} \mathbf{W}^* \quad (\text{C11.4})$$

\mathbf{W} 가 대칭 행렬이므로, 식 (C11.4)는 다음과 같이 다시 쓸 수 있다.

$$\mathbf{W}^{*T} \mathbf{W} = N \mathbf{I} \quad (\text{C11.5})$$

일반적으로 행렬 \mathbf{A} 가 $\mathbf{A}^{*T} \mathbf{A} = \mathbf{I}$ 를 만족하면 유니터리 행렬 unitary matrix이라 한다. 유니터리 행렬은 $\mathbf{A}^T \mathbf{A} = \mathbf{I}$ 를 만족하는 실수 직교행렬 orthogonal matrix을 복소 행렬에 확장시킨 개념이다.

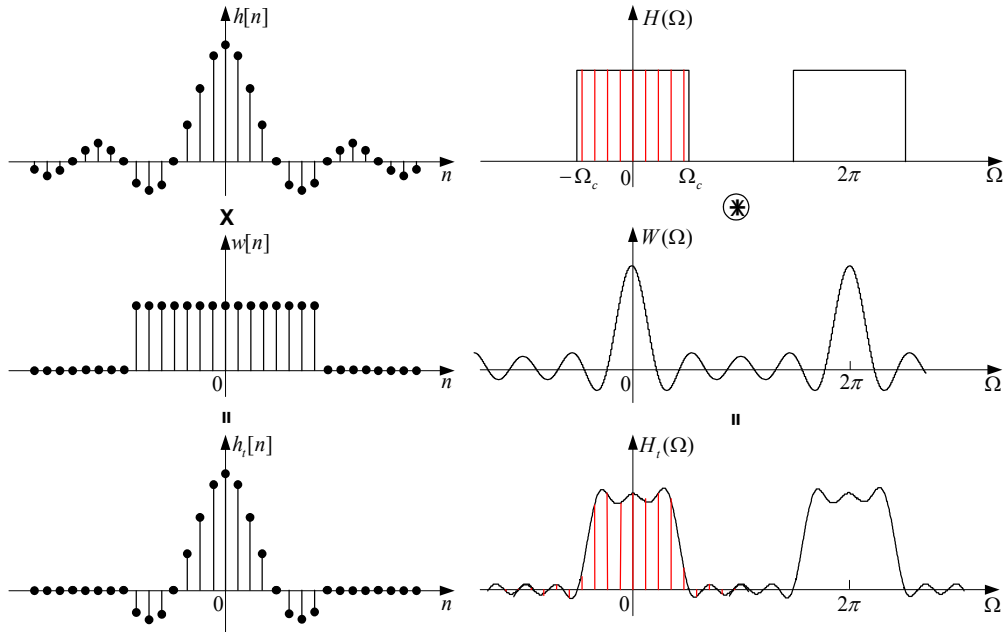
식 (C11.5)에서 보면 행렬 \mathbf{W} 가 엄밀히 말해 유니터리 행렬이 아니지만 IDFT의 계수 행렬에 곱해진 $1/N$ 을 쪼개어 DFT와 IDFT 양쪽의 계수 행렬에 $1/\sqrt{N}$ 을 똑같이 곱하여 이를 계수 행렬로 사용하면 식 (C11.2b)와 식 (C11.3b)는 각각 $\mathbf{X} = \mathbf{W} \mathbf{x}$ 와 $\mathbf{x} = \mathbf{W}^* \mathbf{X}$ 가 되므로 \mathbf{W} 는 유니터리 행렬이 된다. 따라서 **DFT는 유니터리 변환, 즉 직교 변환이다.** 직교변환의 유용성에 대해서는 6장에서 이미 상세히 설명한 바 있다.

11.1.3 DFT를 이용한 신호 처리

유한 구간 데이터와 DFT의 누설

DTFT는 주파수 영역으로 푸리에 변환해야 할 대상 신호가 통상적으로 무한한 시간 구간에 걸쳐 존재하는 것으로 가정하지만, 실제 응용에서는 이산 신호 데이터의 수가 유한한 경우가 거의 대부분이다. 이는 결국 무한한 길이의 이산 신호 $x[n]$ 에 유한한 길이를 갖는 (사각)창을 씌워 절단한 것으로 생각할 수 있는데, 우리는 이미 10장에서 DTFT의 주파수 컨벌루션 성질(창 씌우기 성질)을 통해 신호의 스펙트럼에 어떤 일이 일어나는지 살펴본 바 있다. 그 결과를 다시 돌이켜 보면, 시간 영역에서 신호 $x[n]$ 에 창 $w[n]$ 을 씌우면 주파수 영역에서 두 스펙트럼의 (원형) 컨벌루션이 되어 (책)[그림 10-12]와 같이 된다.

$$x[n]w[n] \Leftrightarrow \frac{1}{2\pi} X(\Omega) \circledast W(\Omega) \quad (\text{책})(10.47)$$



[그림 C11-1] 창 씌우기(주파수 컨벌루션)에 의한 DFT의 누설 효과

이를 다시 나타낸 [그림 C11-1]에서 보면 원래는 오른쪽 맨 위와 같이 사각 펄스로 나타나야 할 신호의 스펙트럼이 오른쪽 제일 밑과 같이 전 주파수에 걸쳐 맥동이 존재하는 스펙트럼으로 나타나게 된다.

DFT는 DTFT를 샘플링한 것이므로, 유한한 길이의 신호 데이터를 이용하여 DFT하게 되면 [그림 C11-1]의 오른쪽 제일 밑의 스펙트럼을 샘플링한 결과가 얻어져서 이상적으로 길이가 무한한 신호의 스펙트럼과는 차이가 난다. 특히 0이 되어야 할 주파수에 대해서도 값을 가지게 됨을 볼 수 있다. 이와 같이 창 함수 스펙트럼의 부엽에 의해 신호의 특정한 주파수 성분이 새어나가 주변의 주파수들로 번지는 현상을 **누설** ^{leakage}이라고 하며, 이론적으로는 무한하지만 유한한 개수의 데이터로 DFT할 때 관찰할 수 있는 현상이다.

■ 예제 C11-1 : 유한 구간 정현파의 DFT의 누설 효과

다음과 같이 길이 $N=2L+1$ 인 유한 구간 이산 정현파의 DFT를 구하여 그려라.

$$x[n] = \cos(\Omega_0 n) \text{rect}[(n-N)/2L] = \begin{cases} \cos(\Omega_0 n), & n = 0, 1, \dots, 2L \\ 0, & \text{그 외} \end{cases}, \quad 0 \leq \Omega_0 \leq \pi$$

<풀이>

무한 구간 이산 정현파 $x'[n] = \cos(\Omega_0 n)$ 의 DTFT는 다음의 스펙트럼이 2π 주기로 반복되는 주기 함수가 된다.

$$X'(\Omega) = \pi[\delta(\Omega + \Omega_0) + \delta(\Omega - \Omega_0)], \quad -\pi \leq \Omega \leq \pi$$

그리고 사각 펄스 $p_L[n] = \text{rect}[n/2L]$ 의 DTFT는 (책)[예제 10-4]에서 다음과 같이 구한 바 있으므로

$$P_L(\Omega) = \frac{\sin(\Omega(2L+1)/2)}{\sin(\Omega/2)} = \frac{\sin(\Omega N/2)}{\sin(\Omega/2)}$$

주어진 유한 구간 정현파 $x[n]$ 의 DTFT는 주파수 컨벌루션 정리에 의해 다음과 같이 된다.

$$\begin{aligned} X(\Omega) &= \frac{1}{2\pi} X'(\Omega) \otimes P_L(\Omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X'(\theta) P_L(\Omega - \theta) d\theta \\ &= \frac{1}{2} [P_L(\Omega + \Omega_0) + P_L(\Omega - \Omega_0)] \end{aligned}$$

따라서 유한 구간 정현파 $x[n]$ 의 N 점 DFT는 다음과 같이 된다.

$$\begin{aligned} X[k] &= X\left(\frac{2\pi}{N}k\right) = \frac{1}{2} \left[P_L\left(\frac{2\pi}{N}k + \Omega_0\right) + P_L\left(\frac{2\pi}{N}k - \Omega_0\right) \right], \quad -L < k \leq L \\ &= \frac{1}{2} \left[\frac{\sin\left(\left(\frac{2\pi}{N}k + \Omega_0\right)N/2\right)}{\sin\left(\left(\frac{2\pi}{N}k + \Omega_0\right)/2\right)} + \frac{\sin\left(\left(\frac{2\pi}{N}k - \Omega_0\right)N/2\right)}{\sin\left(\left(\frac{2\pi}{N}k - \Omega_0\right)/2\right)} \right], \quad -L \leq k \leq L \end{aligned}$$

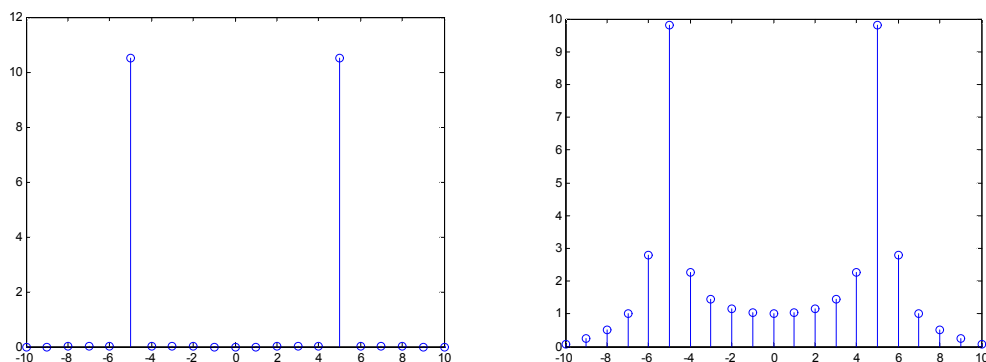
만약 $\Omega_0 = \frac{2\pi}{N}l$ 이면, $X[k]$ 는 다음과 같이 되므로

$$\begin{aligned} X[k] &= \frac{1}{2} [P_L\left(\frac{2\pi}{N}(k+l)\right) + P_L\left(\frac{2\pi}{N}(k-l)\right)] = \frac{1}{2} \left[\frac{\sin((k+l)\pi)}{\sin((k+l)\pi/N)} + \frac{\sin((k-l)\pi)}{\sin((k-l)\pi/N)} \right] \\ &= \begin{cases} \frac{N}{2}, & k = \pm l \\ 0, & -L \leq k \leq L, \quad k \neq \pm l \end{cases} \end{aligned}$$

정확히 $\frac{2\pi}{N}k = \pm \frac{2\pi}{N}l = \pm \Omega_0$ 에서만 값을 갖는 스펙트럼을 얻을 수 있을 것이다. 하지만 $\Omega_0 \neq \frac{2\pi}{N}l$ 인 경우에는 $\left| \frac{2\pi}{N}k \pm \Omega_0 \right|$ 를 0에 가장 가깝게 만드는 k 값에서 최댓값을 가지며, 그 주위에 작지만 0이 아닌 값들을 갖는 스펙트럼이 얻어질 것이다. 즉 누설이 발생한다.

