2.5 연속시간 푸리에 변환

실제에서의 실질적인 신호들은 대부분이 주기신호가 아닌 비주기신호들이다. 따라서 비주기신호들의 주파수 성분을 찾는 방법이 필요하며, 연속시간 푸리에변환(CTFT, Continuous-time Fourier transform)을 사용한다.

비주기신호는 기본 주기가 무한대인 주기신호라고 가정할 수 있으며, 시간영역에서 주기를 무한대로 접근시키면, 주파수영역에서는 주파수 선 간격이 영으로 줄어들어 극한으로는 모든 선이 연결되는 연속스펙트럼이 된다.

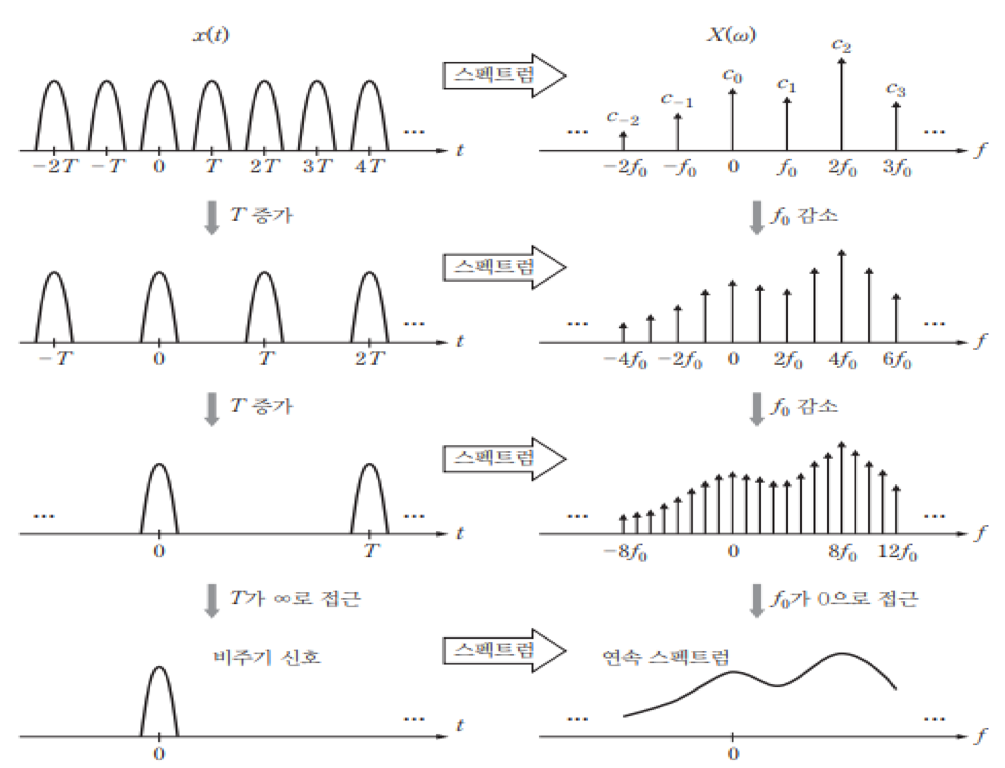


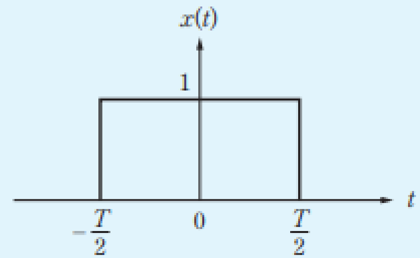
그림 2–6 연속시간 푸리에변환에서의 비주기신호와 주파수스펙트럼의 관계

$$X(\omega) = \int_{t=-\infty}^{\infty} x(t) e^{-j2\pi ft} dt$$

2.5 연속시간 푸리에 변환

예제 2-6 다음과 같은 구형파신호의 연속시간 푸리에변환을 구하고, 선스펙트럼을 나타내시오.

$$x(t)=\begin{cases}1, & -\frac{T}{2}\le t<\frac{T}{2}\\ 0, & \text{otherwise}\end{cases}$$



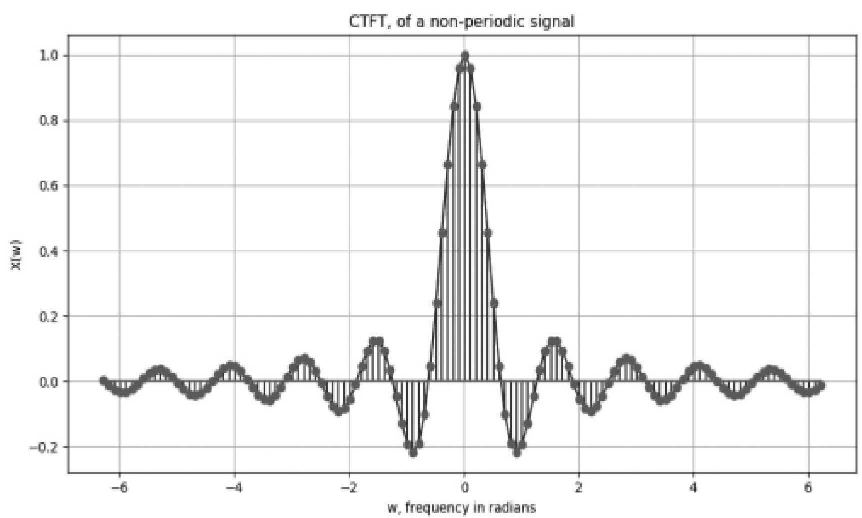
풀이 비주기 구형파신호를 연속시간 푸리에변환식에 대입하면 구형파신호의 연속시간 푸리에변환은 다음과 같은 싱크(sinc)신호를 나타낸다. 실제 스펙트럼은 그림에서의 포락선처럼 연속스펙트럼이 된다.

$$X(\omega)=\int_{t=-\infty}^{\infty}x(t)e^{-j\omega t}dt=\int_{t=-\frac{T}{2}}^{\frac{T}{2}}1\cdot e^{-j\omega t}dt$$

$$=\left[-\frac{1}{j\omega}e^{-j\omega t}\right]_{-\frac{T}{2}}^{\frac{T}{2}}=-\frac{1}{j\omega}\left(e^{-j\omega\frac{T}{2}}-e^{j\omega\frac{T}{2}}\right)$$

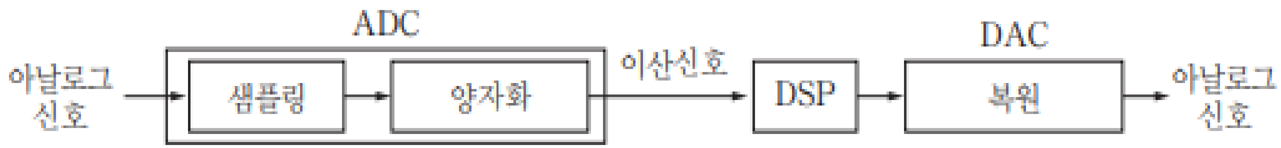
$$=-\frac{1}{j\omega}\left(-2j\sin\left(\frac{\omega T}{2}\right)\right)$$

$$=\frac{\sin\left(\frac{\omega T}{2}\right)}{\frac{\omega}{2}}=\frac{\sin(\pi fT)}{\pi f}=\frac{\sin(\pi f)}{\pi f}$$



2.6 샘플링과 복원

디지털통신 등에서 실제 아날로그신호는 샘플링(sampling)과 양자화(quantization) 동작을 거쳐서 이산신호로 전환된다. 그리고 이산신호는 디지털 신호처리기(digital signal processor)를 통해서 처리되고, 처리된 신호는 복원(reconstruction) 동작을 통해서 아날로그신호로 전환된다.

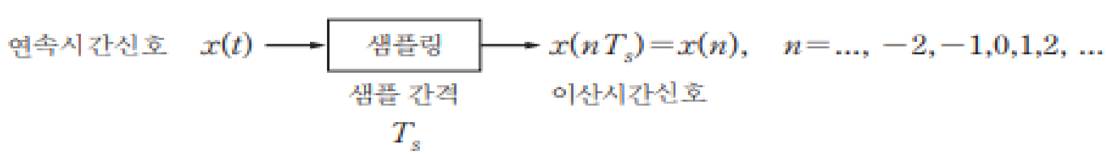


2.6.1 샘플링

신호 x(t)가 절대적으로 적분 가능한 연속시간신호일 때, 그것의 CTFT와 inverse CTFT는 다음 식과 같이 쌍으로 관계한다.

$$X(\omega)=\int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \qquad \Leftrightarrow \qquad x(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}d\Omega$$

연속시간신호 x(t)로부터 이산시간신호 x(n)을 얻는 과정을 도식적으로 나타내면 다음과 같다.