[그림 C11-2]는 길이 $N=21$ ($L=10$)인 정현파의 DFT를 $\Omega_0 = \frac{10\pi}{21}$ 와 $\Omega_0 = \frac{10.5\pi}{21}$ 인 경우에 대해 각각 그린 것이다. 그림에서 볼 수 있듯이 $\Omega_0 = \frac{2\pi}{N}l$ 을 만족하는 $\Omega_0 = \frac{10\pi}{21}$ 의

경우에는 $k=\pm 5$ 에서만 10.5의 값을 갖는 스펙트럼이 얻어지지만, 그렇지 않은 경우에는 다른 주파수에서의 값이 0이 아닌 스펙트럼이 얻어지고, $k=\pm 5$ 의 샘플 스펙트럼의 값은 9.8093이 되어 누설이 발생했음을 알 수 있다.



$$(a) \Omega_0 = \frac{10\pi}{21}$$

$$(b) \Omega_0 = \frac{10.5\pi}{21}$$

[그림 C11-2] 유한 구간 정현파의 DFT의 누설 효과

이 예제에서 한 가지 더 눈여겨 봐 둘 점은 수식에서 볼 수 있듯이 DFT에 의해 얻어지는 스펙트럼의 값이 데이터의 길이(개수)에 종속적이라는 점이다. ■

DFT의 길이 N 의 선정

DFT를 이용하여 연속 신호를 처리할 때, 스펙트럼의 해상도와 정확도가 신호 $x[n]$ 과 DFT의 길이 N 과 밀접하게 관련되어 있으므로 N 의 선정은 중요한 문제이다. (책)식 (11.9)의 결과를 얻어내는 과정을 상세히 살펴보기로 한다.

지속 시간 t_s 인 연속 신호를 간격 T 로 샘플링할 때 주파수 중첩을 피하기 위해 지켜져야 할 조건은 샘플링 정리에 의해 다음과 같이 된다.

$$f_s = \frac{1}{T} \geq 2f_b \quad (C11.6)$$

여기서 f_s 는 샘플링 주파수, f_b 는 신호의 (유효) 대역폭이다. 샘플링된 이산 신호의 스펙트럼은 연속 신호의 스펙트럼이 f_s 를 주기로 하여 반복되고, DFT는 이 스펙트럼의 한 주기에 N 개의 샘플을 취하게 되므로 DFT에 의해 얻어진 샘플 스펙트럼의 샘플 간 간격(해상도)은 다음과 같다.

$$\Delta f = \frac{\text{한 주기 구간에 해당하는 주파수}}{\text{샘플 개수}} = \frac{f_s}{N} = \frac{1}{NT} \quad (C11.7)$$

그런데 N 점 샘플 스펙트럼을 IDFT하여 복원된 이산 신호는 주기 N 인 주기 신호로서 한 주기 길이가 연속 신호의 지속 시간보다 길어야만 시간 영역에서 파형의 중첩(절단 왜곡)이 발생하지 않는다. 샘플의 개수가 N 이고 샘플링 간격이 T 이므로 연속 신호로 환산된 길이는 시간 NT 에 해당된다. 따라서 다음을 만족해야 한다.

$$NT \geq t_s \quad (\text{C11.8})$$

식 (C11.6)과 식 (C11.8)을 동시에 만족시켜야 하므로 다음의 관계가 성립한다.

$$N \geq 2t_s f_b \quad (\text{책})(11.9)$$

11.2 이산 푸리에 변환의 특성

11.3 원형 컨벌루션

11.3.2 원형 컨벌루션에 의한 선형 컨벌루션의 계산

<Note> 원형 컨벌루션과 선형 컨벌루션의 관계

원형 컨벌루션은 신호 데이터를 원 위에 펼쳐서 한 바퀴 돌 때마다 동일한 데이터 패턴이 꼬리를 물고 반복되도록 하여 계산하는 반면, 선형 컨벌루션은 N 개 이외의 샘플 값은 0으로 간주하여 데이터를 직선 위에 펼쳐놓고 계산을 진행하므로 [그림 C11-3]과 같이 시간의 경과에 따라 이동 신호가 지나가면 그 자리로 다시 나타나지 않고 사라진다.

			1	2	3	4	
-----			-----				-----
d	c	b	a				
	d	c	b	a			
		d	c	b	a		
			d	c	b	a	
				d	c	b	a
					d	c	b
						d	c
							d

[그림 C11-3] 선형 컨벌루션 예

그런데 (책)[그림 11-7]과 [그림 C11-3]을 비교해보면 흥미로운 사실을 알 수 있다. (책)[그림 11-7]의 원형 컨벌루션의 이동 신호의 1행은 [그림 C11-3]의 선형 컨벌루션의 이동 신호의 1행과 5행을 더한 것과 같고, 나머지 행들에 대해서도 같은 원리를 관찰할 수 있다. 다시 말해 **두 신호의 원형 컨벌루션은 두 신호의 선형 컨벌루션 결과를 길이 반으로 자른 다음 더한 것과 같다.** 선형 컨벌루션 $z[n] = x[n] * h[n]$ 은 길이 $2N-1$ 인 신호이므로, 원형 컨벌루션 $y[n] = x[n] \otimes h[n]$ 은 다음과 같게 된다.

$$y[n] = [z[0] \ z[1] \ z[2] \ \cdots \ z[N-1]] + [z[N] \ z[N+1] \ \cdots \ z[2N-2] \ 0] \quad (\text{C11.9})$$

■ 예제 C11-2 : DFT를 이용한 선형 컨벌루션의 계산

두 신호 $x[n] = [1, 2, 3, 4]$, $h[n] = [0, 1, 2, 3]$ 의 선형 컨벌루션을 DFT를 이용하여 계산하라.

<풀이>

먼저 각 수열이 $N_1 + N_2 - 1 = 7$ 개의 샘플 값을 갖도록 영 채우기를 하면

$$x_a[n] = [1, 2, 3, 4, 0, 0, 0], \quad h_a[n] = [0, 1, 2, 3, 0, 0, 0]$$

이다. $x_a[n]$ 과 $h_a[n]$ 의 DFT와 이들의 곱인 컨벌루션 $y[n] = x[n] * h[n] = x_a[n] \otimes h_a[n]$ 의 DFT를 [표 C11-1]에 보였다.

[표 C11-3] [예제 C11-2]의 신호 및 컨벌루션 출력의 DFT

$k \backslash$	$X_a[k]$	$H_a[k]$	$Y[k]$
0	10	6	60
1	-2.0245-j6.2240	-2.5245-j4.0333	-19.9928+j23.8775
2	0.3460+j2.4791	-0.1540+j2.2383	-5.6024+j0.3927
3	0.1784-j2.4220	-0.3216-j1.7950	-4.4049+j0.4585
4	0.1784+j2.4220	-0.3216+j1.7950	-4.4049-j0.4585
5	0.3460-j2.4791	-0.1540-j2.2383	-5.6024-j0.3927
6	-2.0245+j6.2240	-2.5245+j4.0333	-19.9928+j23.8775

마지막으로 $Y[k]$ 의 IDFT를 구하면 다음과 같이 컨벌루션의 결과를 얻을 수 있다.

$$y[n] = [0, 1, 4, 10, 16, 17, 12]$$

사실 이 문제의 경우는 동심원을 이용하여 계산하는 것이 훨씬 편리하고 DFT와 IDFT를 손으로 계산하기가 까다롭지만 고속 푸리에 변환(FFT)를 이용하면 보통 직접 원형 컨벌루션을 계산할 때보다 계산량이 훨씬 줄어들므로 예제로 보인 것이다. ■

11.3.3 원형 컨벌루션을 이용한 긴 데이터의 컨벌루션

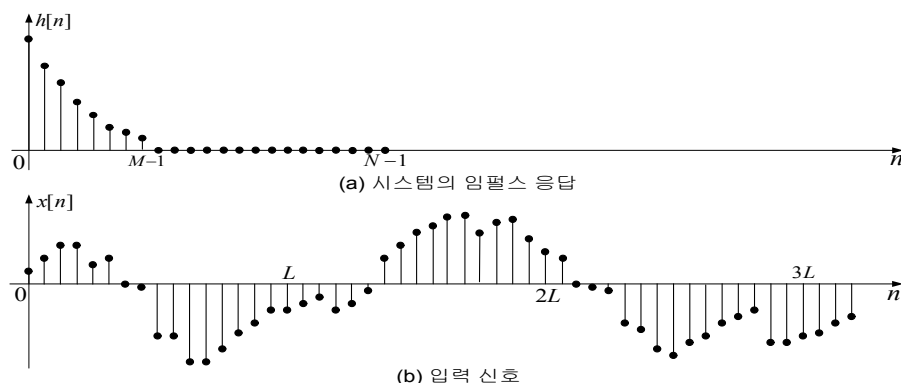
디지털 신호 처리에서는 하드웨어가 아니라 소프트웨어로도 시스템을 구현할 수 있는데, 이때 시스템의 출력을 입력과 임펄스 응답의 컨벌루션으로 처리하도록 프로그램하면 된다. 그런데 시스템이 긴 시간 동안 동작해야 하는 경우를 생각해보자. 예를 들어 MP3 플레이어로 3~4분짜리 노래를 들을 때 그 곡의 데이터 양, 즉 입력 신호의 길이가 엄청나게 길어서 컨벌루션 계산이 다 될 때까지 재생 버튼을 누른 뒤 연주 시간보다 더 긴 시간을 기다려야 한다면, 그리고 매번 노래를 들 때마다 이 과정을 반복해야 한다면 짜증나서 이 기기를 쓰려 하지 않을 것이다.

이처럼 입력 신호의 길이가 매우 길거나 실시간 처리가 필요한 경우라 할지라도 원형 컨벌루션을 이용해서 문제없이 시스템을 구현할 수 있는 좋은 해결책이 있다. 이 해결책의 기본 아이디어는 입력 신호를 (한 번에 실시간 처리가 가능한) 적당한 크기의 구간으로 나누어 N 점 원형 컨벌루션을 계속해서 수행하는 것이다. 이는 일종의 블록 처리로서, 기술적인 어려움은 4장([그림 4-13] 참조)에서 보았던 것처럼, 어떤 시간 순간 n_0 에서의 입력값 $x[n_0]$ 이 임펄스 응답 $h[n]$ 의 길이만큼 이후 출력에 계속 기여하기 때문에 그것을 짧은 구간으로 나누는 원형 컨벌루션의 결과에 어떻게 올바르게 반영해 주느냐 하는 데 있다.

이를 위해 두 가지 방법이 주로 사용된다. 하나는 그 영향을 원형 컨벌루션 계산한 후에 반영해주는 **중복-더하기**^{overlap-add} 방법이고, 다른 하나는 원형 컨벌루션 계산 과정에서 그 영향을 이미 포함시키는 **중복-저장**^{overlap-save} 방법이다.

중복-더하기 방법

컨벌루션할 임펄스 응답 $h[n]$ 과 긴 입력 $x[n]$ 이 [그림 C11-4]와 같다고 하자. 임펄스 응답의 길이는 M 으로 입력 신호에 비해 매우 짧다.