2.6 샘플링과 복원

X(Ω)가 이산시간신호 x(t)의 이산시간 푸리에변환(DTFT)이면, 그때 X(Ω)는 식 (2-30)과 같이 아날로그신호 x(t)의 푸리에변환, 즉 CTFT인 X(ω)를 크기-조정 (magnitude-scaled), 주파수-조정(frequency-scaled), 이동(translated)한 버전이라고 할 수 있다.

$$X(\Omega) = \frac{1}{T_s} \sum_{l=-\infty}^{\infty} X\left(\frac{\omega}{T_s} - \frac{2\pi}{T_s} l\right)$$

아날로그 라디안주파수 ω 와 디지털 라디안주파수 Ω 는 다음의 관계를 가지고 있다.

$$\Omega = 2\pi F = 2\pi f / f_s = \omega / f_s = \omega T_s$$

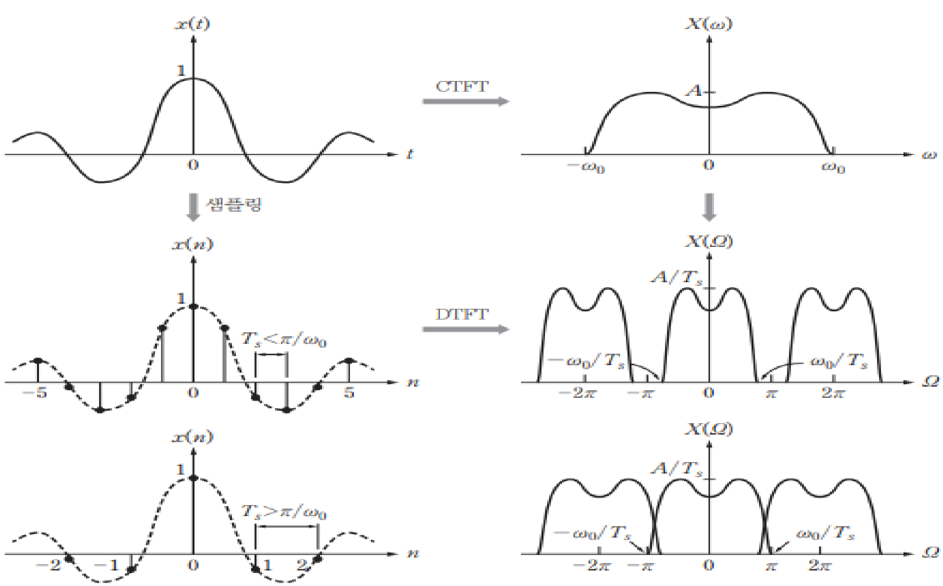


그림 2-7 시간영역에서의 샘플링에 대한 주파수영역에서의 해석

이산시간 푸리에변환 DTFT와 연속시간 푸리에변환 CTFT의 관계

여기서 ω_0 은 입력신호가 가지는 주파수대역폭

$X(\omega)$ 의 무한대 복제(replicas)들이 서로 중복(overlap) 되지 않으면 DTFT $X(\Omega)$ 로부터 CTFT $X(\omega)$ 를 복원시키는 것이 가능하다.

2.6 샘플링과 복원

<샘플링 정리>

대역폭이 ω_0인 한정대역신호 x(t)는 식 (2-33)과 같이 만약 샘플링주파수 $f_s = \frac{1}{T_s}$이 x(t)의 대역폭 ω_0의 두 배보다 더 크면 그것의 샘플 값들인 x(n)= x(nT_s)으로 복원될 수 있다.
즉"샘플링주파수는 입력신호의 대역폭의 두 배 이상이어야 한다."는 것이 샘플링 정리(Sampling theorem)이다.

$$f_s \geq 2\omega_0 \qquad (2\text{-}33)$$

입력신호의 주파수대역폭은 입력신호가 내포하고 있는 최대주파수 f_m으로 결정되므로 다음과 같이 나타낼 수도 있다.

$$f_s \geq 2f_m$$

여기서 fs =2ω_0일 때의 샘플링주파수를 나이키스트주파수(Nyquist frequency)라고 부른다.
따라서 샘플링 정리에 의하면, 연속시간신호 x(t)를 샘플링하여 얻은 이산시간신호 x(n)이 나타낼 수 있는 가장 높은 아날로그주파수는 $\frac{f_s}{2}$[Hz]이다.

예를 들어 어떤 연속시간신호 x(t)을 $f_s = 8kHz$의 샘플링주파수로 샘플링하여 이산시간신호 x(n)을 얻는다면, 이산시간신호 x(n)으로부터 측정할 수 있는 주파수의 최대 범위는 샘플링정리에 의해 $f_m = \frac{f_s}{2} = 4[kHz]$

2.6 샘플링과 복원

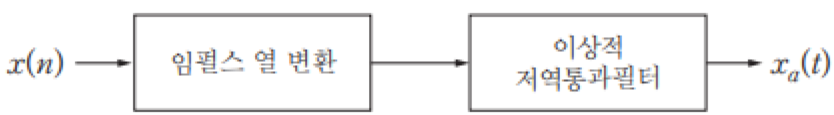
2.6.2 복원

샘플링 정리에 따르면 대역제한신호 x(t)를 그것의 나이키스트비율 이상으로 샘플링하면 샘플 x(n)으로부터 x(t)를 다시 복원(reconstruction)할 수 있다. 복원은 다음과 같이 두 단계로 행해진다.

- 먼저 샘플들을 식 (2-34)와 같이 가중치 임펄스 열로 변환한다.

$$\sum_{n=-\infty}^{\infty} x(n)\delta(t-nT_s) = \cdots + x(-1)\delta(t+T_s) + x(0)\delta(t) + x(1)\delta(t-T_s) + \cdots \quad (2\text{-}34)$$

- 다음으로, 임펄스 열을 다음과 같이 $[-\frac{f_S}{2}, \frac{f_S}{2}]$의 구간으로 대역이 제한된 이상적인 아날로그저역통과필터(sinc 함수)를 통해 필터처리한다.



위의 두 단계 과정은 보간공식을 사용해서 식 (2-35)와 같이 수학적으로 기술될 수 있다. 보간이 이루어진 신호 x(t)는 식 (2-35)와 같다.

$$x_a(t) = \sum_{n=-\infty}^{\infty} x(n)\mathrm{sinc}[f_s(t-nT_s)] \quad (2\text{-}35)$$

여기서 sinc(x)=$\frac{\sin(\pi x)}{\pi x}$은 보간함수(interpolation function)이다.