[그림 C11-4] 시스템의 임펄스 응답과 입력 신호

중복-더하기^{overlap-add} 방법은 입력을 중복 없이 길이 L 로 나누고 임펄스 응답과 입력 모두 $N = L + M - 1$ 샘플이 되도록 영 채우기를 한 뒤, DFT를 이용하여 원형 컨벌루션을 수행하여 각 블록별 컨벌루션 결과를 그대로 더하면 된다.

[그림 C11-5]에서 보면, 입력의 첫 번째 블록에 의한 원형 컨벌루션은 시간 L 에서부터 $N-1$ 까지는 입력이 없음에도 출력이 나타난다. 그 이유는 L 이전의 입력들이 $L-1$ 에 임펄스 응답 길이 M 을 더한 $N-1$ 순간까지의 출력에 계속 기여하기 때문이다. 두 번째 입력 블록에 의한 원형 컨벌루션 결과에서 L 순간부터 $N-1$ 순간까지의 결과는 과거 입력의 영향이 배제된 시간 L 순간부터의 입력 만에 의한 출력 값이므로, 이 둘을 더해야 온전한 출력을 얻게 된다.

이상과 같은 중복-더하기 방법을 정리하면 다음과 같다.

1. $h[n]$ 에 $L-1 = N-M$ 개의 영점 채우기를 시행한다.

$$h_a[n] = \begin{cases} h[n], & 0 \leq n \leq M-1 \\ 0, & M \leq n \leq N-1 \end{cases} \quad (\text{C11.10})$$

2. $x[n]$ 을 중복 없이 길이 L 로 나누어 각각 $M-1$ 개의 영점 채우기를 시행한다.

$$x_i[n] = \begin{cases} x[n], & iL \leq n \leq iL + L-1 \\ 0, & iL + L \leq n \leq iL + N-1 \end{cases} \quad (\text{C11.11})$$

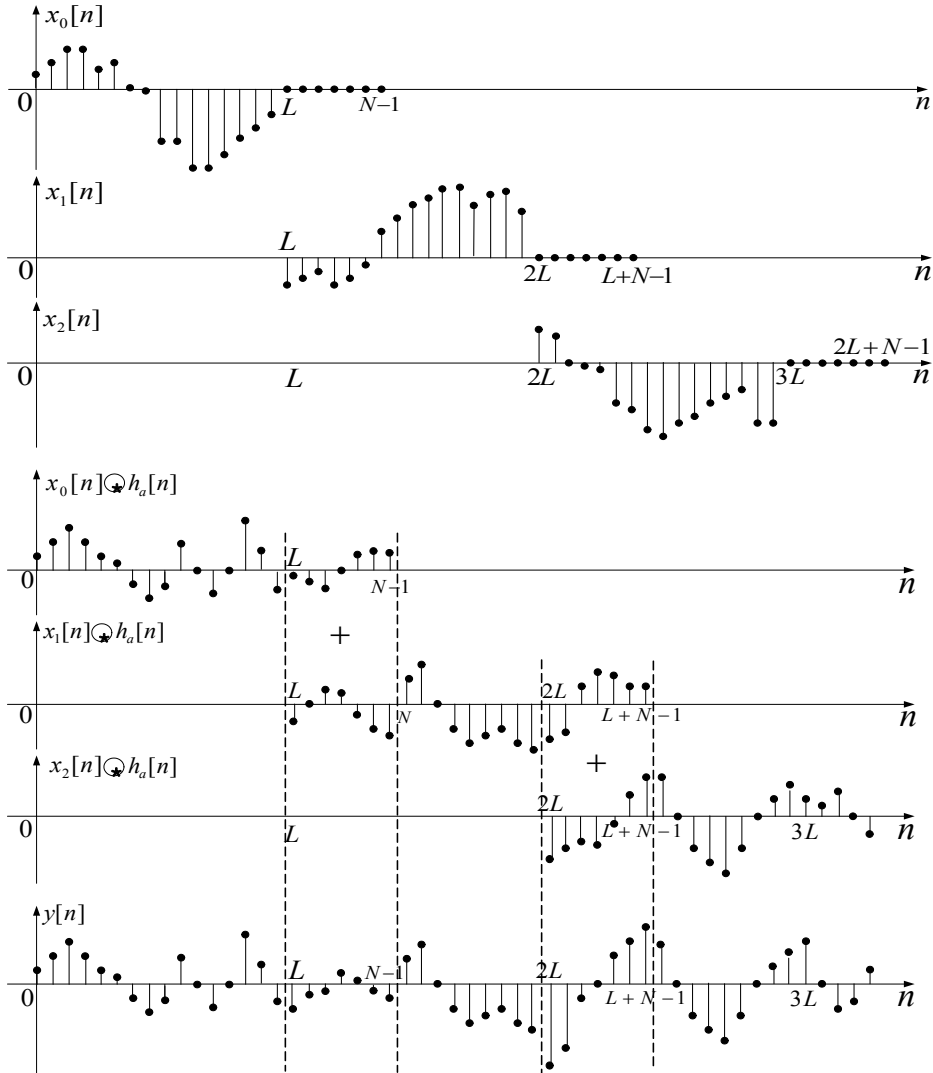
3. DFT와 IDFT를 이용하여 N 점 원형 컨벌루션을 수행한다.

$$y_i[n] = x_i[n] \circledast h_a[n], \quad iL \leq n \leq iL + N - 1 \quad (\text{C11.12})$$

4. 각각의 원형 컨벌루션 수행 결과를 그대로 더하여 출력으로 낸다.

$$y[n] = x[n] * h[n] = \sum_{i=0}^{\infty} y_i[n] \quad (\text{C11.13})$$

위의 알고리즘에서 $y_i[n]$ 은 $x[n]$, $iL \leq n \leq iL + L - 1$ 에 의한 응답 성분이고 $y_{i+1}[n]$ 은 $x[n]$, $iL + L \leq n \leq iL + N - 1$ 에 의한 응답 성분이므로 두 응답이 겹쳐지는 부분은 과거의 입력에 의한 응답과 현재의 입력에 의한 응답이 따로 계산되어 합쳐지는 부분에 해당된다. 따라서 출력이 식 (C11.13)과 같이 계산되는 것은 타당하다.



[그림 C11-5] 중복-더하기 방법에 의한 긴 데이터의 컨벌루션 계산

중복-저장 방법

중복-저장 방법^{overlap-add}은 임펄스 응답의 길이만큼 중복되게 입력 신호를 길이 $N=L+M-1$ 으로 나눈다. 그리고 임펄스 응답도 N 샘플이 되도록 영 채우기를 한 다음 원형 컨벌루션을 수행한다. 최종적으로 각 블록의 컨벌루션 결과들에서 앞의 $M-1$ 샘플을 버린 뒤 더하면 제대로 된 출력이 얻어진다.

[그림 C11-6]에서 보면, 입력 블록 앞부분의 중복되게 자른 $M-1$ 샘플은 단지 임펄스 응답의 길이에 의한 이전 입력의 영향을 시간 M 이후의 출력에 반영하기 위해 포함시켜둔 것이다. 그 결과 M 이후의 출력은 과거와 현재의 입력의 영향이 모두 반영된 온전한 출력 이므로, 이전 입력의 영향이 반영되지 않은 불완전한 출력인 중복된 입력 구간의 출력은 버려도 상관이 없다.

중복-저장 방법을 정리하면 다음과 같다.

1. $h[n]$ 에 $L-1 = N-M$ 개의 영점 채우기를 시행한다.

$$h_a[n] = \begin{cases} h[n], & 0 \leq n \leq M-1 \\ 0, & M \leq n \leq N-1 \end{cases} \quad (\text{C11.14})$$

2. $x[n]$ 을 $M-1$ 샘플씩 중복되게 길이 N 으로 나눈다. 이때 첫 번째의 경우는 중복 샘플이 없으므로 맨 앞부분에 $M-1$ 개의 영점을 채운다.

$$x_0[n] = \begin{cases} 0, & -M+1 \leq n \leq -1 \\ x[n], & 0 \leq n \leq L-1 \end{cases} \quad (\text{C11.15})$$

$$x_i[n] = x[n], \quad iL-M+1 \leq n \leq iL+L-1 \quad (\text{C11.16})$$

3. DFT와 IDFT를 이용하여 N 점 원형 컨벌루션을 수행한다.

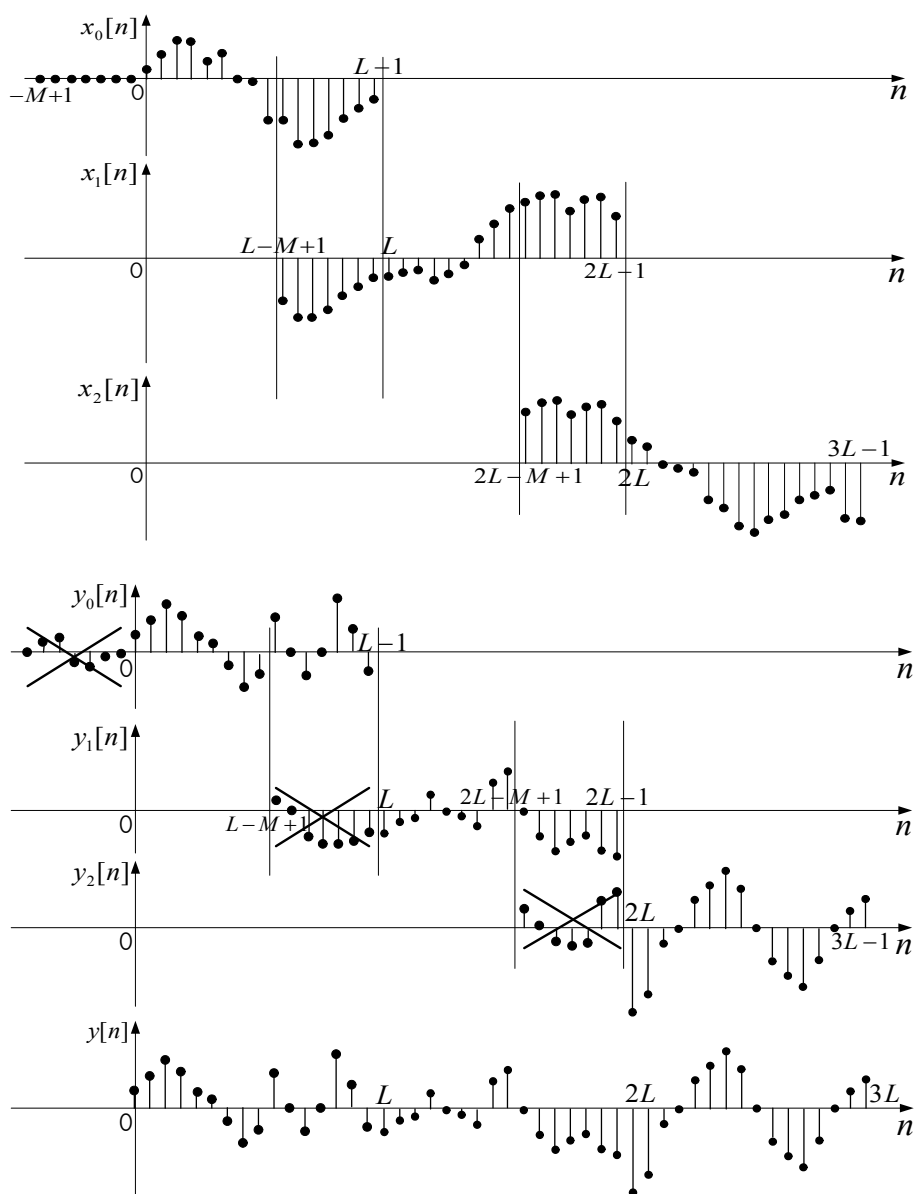
$$y_i[n] = x_i[n] \otimes h_a[n], \quad iL-M+1 \leq n \leq iL+L-1 \quad (\text{C11.17})$$