2.6 샘플링과 복원

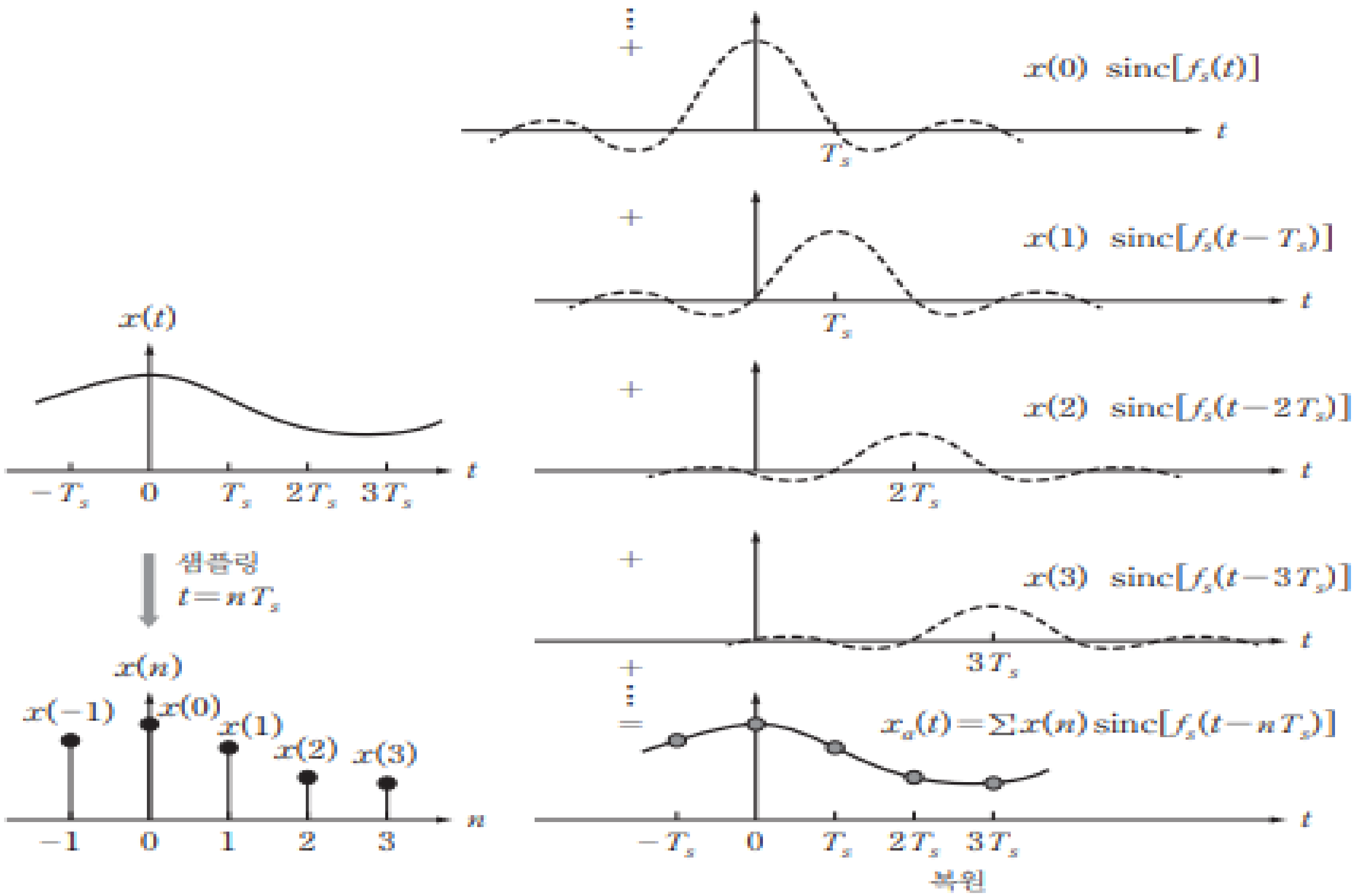


그림 2-8 대역제한 신호의 복원

2.6 샘플링과 복원

예제 2-7 연속시간신호 $x(t)$을 $f_s = 5,000$[samples/ sec]로 샘플링해서 다음 식과 같은 이산시간신호 $x(n)$을 얻었다. 함수를 사용하여 $x(t)$를 복원하시오.

$$x(n) = e^{-1000|nT_s|}, \quad -25 \le n \le 25$$

풀이 f_s=5,000[samples/ sec]이므로 샘플링주기는 $T_s = \frac{1}{f_s} = \frac{1}{5000} = 0.0002$[sec]이다. 그리고 신호는 $x(n) = e^{-1000|nT_s|}$, $-25 \le n \le 25$이므로, 시간으로는 −0.005초에서 0.005초까지 구간이며, 51개의 샘플로 구성된다. 그림 2-9의 상단 그림과 같다. 그리고 복원을 위해 싱크함수를 적용하는데 유한시간 구간 $-0.005 \le t \le 0.005$에 대해 무한대의 시간 구간을 사용할 수 없으므로 매우 작은 0.00005의 시간 구간으로 원래 샘플수보다 4배가 많은 샘플로 확장 적용한다.

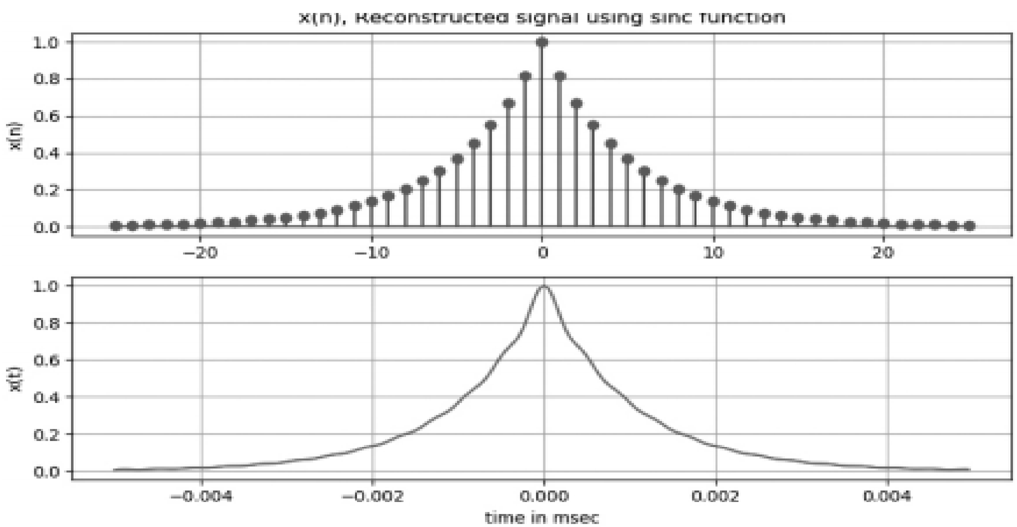


그림 2-9 예제 2-7의 복원된 신호 결과

실제적인 D/A 변환기

대역제한 신호를 복원하는 데 있어서, 실제적으로는 싱크함수를 사용하는 것이 아니라 다음과 같은 세 가지 저역통과필터로 대체한다.

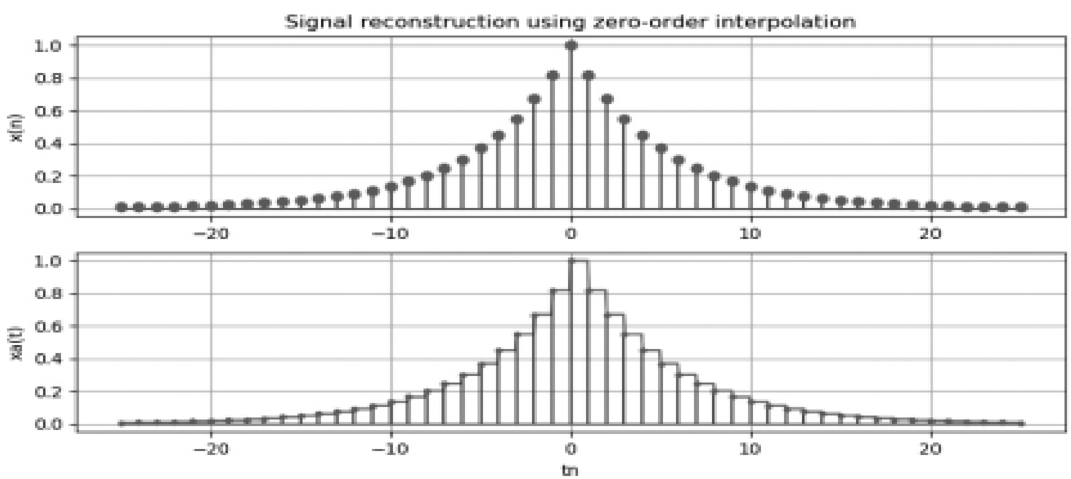
1. 영차유지보간(zero-order hold interpolation): 이 보간 방법은 현재 샘플에서 다음 샘플까지의 구간 동안을 현재의 샘플 값으로 일정하게 유지하게 하는 방법이다. 입출력 사이의 관계가 영차방정식이 되므로 영차유지보간 방법이라 부른다.

$$\hat{x}_a(t) = x(n), \quad nT_s \le n < (n+1)T_s$$

$$x(n) \longrightarrow \boxed{\text{ZOH}} \longrightarrow \hat{x}_a(t) \longrightarrow \boxed{\text{후처리필터}} \longrightarrow x_a(t)$$

예제 2-8 예제 2-7의 $x(n)$을 영차유지 보간법을 사용하여 복원하시오.

풀이 이산시간신호 $x(n)$은 $x(n) = e^{-1000|nT_s|}$, $-25 \le n \le 25$ 이다.