4. 각각의 원형 컨벌루션 수행 결과에서 앞의 $M-1$ 샘플을 버린 뒤 더하여 출력으로 낸다.

$$\bar{y}_i[n] = \begin{cases} 0, & iL-M+1 \leq n \leq iL-1 \\ y_i[n], & iL \leq n \leq iL+L-1 \end{cases} \quad (\text{C11.18})$$

$$y[n] = x[n] * h[n] = \sum_{i=0}^{\infty} \bar{y}_i[n] \quad (\text{C11.19})$$

위에서 $y_i[n]$ 을 만드는 입력을 $x[n]$, $iL - M + 1 \leq n \leq iL - 1$ 과 $x[n]$, $iL \leq n \leq (i+1)L - 1$ 의 둘로 나누어보면, $iL - M + 1 \leq n \leq iL - 1$ 에서는 그 구간의 입력, 즉 전자에 의한 응답만이 나오므로 불완전하다. $iL \leq n \leq (i+1)L - 1$ 은 전자의 입력이 들어오기 시작한 때로부터 임펄스 응답의 지속 시간만큼 지난 이후이므로 전자의 영향과 현 구간의 입력, 즉 후자의 영향이 모두 반영된 응답이 만들어지기 때문에 정상적인 출력이다. $y_{i+1}[n]$ 을 만들어내는 입력의 앞부분은 $y_i[n]$ 을 만들어내는 입력과 임펄스 응답의 지속 시간만큼 겹쳐 있는데, 이 구간의 컨벌루션 결과 값은 조금 전에 설명한 것처럼 불완전할 뿐만 아니라 어차피 $y_i[n]$ 에서 올바르게 계산된 결과가 이미 있기 때문에 불필요하다. 따라서 각 $y_i[n]$ 의 앞부분 $M - 1$ 샘플을 버리고 더하여 출력으로 내는 동작이 바로 식 (C11.19)이다.



[그림 C11-6] 중복-저장 방법에 의한 긴 데이터의 컨벌루션 계산

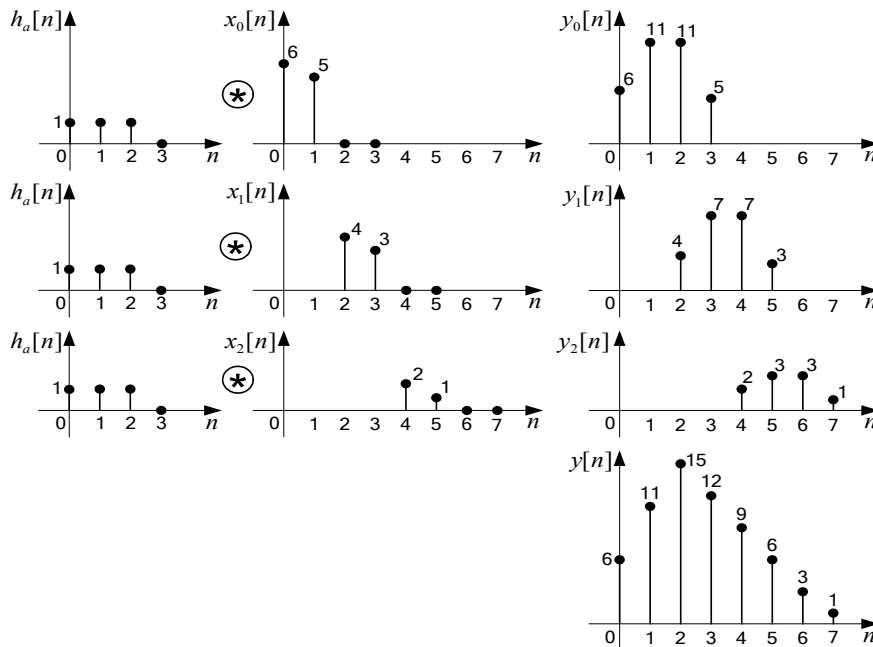
이러한 두 방법은 시스템의 특성에 의해 입력이 발생 순간만이 아니라 이후 상당 기간 출력에 영향을 미치는 것을 어떻게 반영해서 올바른 출력을 얻을 것인가에 대한 교묘한 기술적 해결 방안이다. 이렇게 하면 다음에 배울 FFT를 이용하여 원형 컨벌루션을 실시간 처리가 가능하게 수행하는 프로그램을 짜서 컴퓨터를 이용하여 디지털 신호 처리를 할 수 있게 된다.

■ 예제 C11-3 : 중복-더하기 방식에 의한 선형 컨벌루션 계산

(책)[예제 11-6]의 컨벌루션을 중복-더하기 방식으로 수행하라.

<풀이>

[그림 C11-7]에 나타낸 것처럼 입력 $x[n]$ 을 2개씩 잘라서, 즉 $L=2$ 로 하여 원형 컨벌루션을 수행하면, 임펄스 응답의 길이 $M=3$ 이므로 $N=2+3-1=4$ 가 된다. 따라서 각 $x_i[n]$ 에는 2개, $h_a[n]$ 에는 1개의 영 채우기를 해야 한다. 그런 다음 $y_i[n] = x_i[n] \otimes h_a[n]$ 을 계산하여 그 결과를 모두 더하면 된다. 이러한 계산 과정을 [그림 C11-7]에 나타내었다.



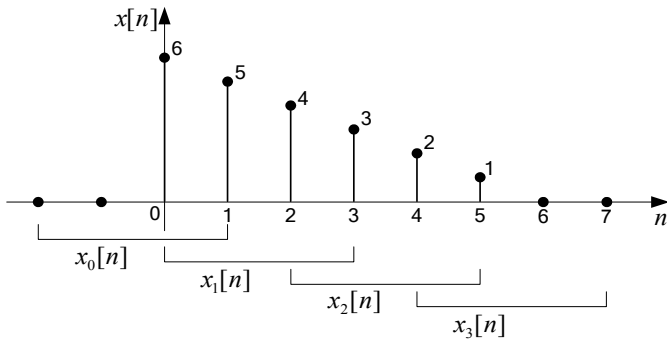
[그림 C11-7] [예제 C11-3]의 중복-더하기 방법 계산

■ 예제 C11-4 : 중복-저장 방식에 의한 선형 컨벌루션 계산

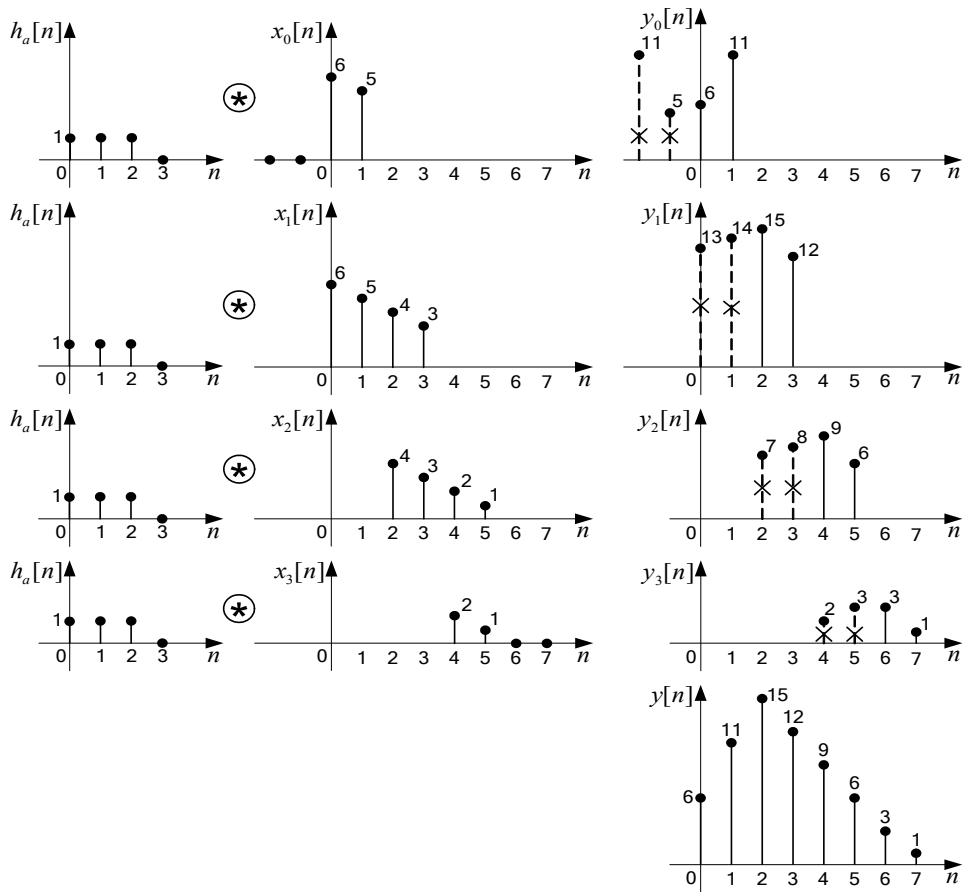
(책)[예제 11-6]의 컨벌루션을 중복-저장 방식으로 수행하라.

<풀이>

[그림 C11-8]과 같이 입력 $x[n]$ 을 4개씩 자르되 2개씩 겹치도록 잘라보자. 원형 컨벌루션을 수행하려면 $L=2$, 길이 $M=3$ 이므로 $N=2+3-1=4$ 가 된다. 처음 입력 블록에는 겹치는 입력이 없으므로 2개의 0을 앞부분에 채워 넣어야 한다. 그런 다음 $y_i[n] = x_i[n] \otimes h_a[n]$ 을 계산하여 각 블록마다 입력이 겹치는 부분에 해당되는 앞의 2 샘플은 버리고 모두 더하면 된다. 이러한 계산 과정은 [그림 C11-9]에 나타냈다.



[그림 C11-8] [예제 C11-4]의 신호 블록 분할



[그림 C11-9] [예제 C11-4]의 중복-저장 방법 계산

11.4 고속 푸리에 변환 (FFT)

11.4.1 DFT의 계산

실제로 수행해야 할 DFT의 계산량은 DFT의 대칭성을 이용하여 줄일 수 있다. 실수 신호의 경우 $X[k] = X^*[-k]$ 인데, 주기성에 의해 $X[-k] = X[N-k]$ 이므로, 실수 신호의 DFT는 $X[k] = X^*[N-k]$ 를 만족한다. 따라서 $k = N/2$ 에 대해서도 (책)[표 11-1]의 대칭성을 그대로 만족한다. 그러므로 $X[k]$, $0 \leq k \leq N/2$ 만 계산하면 전체 $X[k]$ 를 구성할 수 있다.

그런데 이와 같이 $N/2 + 1$ 개의 $X[k]$ 를 계산할 때, $X[k]$ 는 일반적으로 (두 개의 실수로 이루어진) 복소수이지만 다음의 두 값은 실수이므로 결국 DFT는 N 개의 실수 값으로 N 개의 실수 값을 계산하는 연산이 된다.

$$X[0] = \sum_{n=0}^{N-1} x[n], \quad X\left[\frac{N}{2}\right] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}\frac{N}{2}n} = \sum_{n=0}^{N-1} x[n](-1)^n \quad (\text{C11.20})$$

만약 실수 신호가 $x[n] = \pm x[N-n]$, $0 \leq n \leq N-1$ 이면, 즉 $n = N/2$ 에 대해 대칭성을 가지게 되면, DFT의 정의식에 의해 다음이 성립한다.

$$\begin{aligned} X[k] &= \pm \sum_{n=0}^{N-1} x[N-n] W_n^{kn} = \pm \sum_{m=1}^N x[m] W_n^{k(N-m)} \\ &= \pm \left(\sum_{n=0}^{N-1} x[n] W_N^{kn} \right)^* = \pm X^*[k] \end{aligned} \quad (\text{C11.21})$$

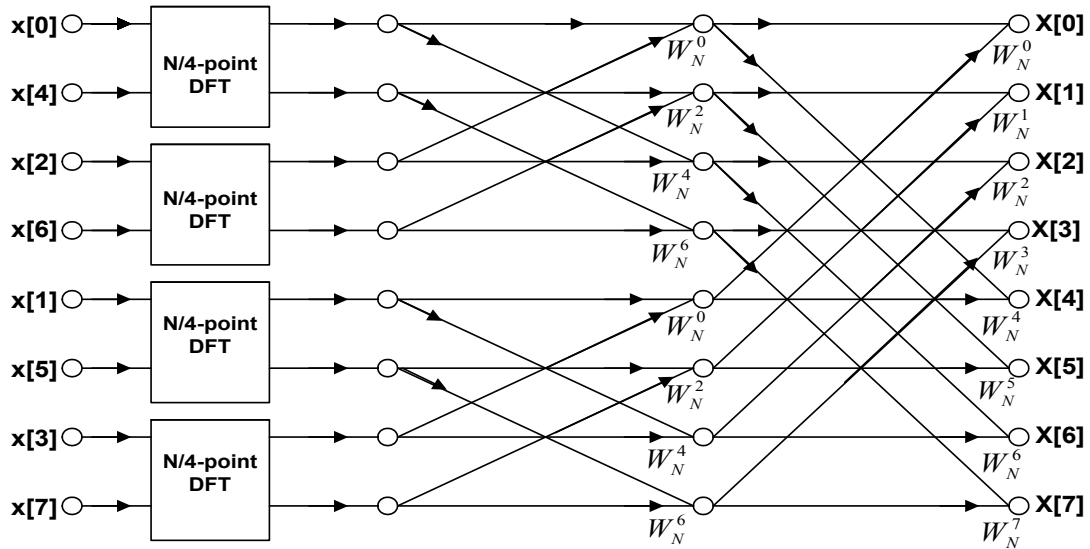
식 (C11.21)의 세 번째 등식은 $x[N] = x[0]$, $W_N^{kN} = 1$, $W_N^{-kN} = W_N^0$ 임을 이용하여 얻은 것이다. 식 (C11.21)은 다음과 같이 $x[n]$ 이 $n = N/2$ 에 대해 우함수 대칭이라면 $X[k] = X^*[k]$ 가 되므로 $\text{Im}\{X[k]\} = 0$ 이고, 기함수 대칭이라면 $X[k] = -X^*[k]$ 이므로 $\text{Re}\{X[k]\} = 0$ 이 됨을 의미한다.

$$\begin{cases} \text{Im}\{X[k]\} = 0, & x[n] = x[N-n], \quad x[n] \text{은 실수} \\ \text{Re}\{X[k]\} = 0, & x[n] = -x[N-n], \quad x[n] \text{은 실수} \end{cases} \quad (\text{C11.22})$$

따라서 샘플 스펙트럼 $X[k]$ 의 실수부 또는 허수부만 계산하면 되므로 실제 계산량을 더욱 줄일 수 있다.

11.4.2 밑수 2 시분할 FFT

(책)[그림 11-14]의 두 개의 $N/2$ 점 DFT를 4개의 $N/4$ 점 DFT로 쪼개면 [그림 C11-10]과 같이 된다.



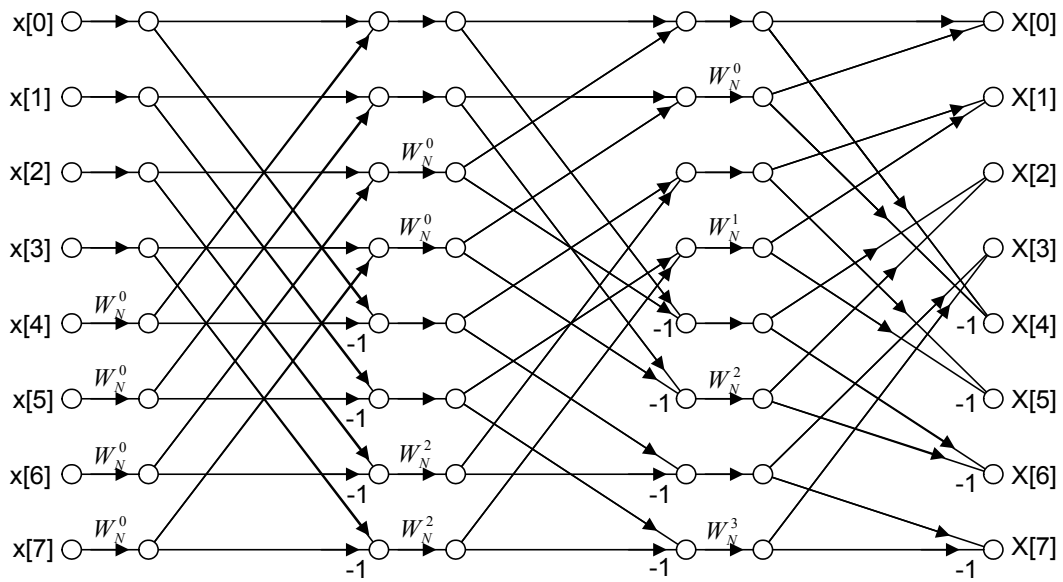
[그림 C11-10] N 점 DFT를 4개의 $N/4$ 점 DFT로 분할 계산($N=8$)

시분할 FFT의 변형

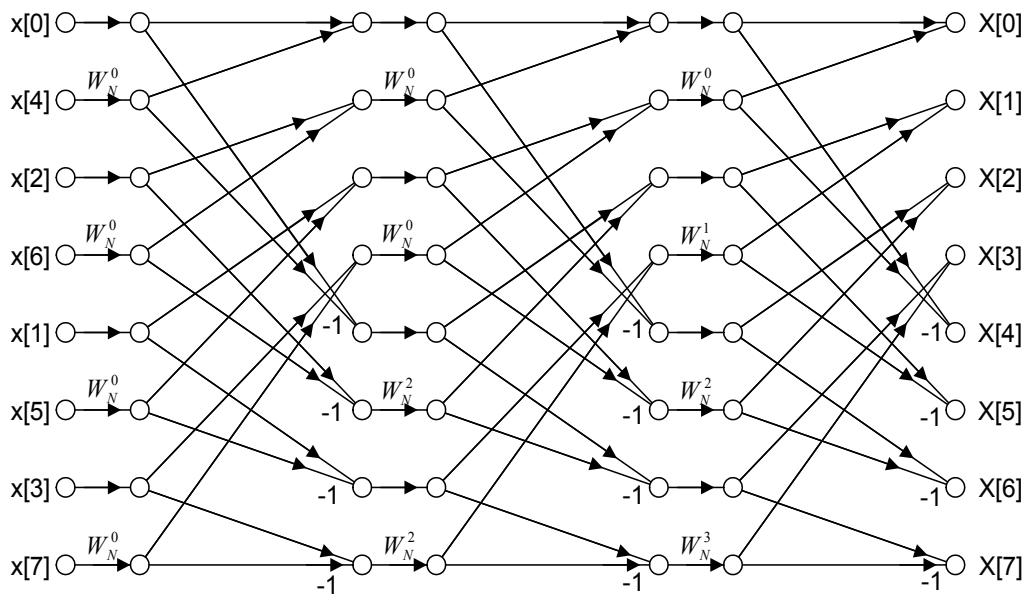
신호 흐름 선도의 중요한 특성은 마디의 연결 관계만 바뀌지 않으면 그림의 형태가 바뀌어도 동일한 대상을 표현한 것이라는 점이다. 그러므로 FFT 선도에서 마디의 연결만 달라지지 않으면 그림에 나타난 마디의 순서가 바뀌어도 계산 결과는 같게 된다.

따라서 밑수 2 시분할 FFT를 몇 가지 형태로 변형할 수 있다. 우선 (책)[그림 11-15]의 입력 역 비트순 형에서 $x[4] \leftrightarrow x[1]$, $x[6] \leftrightarrow x[3]$ 의 위치를 교환하면 (책)[그림 11-20]과 같이 입력 $x[n]$ 은 정상 순서, 출력 $X[k]$ 가 역 비트순인 형태로 바꿀 수 있다.

필요에 따라서는 [그림 C11-11]과 같이 입력 $x[n]$ 과 출력 $X[k]$ 모두 정상 순서인 형태로 바꿀 수도 있다. 또 다른 변형으로는 [그림 C11-12]와 같이 각 단계의 기하 구조가 동일한 형태로 만드는 경우를 들 수 있다. 앞에서 살펴본 세 가지 FFT 선도는 각 단계의 기하 구조가 달라서 데이터의 순차적인 액세스^{access}가 안 되지만, 이 경우는 각 단계의 기하 구조를 같게 하여 데이터의 순차적 액세스를 가능하게 함으로써 프로그래밍이나 하드웨어 구현에 매우 유리하다. 왜냐하면 하나의 단계에 대한 프로그램이나 하드웨어를 변형 없이 그대로 반복하여 활용할 수 있기 때문이다.