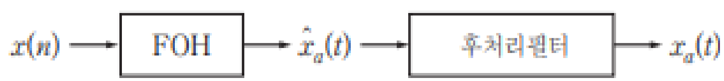


2.6 샘플링과 복원

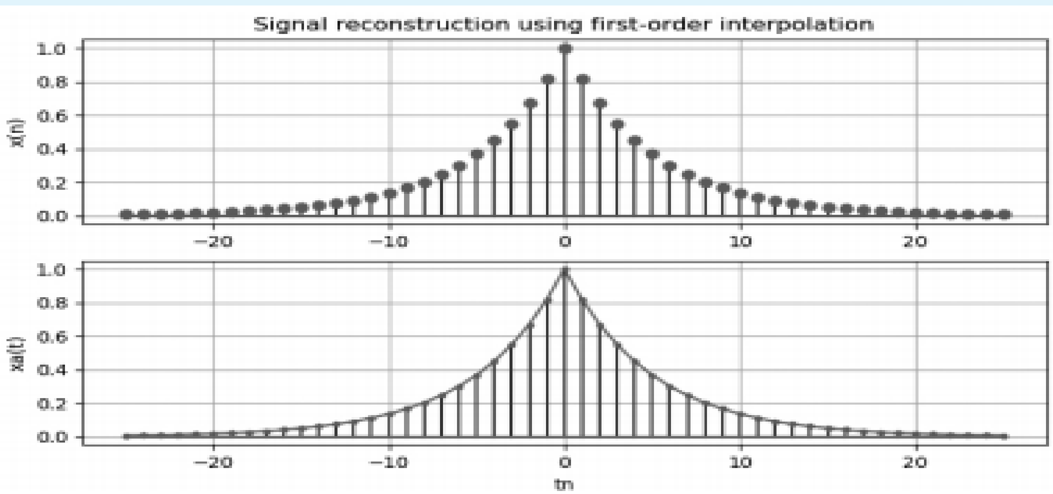
2. 일차유지보간(first-order hold interpolation): 식 (2-39)에서와 같이 이웃하는 샘플들 사이를 직선으로 연결하는 방법이다. 입출력 사이의 관계 방정식은 1차방정식이다. 따라서 일차유지보간 방법이라 부른다.

$$h_1(t) = \begin{cases} 1 + \dfrac{t}{T_s}, & 0 < t < T_s \\ 1 - \dfrac{t}{T_s}, & T_s \leq t \leq 2T_s \\ 0, & \text{otherwise} \end{cases} \tag{2-39}$$

정확한 복원을 위해서는 이후에 적절하게 설계된 아날로그 후처리필터가 필요하다.



예제 2-9 예제 2-7의 $x(n)$에 대해 일차유지 보간법을 사용하여 복원하시오.

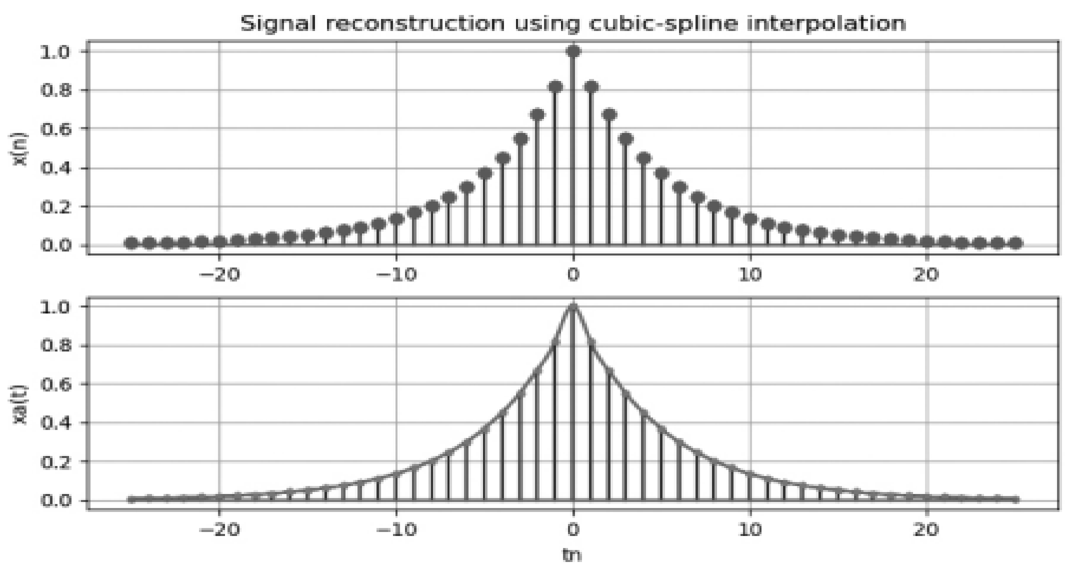


3. 큐빅-스플라인보간(Cubic-Spline interpolation): 큐빅-스플라인 보간은 보다 정밀하고 유연한 보간을 위해서 3차 보간식을 적용한다. 보간 함수식은 식 (2-40)과 같다.

$$h_c(t) = a_0(n) + a_1(n)(t - nT_s) + a_2(n)(t - nT_s)^2 + a_3(n)(t - nT_s)^3 \quad (2\text{-}40)$$

여기서 $\{\alpha_i(n),\ 0 \le i \le 3\}$은 샘플 값들에 대해 최소제곱법을 사용해서 결정한 다항식 계수들이다

예제 2-10 예제 2-7의 $x(n)$에 대해 큐빅-스플라인 보간을 사용하여 신호를 복원하시오.



참고

상관관계의 종류에는 자기 상관(auto correlation) 과 상호 상관(cross correlation) 두 종류

- 자기 상관(Auto correlation) :

  어떤 신호와 그 신호와 똑같은 신호에 대한 상관관계

  [스마트폰에 저장된 본인 지문과 스마트폰 사용을 위한 지문 터치 신호] [잡음이 섞인 신호에서 원래 신호를 찾는데 사용]

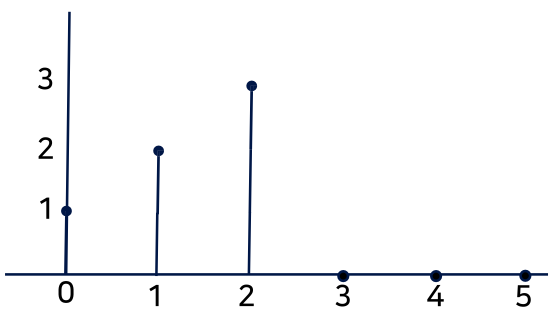
- 상호 상관(Cross correlation) :

  어떤 신호와 다른 신호 2개에 대한 상관관계

  [레이다에서의 수신된 미확인 물체와 저장된 적기들 과의 상관 관계] [패턴인식, 생체인식, 문자인식…]

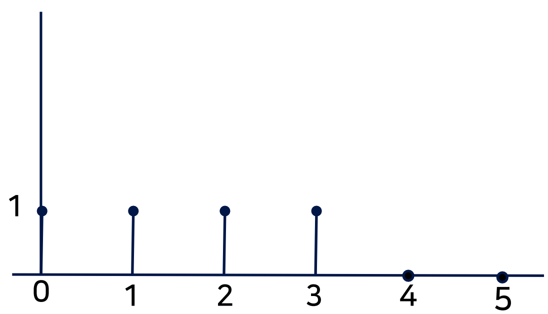
참고

상관관계는 곱하기, 덧셈과 이동 3가지 연산이 반복적으로 이루어진다.