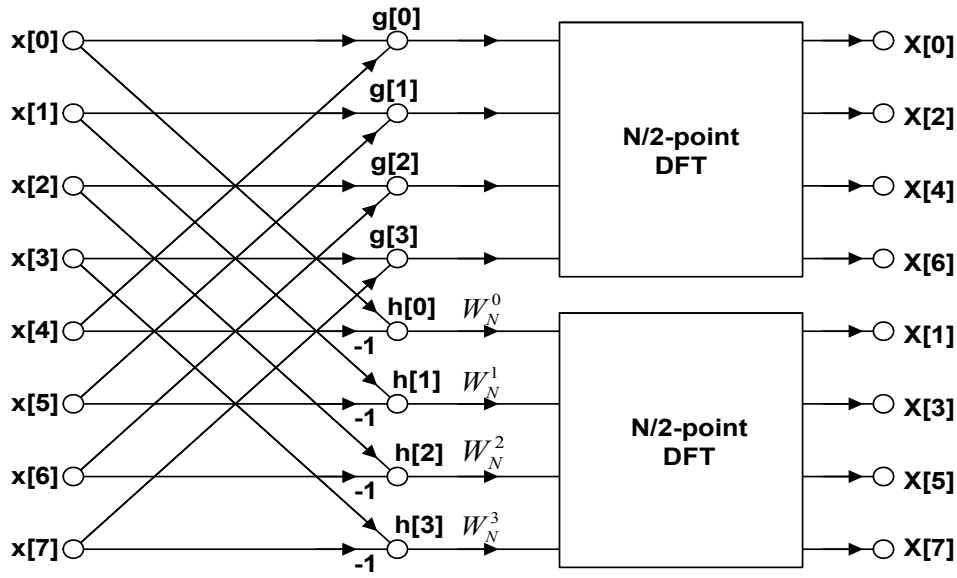
[그림 C11-11] 입력과 출력이 모두 정상순인 시분할 FFT 선도 (N=8)



[그림 C11-12] 각 단계의 기하 구조가 같은 시분할 FFT 선도 (N=8)

11.4.5 밑수 2 주파수 분할 FFT

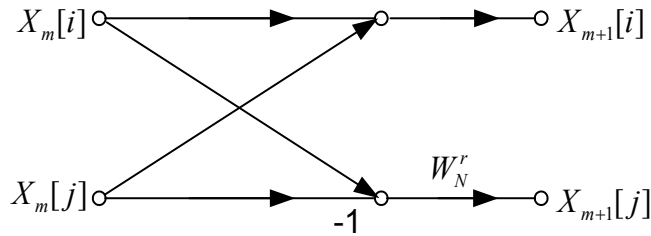
책에서 설명한 대로 주파수 분할에 의해서 N 점 DFT를 2개의 $N/2$ 점 DFT로 쪼갠 것을 그림으로 나타낸 것이 [그림 C11-13]이다.



[그림 C11-13] 주파수 분할에 의한 두 개의 $N/2$ 점 DFT로 분할($N=8$)

같은 방법을 반복해서 적용하여 2점 DFT가 될 때까지 분할할 수 있으며, 이때 기본 버터플라이는 [그림 C11-14]와 같고 이의 입출력 관계는 다음과 같다.

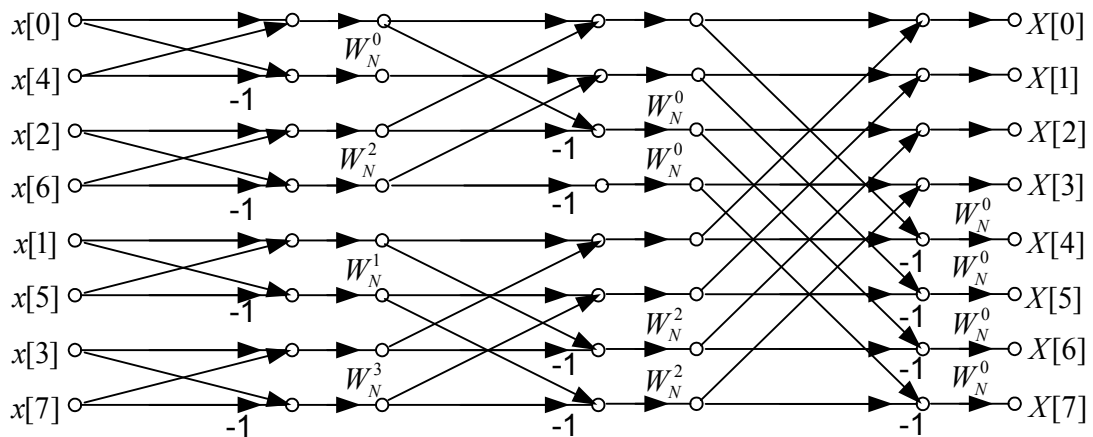
$$\begin{cases} X_{m+1}[i] = X_m[i] + X_m[j] \\ X_{m+1}[j] = (X_m[i] - X_m[j]) W_N^r \end{cases} \quad (C11.23)$$



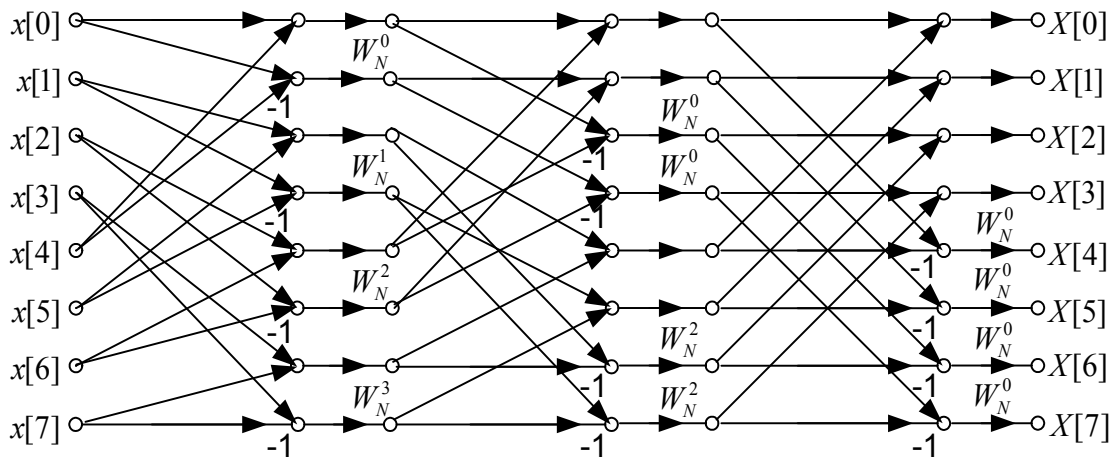
[그림 C11-14] 주파수 분할 FFT 버터플라이

주파수 분할 FFT의 변형

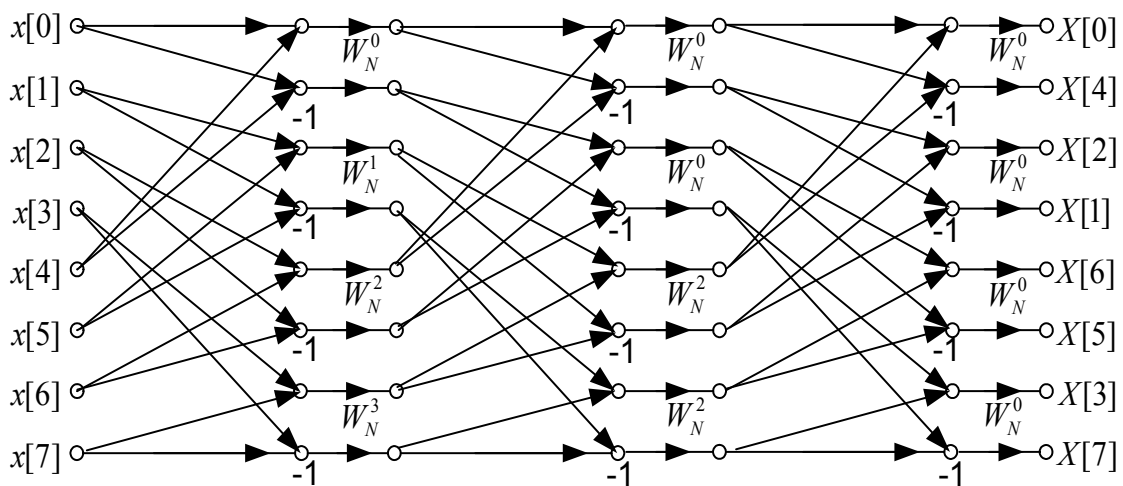
주파수 분할 FFT의 경우에도 시분할 FFT와 마찬가지로 여러 가지 변형이 가능하다. (책) [그림 11-21]의 출력 역 비트순 외에도 [그림 C11-15]의 입력 역 비트순, [그림 C11-16]과 같이 입출력이 모두 정상순인 FFT 선도와 각 단계별 기하 구조가 같은 [그림 C11-17]의 FFT 선도로 변형할 수 있다.



[그림 C11-15] 입력 역 비트순 주파수 분할 FFT 선도($N=8$)



[그림 C11-16] 입출력 정상순 주파수 분할 FFT ($N=8$)



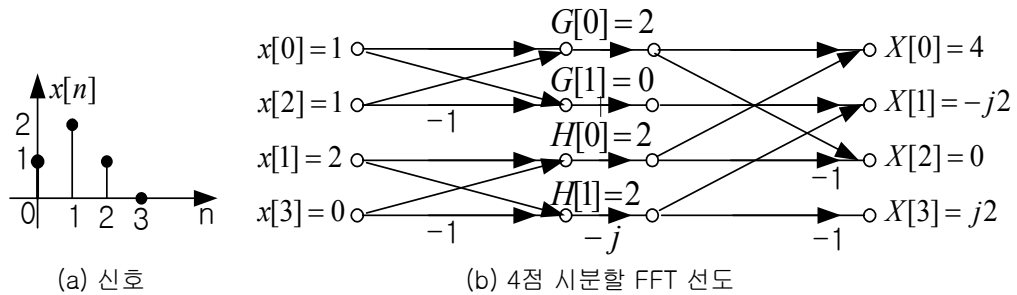
[그림 C11-17] 동일 기하 구조 주파수 분할 FFT 선도 ($N=8$)

■ 예제 C11-5 : 4점 밑수 2 시분할 FFT 계산

[그림 C11-18(a)]와 같은 신호 $x[n]$ 에 대해 4점 밑수 2 시분할 FFT 선도를 그리고 이를 이용하여 샘플 스펙트럼을 구하라.

<풀이>

주어진 신호에 대해 4점 시분할 FFT 선도를 그리면 [그림 C11-18(b)]와 같고, 이로부터 샘플 스펙트럼을 계산하면 다음과 같다.



[그림 C11-18] [예제 C11-5]의 신호 및 FFT 선도

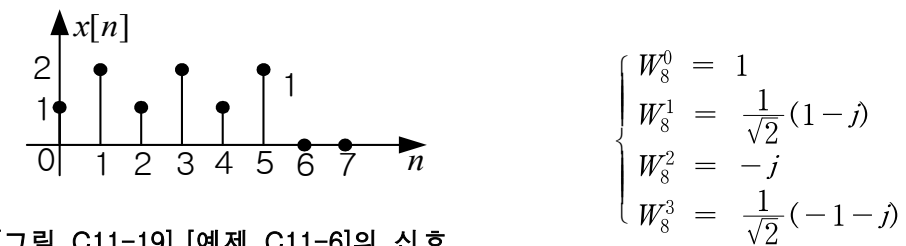
$$\begin{aligned}
 G[0] &= x[0] + x[2] = 2 & \Rightarrow & X[0] = G[0] + H[0] = 4 \\
 G[1] &= x[0] - x[2] = 0 & & X[1] = G[1] + (-j) \cdot H[1] = -j2 \\
 H[0] &= x[1] + x[3] = 2 & & X[2] = G[0] - H[0] = 0 \\
 H[1] &= x[1] - x[3] = 2 & & X[3] = G[1] - (-j) \cdot H[1] = j2
 \end{aligned}$$

■ 예제 C11-6 : 8점 밑수 2 시분할 FFT 계산

[그림 C11-19]와 같은 신호 $x[n]$ 에 대해 8점 밑수 2 시분할 FFT 선도를 그리고 이를 이용하여 샘플 스펙트럼을 구하라.