Signal a[n]

a[0] = 1
a[1] = 2
a[2] = 3
a[3] = 0
a[4] = 0
a[5] = 0



Signal b[n]

b[0] = 1
b[1] = 1
b[2] = 1
b[3] = 1
b[4] = 0
b[5] = 0

그림. 2개의 디지털 신호 a[n], b[n]

상관관계는 다음 3가지 연산을 진행

1. 이동(Shifting)
2. 곱하기(Multiplication)
3. 더하기(Addition) 를

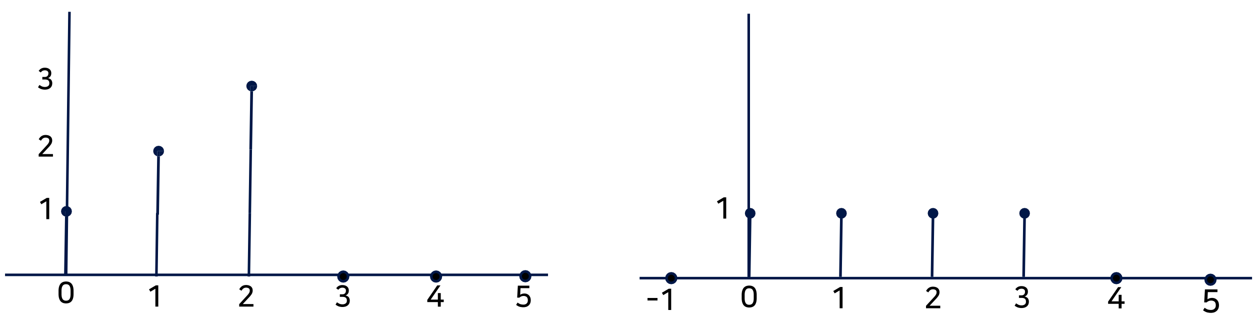
계속 해서 진행

왼쪽 그림 1. 과 같은 a[n], b[n] 신호에 대해서 상관관계는

2 신호가 다르므로, 상호 상관관계

참고

상관관계는 곱하기, 덧셈과 이동 3가지 연산이 반복적으로 이루어진다.



Signal a[n]
a[-1] = 0
a[0] = 1
a[1] = 2
a[2] = 3
a[3] = 0
a[4] = 0
a[5] = 0

Signal b[n]
b[-1] = 0
b[0] = 1
b[1] = 1
b[2] = 1
b[3] = 1
b[4] = 0
b[5] = 0

그림. 2개의 디지털 신호 a[n], b[n]

2개의 신호 중 하나를 선택
(b[n] 선택)

① 이동을 하지 않는다. 이때 k=0

각각의 샘플링 된 위치
(n=0,1,2,3..) 에서

a[n] 과 b[n] 을 곱해서 더한다.

n=0 n=1 n=2 n=3

n=4 n=5 1X1 +

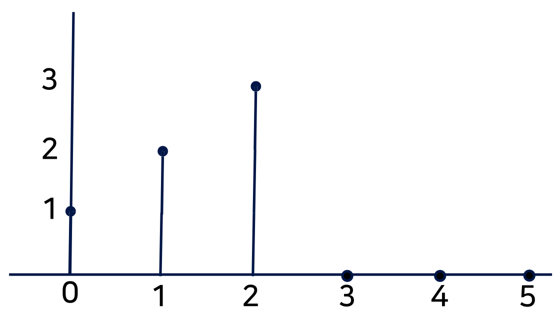
2X1 + 3X1 + 0X1 + 0X0 +

0X0 = 6.

k= 0 일 때, $C_{k=0}$ = 6

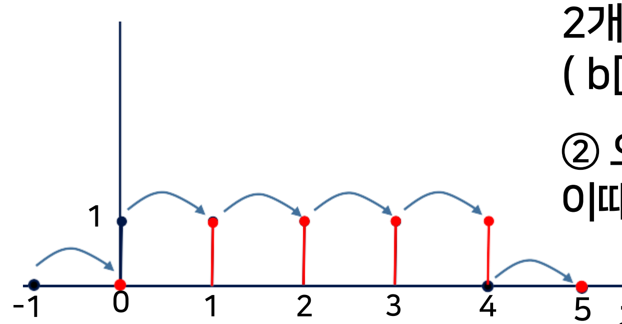
참고

상관관계는 곱하기, 덧셈과 이동 3가지 연산이 반복적으로 이루어진다.



Signal a[n]

a[0] = 1
a[1] = 2
a[2] = 3
a[3] = 0
a[4] = 0
a[5] = 0



Signal b[n]

b[0] = 0
b[1] = 1
b[2] = 1
b[3] = 1
b[4] = 1
b[5] = 0

그림. 2개의 디지털 신호 a[n], b[n]

2개의 신호 중 하나를 선택
(b[n] 선택)

② 오른 쪽으로 1개 샘플 이동한다.
이때 k= +1

각각의 샘플링된 위치

(n=0,1,2,3..) 에서 a[n] 과

b[n] 을 곱해서 더한다.

n=0 n=1 n=2 n=3

n=4 n=5

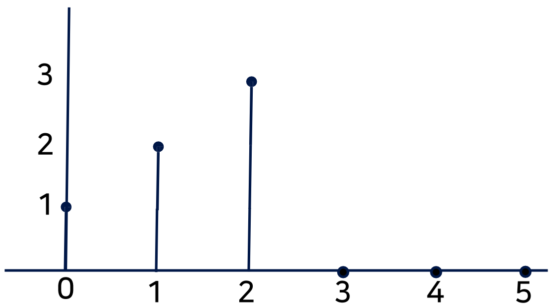
1X0 + 2X1 + 3X1 + 0X1 +

0X1 + 0X0 = 5.

k= 1 일 때, $C_{k=1} = 5$

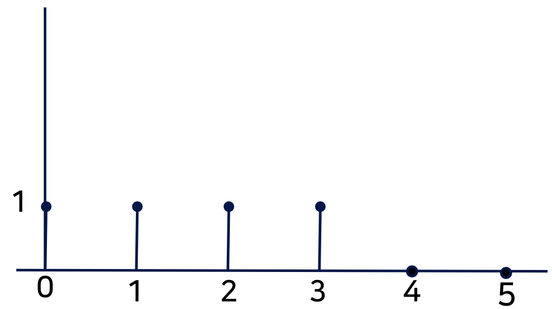
참고

상관관계는 곱하기, 덧셈과 이동 3가지 연산이 반복적으로 이루어진다.



Signal a[n]

a[0] = 1
a[1] = 2
a[2] = 3
a[3] = 0
a[4] = 0
a[5] = 0



Signal b[n]

b[0] = 1
b[1] = 1
b[2] = 1
b[3] = 1
b[4] = 0
b[5] = 0

그림. 2개의 디지털 신호 a[n], b[n]

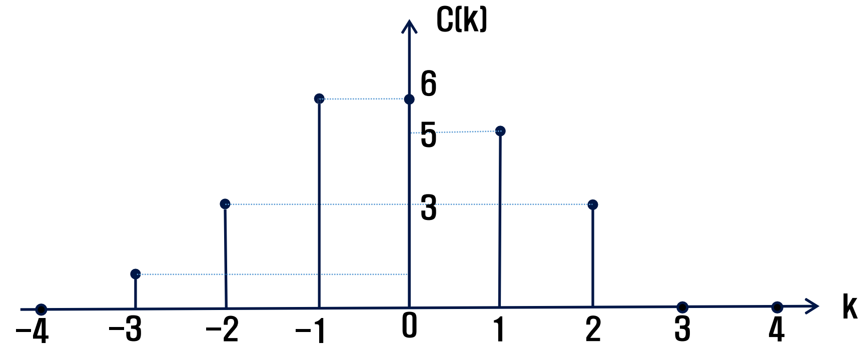
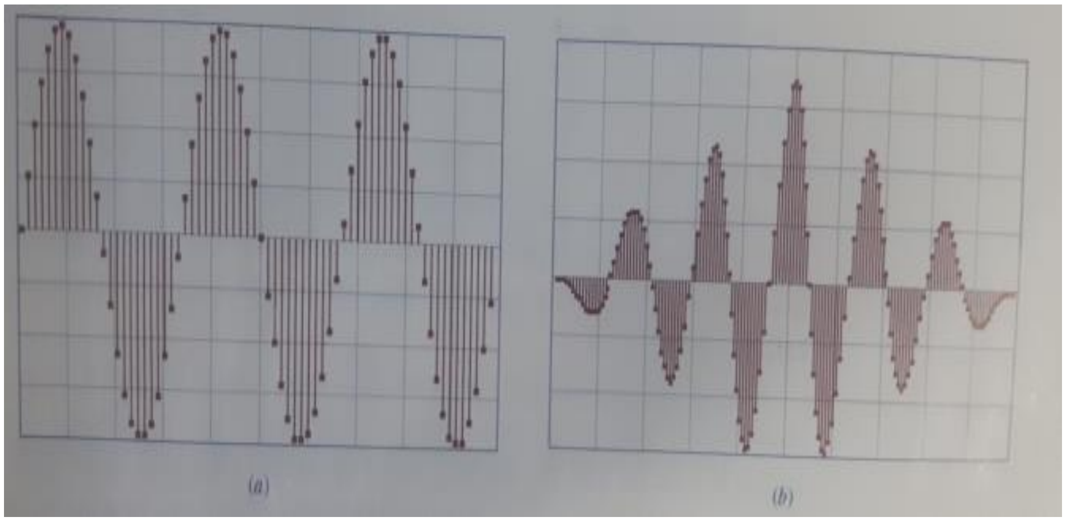


그림. a[n], b[n] 디지털신호의 상관관계 C(k)

참고

자기 상관(auto correlation) 응용 2 : 잡음이 섞인 신호에서 원래 신호를 식별하는데 응용



사인파형 신호(sine wave)

사인파형 신호에 대한 자기 상관 결과

- 사인 파에 대해서 상관관계 구하는 과정을 통하여 얻은 결과는 그림과 같은 형태
- 주기신호의 자기상관 결과도 주기 신호
- 결정된 신호(deterministic signal) 향후 값을 정확하게 예측할 수 있다. 주기 신호가 결정된 신호

참고

자기 상관(auto correlation) 응용 2 : 잡음이 섞인 신호에서 원래 신호를 식별하는데 응용

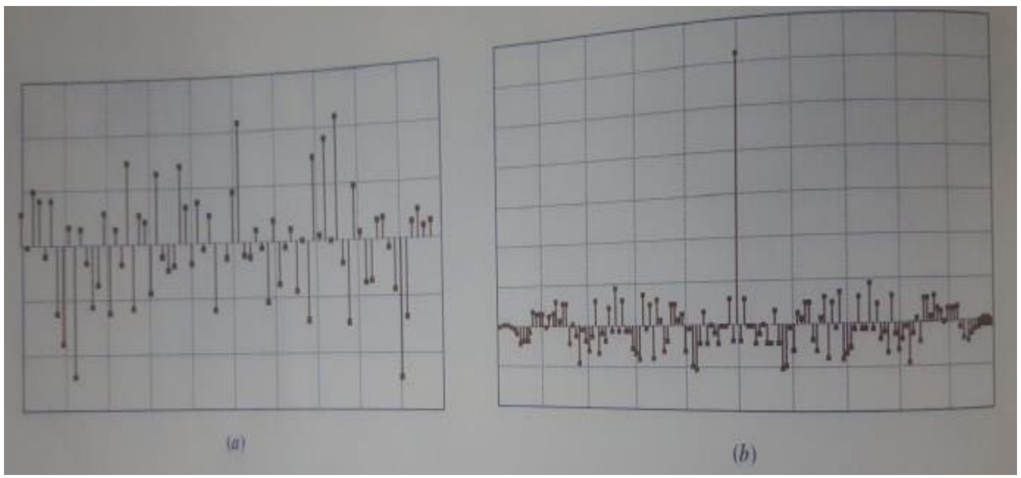


그림. 잡음 신호(random noise)　　그림. 잡음 신호에 대한 자기 상관관계

- 잡음(random noise)

측정을 통해서만 값을 알 수 있다.

-1 에서 +1 사이에 랜덤하게 값 변화

잡음에 대한 자기 상관 결과는 임펄스 형태

- 임펄스(Impulse)

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$

참고

자기 상관(auto correlation) 응용 2 : 잡음이 섞인 신호에서 원래 신호를 식별하는데 응용

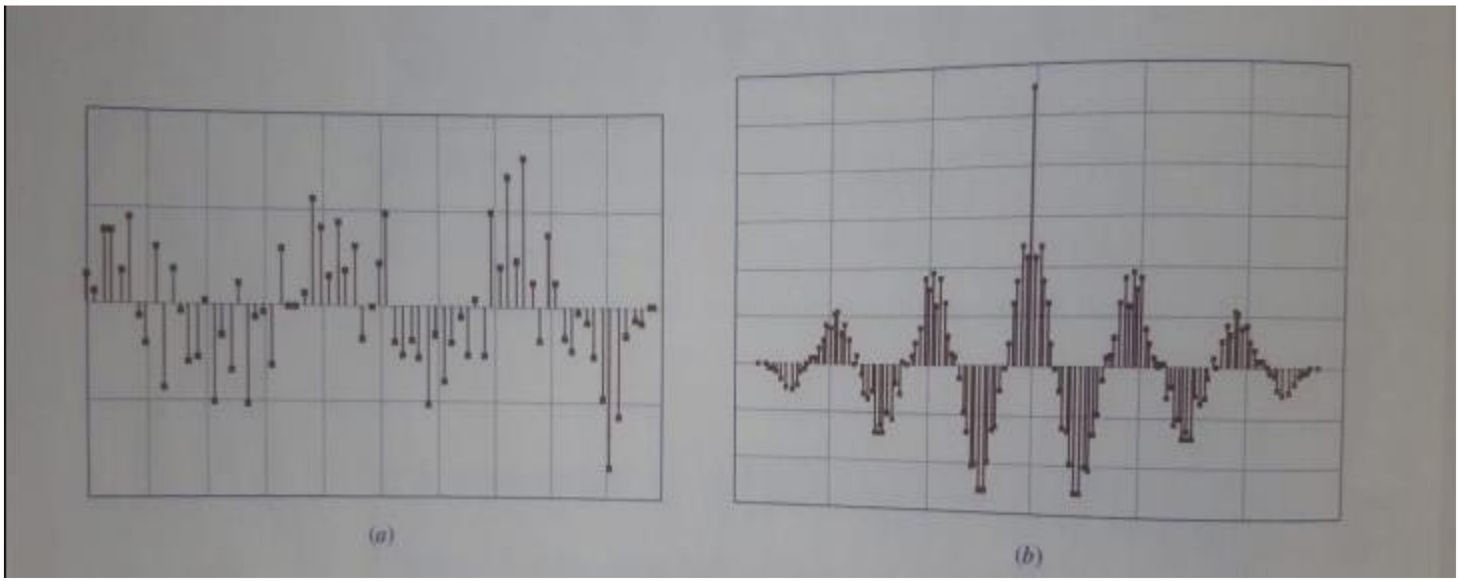


그림 . 사인파에 잡음이 섞인 신호

그림. 사인파에 잡음이 섞인 신호에 대한 자기 상관관계

Chapter 03
이산시간신호와 시스템

3.1 이산시간신호와 이산신호

3.1.1 이산시간신호

1. 연속시간신호 $x(t)$을 일정환 시간간격 T_s로 표본화(샘플링)하면 이산시간신호 $x(nT_s)$를 얻는다.
2. 이산시간신호 $x(nT_s)$에서 시간 개념인 T_s를 생략하면 이산신호 $x(n)$을 얻는다.
3. 이산신호 $x(n)$을 양자화하면 이산적인 값을 가지는 이산신호를 얻을 수 있으며, 샘플링할 때의 이산순간(discrete instances)에서의 값들 만을 모아놓은 일련의 샘플시퀀스(sample sequence)가 된다. 이를 디지털신호라고도 한다.

3.1.2 이산신호

이산신호를 표현하기 위해서는 샘플링 순간에서의 샘플 값들의 배열, 즉 샘플시퀀스(sample sequence)와 샘플링 차례를 정수 형태로 나타내는 숫자시퀀스(number sequence), 즉 순서시퀀스(order sequence)도 함께 제공되어야 한다.

3.1 이산시간신호와 이산신호

이산신호를 표현하기 위해서는 샘플링 순간에서의 샘플 값들의 배열, 즉 샘플시퀀스(sample sequence)와 샘플링 차례를 정수 형태로 나타내는 숫자시퀀스(number sequence), 즉 순서시퀀스(order sequence)도 함께 제공되어야 한다.

<이산신호를 표기하는 두 가지 방법>

1. 샘플시퀀스 $x(n)$과 순서시퀀스 n, 둘 다를 나타내는 방법:

$$x(n) = \{2, 6, 3, 4, 8, 0, 7, 2, 5\}, \quad n = \{-3, \ -2, \ -1, 0, 1, 2, 3, 4, 5\}$$

위의 샘플시퀀스 $x(n)$ 은 순서시퀀스 n 을 참조하면 이 신호의 영점이 네 번째 샘플에 있으며, 이 영점을 기준으로 음과 양의 방향으로 순서가 정해진다.