<풀이>

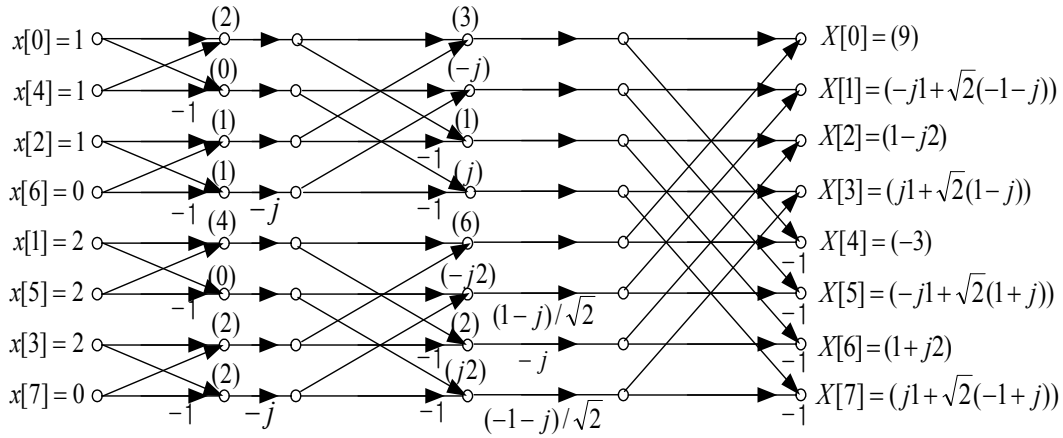
8점 FFT를 수행하기 위해 필요한 회전 인자는 W_8^m , $m = 0, 1, 2, 3$ 이므로 이를 구하면 다음과 같다.



주어진 신호에 대해 8점 시분할 FFT 선도를 그리면 [그림 C11-20]과 같고, 이로부터 샘플 스펙트럼을 계산하면 다음과 같다.

$$\begin{array}{ll}
B_1[0] = x[0] + x[4] = 2 & \Rightarrow \quad G[0] = B_1[0] + B_2[0] = 3 \\
B_1[1] = x[0] - x[4] = 0 & G[1] = B_1[1] + (-j)B_2[1] = -j \\
B_2[0] = x[2] + x[6] = 1 & G[2] = B_1[0] - B_2[0] = 1 \\
B_2[1] = x[2] - x[6] = 1 & G[3] = B_1[1] - (-j)B_2[1] = j \\
B_3[0] = x[1] + x[5] = 4 & H[0] = B_3[0] + B_4[0] = 6 \\
B_3[1] = x[1] - x[5] = 0 & H[1] = B_3[1] + (-j)B_4[1] = -j2 \\
B_4[0] = x[3] + x[7] = 2 & H[2] = B_3[0] - B_4[0] = 2 \\
B_4[1] = x[3] - x[7] = 2 & H[3] = B_3[1] - (-j)B_4[1] = j2
\end{array}$$

$$\begin{aligned}
\Rightarrow \quad X[0] &= G[0] + H[0] = 9 \\
X[1] &= G[1] + H[1] = -j + \sqrt{2}(-1-j) = -\sqrt{2} - j(\sqrt{2}+1) \\
X[2] &= G[2] + H[2] = 1 - j2 \\
X[3] &= G[3] + H[3] = j + \sqrt{2}(1-j) = \sqrt{2} - j(\sqrt{2}-1) \\
X[4] &= G[0] + H[0] = -3 \\
X[5] &= G[1] - \left(\frac{1-j}{\sqrt{2}}\right)H[1] = -j + \sqrt{2}(1+j) = \sqrt{2} + j(\sqrt{2}-1) \\
X[6] &= G[2] - (-j)H[2] = 1 + j2 \\
X[7] &= G[3] - \left(\frac{-1-j}{\sqrt{2}}\right)H[3] = jj + \sqrt{2}(-1+j) = -\sqrt{2} + j(\sqrt{2}+1)
\end{aligned}$$



[그림 C11-20] [예제 C11-6]의 시분할 FFT 선도

11.4.6 역 FFT(IFFT)

역 FFT의 유도

$X[k]$ 를 짝수 항과 홀수 항으로 분할하면

$$\begin{aligned}
x[n] &= \sum_{k=0}^{N-1} \frac{X[k]}{N} W_N^{-kn} = \sum_{r=0}^{N/2-1} \frac{X[2r]}{N} W_N^{-2rk} + \sum_{r=0}^{N/2-1} \frac{X[2r+1]}{N} W_N^{-(2r+1)k} \quad (C11.24) \\
&= \sum_{r=0}^{N/2-1} \frac{X[2r]}{N} (W_N^2)^{-kr} + W_N^{-k} \sum_{r=0}^{N/2-1} \frac{X[2r+1]}{N} (W_N^2)^{-kr} \\
&= \sum_{r=0}^{N/2-1} \frac{X[2r]}{N} W_{\frac{N}{2}}^{-kr} + \sum_{r=0}^{N/2-1} \frac{X[2r+1]}{N} W_{\frac{N}{2}}^{-kr} \cdot W_N^{-k} \\
&= \sum_{r=0}^{N/2-1} C_0[k] + \sum_{r=0}^{N/2-1} W_N^{-k} C_1[k]
\end{aligned}$$

식 (C11.24)를 다시 짝수 항과 홀수 항으로 계속 분할해나가면 시분할 FFT의 유도 때와 마찬가지로 궁극적으로는 다음과 같은 2점 IDFT로 나뉜다.

$$U[k] = \sum_{i=0}^1 \frac{X[v(i)]}{N} W_2^{-ik}, \quad k=0, 1 \quad (C11.25)$$

여기서 $W_2^{-0k} = 1$, $W_2^{-1k} = e^{j\frac{2\pi}{2}k} = \begin{cases} 1, & k=0 \\ -1, & k=1 \end{cases}$ 이다.

따라서 식 (C11.25)를 다시 쓰면 다음과 같다.

$$\begin{cases} U[0] = \frac{X[v(0)]}{N} + \frac{X[v(1)]}{N} \\ U[1] = \frac{X[v(0)]}{N} - \frac{X[v(1)]}{N} \end{cases} \quad (C11.26)$$

11.4.7 일반화된 FFT

$N \neq 2^q$ 일 경우에도 N 을 소수의 곱, 즉 소인수 분할에 의해 쪼개어 FFT 구조로 만들 수 있다. 이와 같이 일반화된 FFT를 어떻게 유도할 수 있는지 살펴보기로 하자.

$N = p_1 \cdot q_1$ 이라면 입력 수열을 p_1 개의 q_1 샘플 수열로 분할하여 N 점 DFT를 p_1 개의 q_1 점 DFT로 바꿀 수 있다. 그러면 N 점 DFT는 p_1 개의 q_1 점 DFT를 계산하여 W_N^{lk} 곱해서 더하면 된다. 따라서 이 경우에는 $p_1 q_1^2 + N(p_1 - 1)$ 의 곱셈이 필요하다.

$$\begin{aligned}
X[k] &= \sum_{n=0}^{N-1} x[n] W_N^{kn} \quad (C11.27) \\
&= \sum_{r=0}^{q_1-1} x[p_1 r] W_N^{p_1 r k} + \sum_{r=0}^{q_1-1} x[p_1 r + 1] W_N^k W_N^{p_1 r k} + \dots \\
&\quad + \sum_{r=0}^{q_1-1} x[p_1 r + p_1 - 1] W_N^{(p_1-1)k} W_N^{p_1 r k} \\
&= \sum_{l=0}^{p_1-1} W_N^{lk} \sum_{r=0}^{q_1-1} x[p_1 r + l] W_N^{p_1 r k} = \sum_{l=0}^{p_1-1} W_N^{lk} G_l[k]
\end{aligned}$$

여기서

$$G_l[k] \equiv \sum_{r=0}^{q_1-1} x[p_1 r + l] W_N^{p_1 r k} = \sum_{r=0}^{q_1-1} x[p_1 r + l] W_{q_1}^{r k} \quad (C11.28)$$

식 (C11.28)은 $W_N^{p_1 r k} = e^{-j \frac{2\pi}{N} p_1 r k} = W_{q_1}^{r k}$ 임을 이용한 것이다.

예를 들어 $N=12=3 \times 4$ 으로 나눈다면, 다음과 같이 3개의 4점 DFT로 분할이 될 것이다.

$$\begin{array}{cccc} x[0] & x[3] & x[6] & x[9] \\ x[1] & x[4] & x[7] & x[10] \\ x[2] & x[5] & x[8] & x[11] \end{array} \quad (C11.29)$$

위와 같은 분할에서 같은 그룹의 신호 샘플은 $x[p_1 r + l]$, $r=0, \dots, q_1-1$ 이다. 이때 그룹은 $l=0, \dots, p_1-1$ 에 따라 분류된다.

다시 $q_1 = p_2 q_2$ 로 두어 q_1 점 DFT를 p_2 개의 q_2 점 DFT로 분할하게 되면, p_2 개의 q_2 점 DFT는 $p_2 q_2^2 + q_1(p_2 - 1)$ 의 곱셈이 필요하다. 따라서 N 점 DFT는 $N(p_1 - 1) + N(p_2 - 1) + p_1 p_2 q_2^2$ 의 곱셈이 필요하게 된다. 끝까지 분할하면 $N = p_1 p_2 \dots p_\nu$ 이 되므로 p_ν 점 DFT로 시작하는 ν 단계를 갖는 FFT로 변환된다. 이때 필요한 곱셈의 양은 다음과 같다.

$$N(p_1 - 1) + N(p_2 - 1) + \dots + N(p_{\nu-1} - 1) + p_1 p_2 \dots p_{\nu-1} p_\nu^2 = N(p_1 + \dots + p_\nu - (\nu - 1)) \quad (C11.30)$$

N 을 분할할 때 더 이상 나누어지지 않도록 소인수 분해하는 것이 계산량이 더 적게 된다. 예를 들어 $N = p_1 p_2 \dots p_\nu$ 로 분할했을 때 $p_i = r_i s_i$ 라면, $N = p_1 \dots p_{i-1} r_i s_i p_{i+1} \dots p_\nu$ 이므로 더 분할된 FFT의 곱셈 연산량은 $N(p_1 + \dots + p_{i-1} + r_i + s_i + p_{i+1} + \dots + p_\nu - (\nu - 1))$ 이 된다. 그런데 $p_i = r_i s_i > r_i + s_i$ 이므로 다음과 같이 덜 쪼갠 때의 곱셈 연산량이 더 크다.

$$N(p_1 + \dots + p_i + \dots + p_\nu - (\nu - 1)) > N(p_1 + \dots + r_i + s_i + \dots + p_\nu - (\nu - 1)) \quad (C11.31)$$

이와 같이 소인수 분해에 의해 DFT를 분할할 때 어떤 규칙성을 가지고 식 (C11.29)와 같이 신호 샘플들이 그룹화되는지 살펴보자.