2. 순서시퀀스 n을 별도로 나타내지 않고 샘플시퀀스 $x(n)$에 직접 영점인 지점을 위로 향하는 화살표로 표기하는 방법 :

$$x(n) = \{2, 6, 3, \underset{\uparrow}{4}, 8, 0, 7, 2, 5\}$$

위의 방법으로 표시할 경우, 화살표를 기준으로 $n = 0$으로 두면 순서시퀀스 n을 찾아낼 수 있다.

$$n = \{-3, \ -2, \ -1, 0, 1, 2, 3, 4, 5\}$$

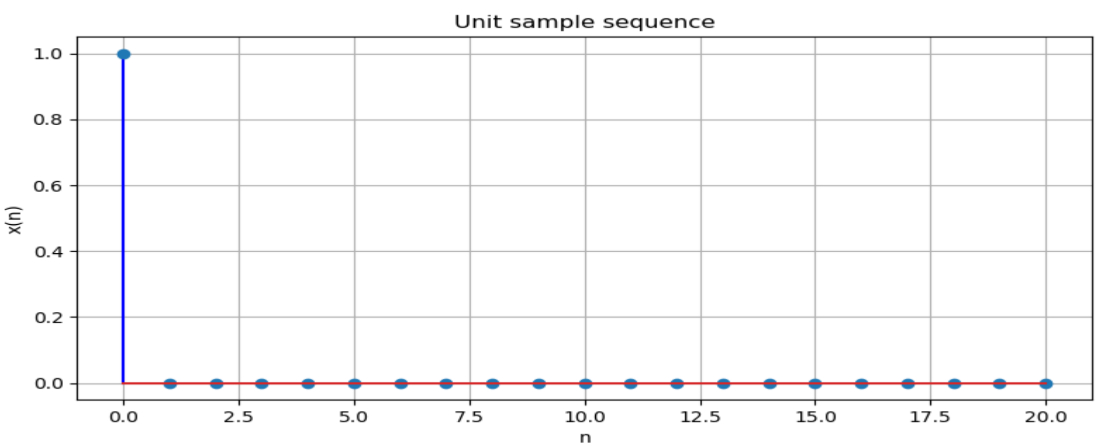
3.2 이산신호의 종류

디지털 신호처리에서 사용되는 기본 시퀀스(basic sequences)에는 다음과 같은 것들이 있다.

1. 단위샘플 시퀀스 또는 단위임펄스(unit impulse)
2. 단위계단 시퀀스
3. 실수지수 시퀀스
4. 복소지수 시퀀스
5. 정현파 시퀀스
6. 무작위 시퀀스
7. 주기 시퀀스

1. 단위샘플(unit sample) 시퀀스: 다음 식 (3-1)과 같은 델타(delta)함수로 표기하며, n=0인 위치에서만 단위샘플이 존재하고 그 외에서는 0인 시퀀스를 말한다.

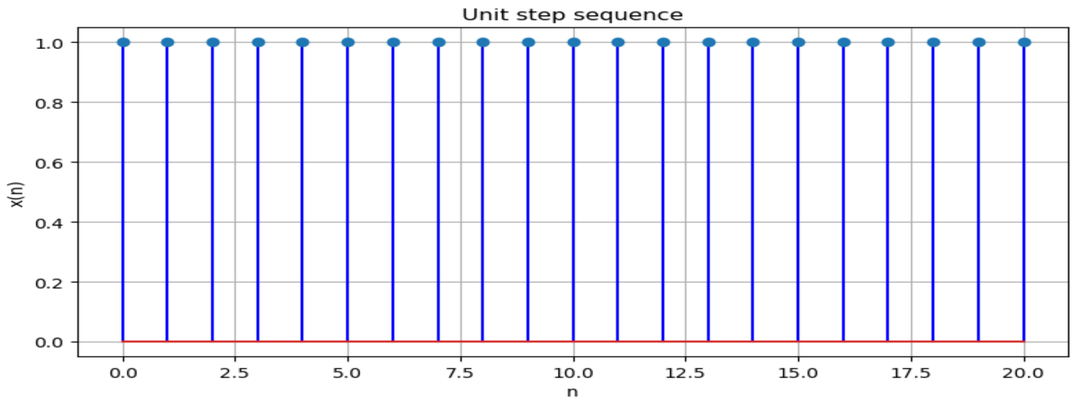
$$\delta(n) = \begin{cases} 1, & n=0 \\ 0, & n \neq 0 \end{cases}, \qquad \delta(n) = \{\cdots, 0, 0, \underset{\uparrow}{1}, 0, 0, \cdots\} \qquad (3\text{-}1)$$



3.2 이산신호의 종류

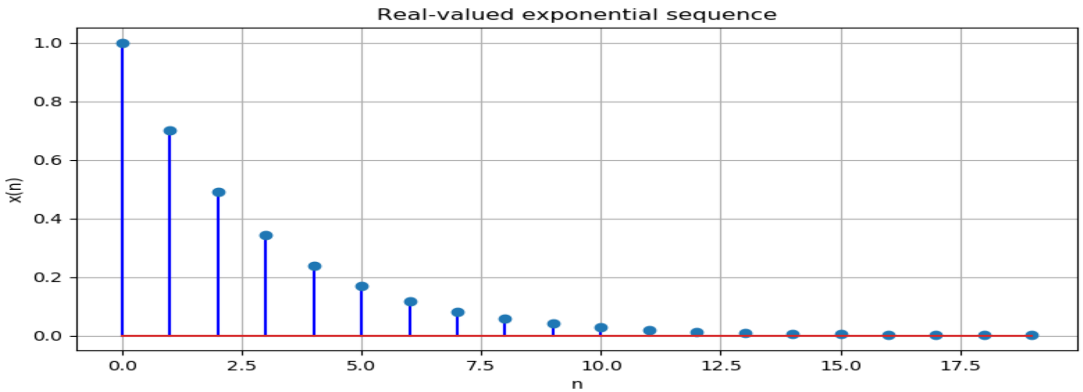
2. 단위계단(unit step) 시퀀스: 다음 식 (3-2)와 같이 단위샘플이 연속으로 나타나는 시퀀스이다.

$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}, \qquad u(n) = \{\cdots, 0, 0, 0, \underset{\uparrow}{1}, 1, 1, \cdots\} \qquad (3\text{–}2)$$



3. 실수지수(real-valued exponential) 시퀀스 : 다음 식 (3-3)과 같이 실수의 지수 형태로 표현되는 신호이다. 양의 n에 대해, $a < 0$인 경우에는 1에서부터 시작해서 지속적으로 감소하여 0에 근접해가는 형태를 나타낸다.

$$x(n) = a^n, \qquad \forall n; a \in R \qquad (3\text{–}3)$$



3.2 이산신호의 종류

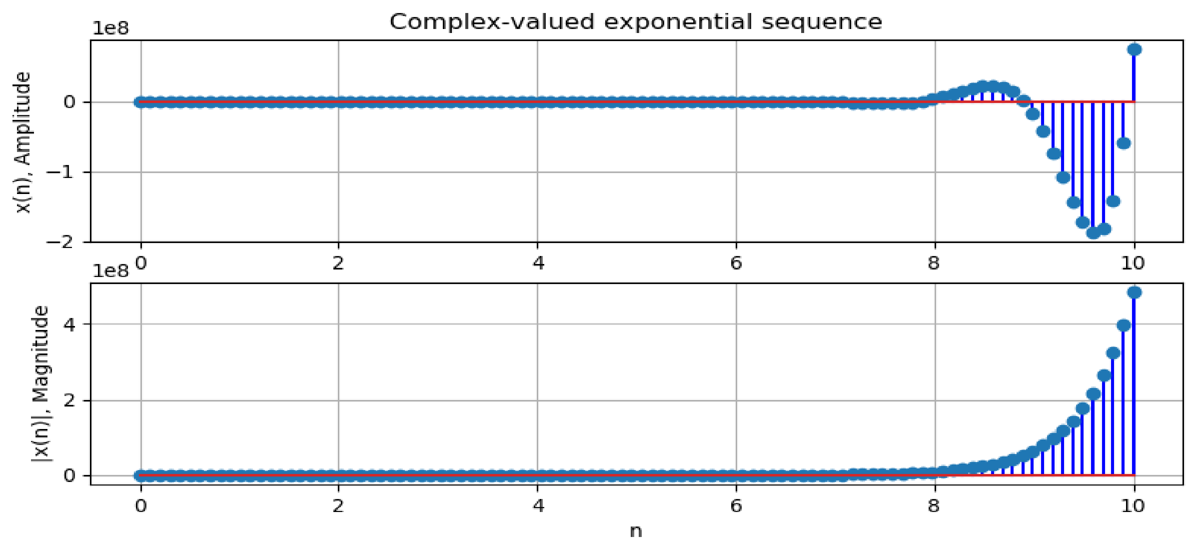
4. 복소지수(complex-valued exponential) 시퀀스: 다음 식 (3-4)와 같이 복소수 지수의 형태로 정의된다.