N 을 $N = p_1 p_2 \dots p_\nu$ 로 소인수 분해하면 시간 변수 n 은 다음과 같이 나타낼 수 있다.

$$\begin{aligned} n &= p_1 q_1 + r_1 = p_1 (p_2 q_2 + r_2) + r_1 = p_1 (p_2 (p_3 q_3 + r_3) + r_2) + r_1 = \dots \\ &= (p_1 p_2 \dots p_{\nu-1}) q_{\nu-1} + (p_1 p_2 \dots p_{\nu-2}) r_{\nu-1} + \dots + (p_1) r_2 + r_1 \\ &= (p_1 p_2 \dots p_{\nu-1}) n_1 + (p_1 p_2 \dots p_{\nu-2}) n_2 + \dots + (p_1) n_{\nu-1} + n_\nu \end{aligned} \quad (C11.32)$$

따라서 시간 변수 n 과 주파수 변수 k 는 다음과 같이 나타낼 수 있다.

$$\begin{cases} n = (p_1 p_2 \cdots p_{\nu-1})n_1 + (p_1 p_2 \cdots p_{\nu-2})n_2 + \cdots + (p_1)n_{\nu-1} + n_{\nu} \\ k = (p_{\nu} p_{\nu-1} \cdots p_2)n_{\nu} + (p_{\nu} p_{\nu-1} \cdots p_3)n_{\nu-1} + \cdots + (p_{\nu})n_2 + n_1 \end{cases} \quad (C11.33)$$

여기서 $0 \leq n_1 \leq p_{\nu}-1$, $0 \leq n_2 \leq p_{\nu-1}-1$, \cdots , $0 \leq n_{\nu} \leq p_1-1$ 로 각 자리^{digit}의 값이다.

식 (C11.33)은 샘플 신호 $x[n]$ 과 샘플 스펙트럼 $X[k]$ 가 밀수 2 FFT때와 비슷하게 역 자리순으로 대응됨을 말해준다.

예를 들어 $N=18$ 의 경우, $N=3 \cdot 3 \cdot 2 = p_1 p_2 p_3$ 로 소인수 분할된다. 따라서 n 과 k 는 다음과 같이 나타낼 수 있다.

$$\begin{cases} n = p_1 p_2 n_1 + p_1 n_2 + n_3 \\ k = p_3 p_2 n_3 + p_3 n_2 + n_1 \end{cases} \quad (C11.34)$$

이로부터 $n=16$ 에 대해 따져보면 $n=16=9 \cdot 1+3 \cdot 2+1$ 이므로 $n_1=1$, $n_2=2$, $n_3=1$ 이다. 따라서 $k=6 \cdot 1+2 \cdot 2+1=11$ 에 대응된다. 또한 $n=11$ 은 $n=11=9 \cdot 1+3 \cdot 0+2$ 이므로, $n_1=1$, $n_2=0$, $n_3=2$ 가 되어 $k=6 \cdot 2+2 \cdot 0+1=13$ 에 대응된다.

이상의 결과는 밀수 2 FFT를 일반화한 것이므로 당연히 밀수 2 FFT에서도 만족된다. 예를 들어 $N=4=2 \cdot 2=p_1 \cdot p_2$ 의 경우 $n=2n_1+n_2$, $\leftrightarrow k=n_1+2n_2$, $0 \leq n_1, n_2 \leq 1$ 이므로 다음과 같이 변환 관계를 그림으로 나타낼 수 있다.

$n_2 \backslash n_1$	0	1
0	0	2
1	1	3

$n_1 \backslash n_2$	0	1
0	0	1
1	2	3

n_1	n_2	n	n_2	n_1	k
0	0	0	0	0	0
1	0	2	0	1	1
0	1	1	1	0	2
1	1	3	1	1	3

[그림 C11-21] 소인수 분해 FFT의 인덱스 대응 관계

그림에서 보면, 식 (C11.33)의 대응 관계를 적용하여 얻은 결과임에도 입력은 역 비트순 출력은 정상순으로 대응되고 있다. 따라서 식 (C11.33)이 더 일반적인 관계임을 알 수 있다.

■ 예제 C11-7 : 16점 밀수 4 FFT의 DFT 분할 관계

$N=16$ 일 때 밀수 4 FFT의 시간 신호 및 주파수 스펙트럼 분할 관계를 구하라.

<풀이>

$N=16=4 \cdot 4=p_1 \cdot p_2$ 이므로 $n=4n_1+n_2$, $\leftrightarrow k=n_1+4n_2$, $0 \leq n_1, n_2 \leq 3$ 이 된다. 따라서 샘플 신호와 샘플 스펙트럼의 분할 대응 관계는 [그림 C11-22]와 같이 된다.

그림에서 $\{x[0], x[4], x[8], x[12]\}$, $\{x[1], x[5], x[9], x[13]\}$, $\{x[2], x[6], x[10], x[14]\}$, $\{x[3], x[7], x[11], x[15]\}$ 의 4개의 4점 DFT로 분할되어 정상순인 스펙트럼에 대응된다.

$n_2 \backslash n_1$	0	1	2	3
0	0	4	8	12
1	1	5	9	13
2	2	6	10	14
3	3	7	11	15

$n_1 \backslash n_2$	0	1	2	3
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15

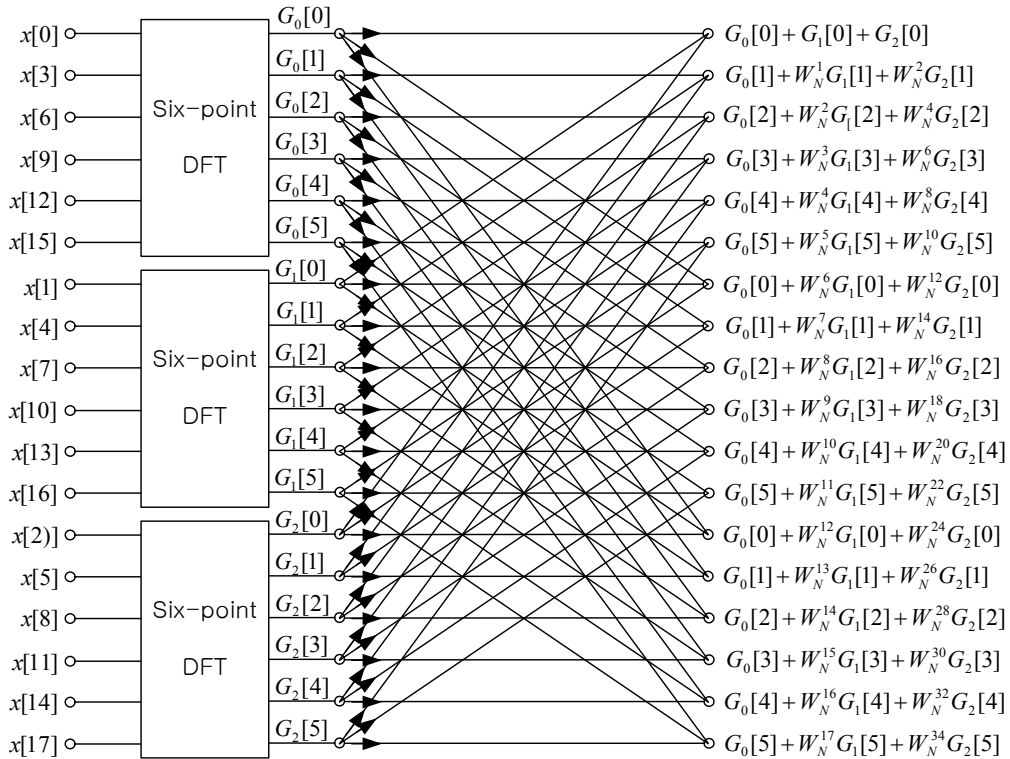
[그림 C11-22] [예제 C11-7]의 인덱스 대응 관계

이상과 같은 소인수 분해 FFT 원리를 이용하여 $N=18$ 일 경우의 FFT 선도를 구해보자. 우선 $N = 3 \cdot 6$, 즉 3개의 6점 DFT로 분할하면 다음과 같다.

$$\begin{array}{cccccc} x[0] & x[3] & x[6] & x[9] & x[12] & x[15] \\ x[1] & x[4] & x[7] & x[10] & x[13] & x[16] \\ x[2] & x[5] & x[8] & x[11] & x[14] & x[17] \end{array} \quad (C11.35)$$

이를 수식으로 나타내면 다음과 같고, 이의 FFT 선도는 [그림 C11-23]에 나타내었다.

$$X[k] = \sum_{l=0}^2 W_{18}^{lk} \sum_{r=0}^5 x[3r+l] W_{18}^{3rk} = G_0[k] + W_{18}^k G_1[k] + W_{18}^{2k} G_2[k] \quad (C11.36)$$



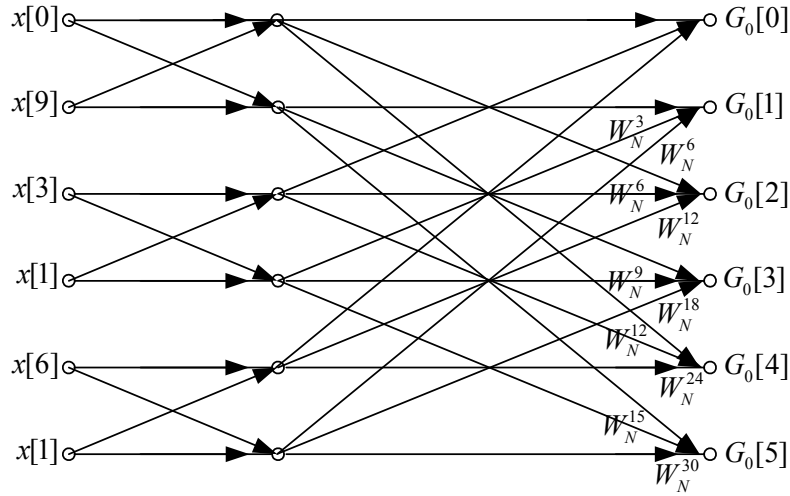
[그림 C11-23] 3개의 6점 DFT로 분할한 FFT 선도 ($N=18$)

[그림 C11-23]의 6점 DFT를 3개의 2점 DFT로 변환하면 다음과 같다.

$$\begin{array}{ccc} x[0] & x[9] & x[1] & x[10] & x[2] & x[11] \\ x[3] & x[12] & x[4] & x[13] & x[5] & x[14] \\ x[6] & x[15] & x[7] & x[16] & x[8] & x[17] \end{array} \quad (C11.37)$$

이를 수식으로 나타내면 다음과 같고, 첫 번째 블록에 대한 FFT 선도를 [그림 C11-24]에 나타내었다.

$$G_l[k] = \sum_{r=0}^5 x[3r+l] W_6^{rk} = \sum_{s=0}^2 W_6^{sk} \sum_{p=0}^1 x[9p+3s+l] W_6^{3pk} \quad (C11.38)$$



[그림 C11-24] 3개의 2점 DFT로 분할한 FFT 선도 ($N=6$)