$$x(n) = e^{(\sigma + j\omega)n}, \qquad \forall n \qquad (3\text{-}4)$$

여기서 σ는 감쇄(attenuation)(< 0) 또는 증폭(amplification)(> 0)작용을 하고, ω는 라디안주파수이다. 예를 들어 $\sigma = 2, \omega = 3$인 복소지수 시퀀스는 다음과 같이 표현된다. 따라서 n에 따라서 크기는 e^n으로 변하고 파형은 사인파와 코사인파의 합성 형태로 나타날 것이다.

$$x(n) = e^{(2+j3)n} = e^{2n} \cdot e^{j3n} = e^{2n}(\cos(3n) + j\sin(3n))$$

복소지수 시퀀스 $x(n)$과 크기 $|x(n)|$의 파형은 다음 그림과 같이 나타난다.



3.2 이산신호의 종류

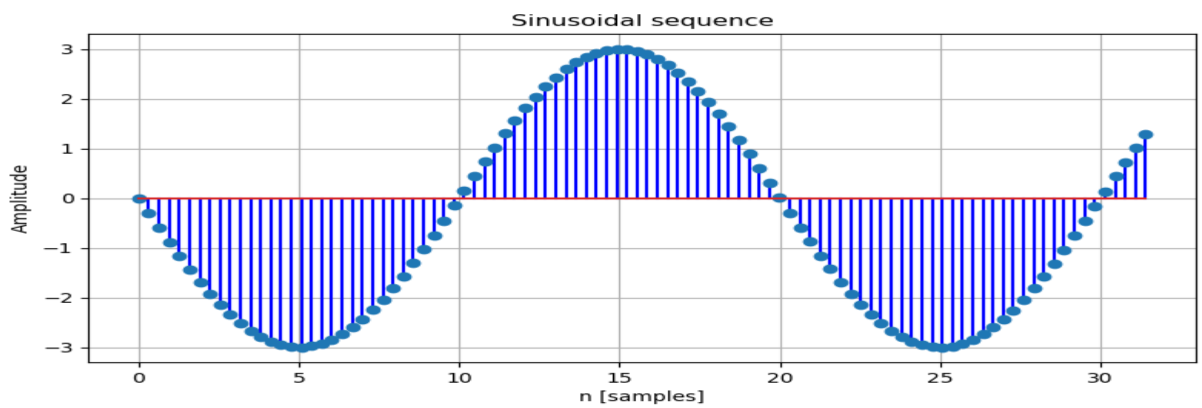
5. 정현파(sinusoidal wave) 시퀀스: 다음 식 (3-2)와 같이 코사인(또는 사인)함수로 정의된다. 여기서 A는 진폭이며, ϕ는 라디안 단위의 위상을 나타낸다.

$$x(n) = A\cos(\omega n + \phi), \qquad \forall n \qquad (3-5)$$

예를 들어 다음과 진폭이 3이고 디지털 라디안주파수가 0.1π, 위상은 $\pi/2(= 90^o)$인 정현파 시퀀스는 다음 식과 같다.

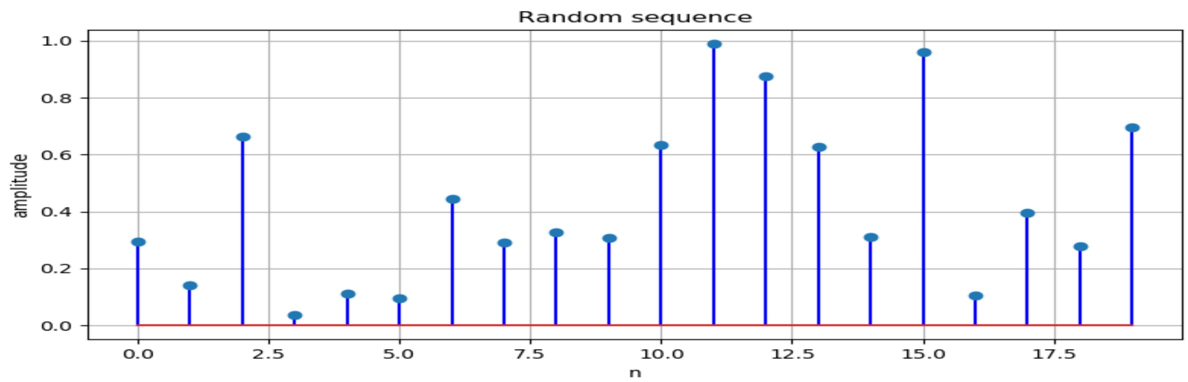
$$x(n) = 3\cos\left(0.1\pi n + \frac{\pi}{2}\right)$$

이 정현파 시퀀스는 20개 샘플($n = 20$) 만에 2π 라디안($0.1\pi \times 20 = 2\pi$), 즉 $n = 0$에서 시작하여 $n = 19$일 때 한 주기가 완성된다.

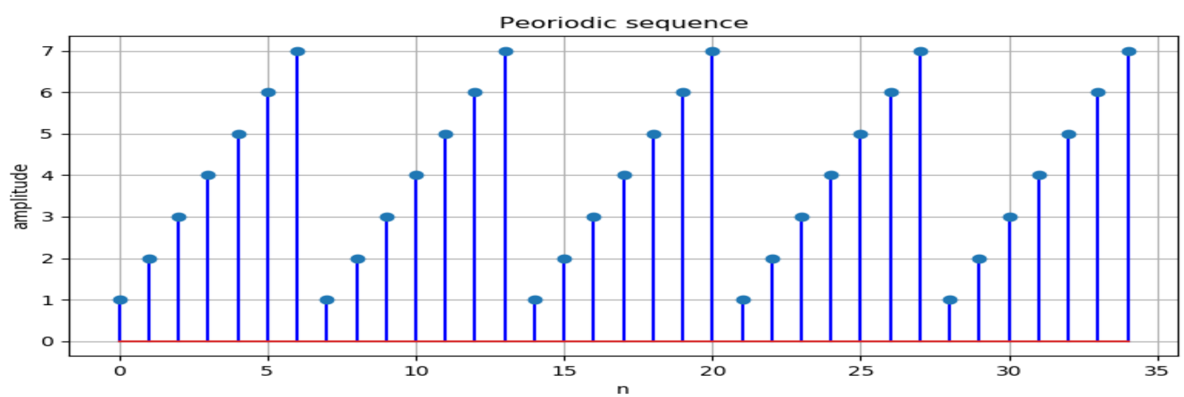


3.2 이산신호의 종류

6. 무작위(random) 시퀀스: 많은 실제적인 시퀀스에는 앞의 시퀀스들처럼 수학적 표현으로 기술될 수 없는 잡음(noise)이 섞여 있는 경우가 대부분이다. 잡음과 같이 수학적으로 기술할 수 없는 시퀀스를 무작위 또는 난수(random) 시퀀스 또는 확률론적(stochastic) 시퀀스라 부른다.



7. 주기(periodic) 시퀀스: 일정한 시퀀스가 특정 구간(주기)마다 반복적으로 나타나는 형태의 시퀀스이며, 만약 $x(n) = x(n+T)$이면 시퀀스 $x(n)$은 주기적이다. 여기서 T는 관계를 만족하는 가장 작은 정수이며 기본 주기(fundamental period)를 나타낸다. 주기적 시퀀스의 표기는$\widetilde{x(n)}$으로 표기한다.



3.3 이산신호의 연산

이산 신호들 사이의 기본 연산 종류:

1. 신호 덧셈과 곱셈 연산
2. 크기 조정, 이동, 반전 연산
3. 샘플 합과 곱 연산
4. 신호 에너지와 전력 연산

3.3.1 기본 연산

1. 신호 덧셈(signal addition): 이산 신호 사이의 더하기이다. 즉 샘플 시퀀스 사이의 합 연산이다. 연산을 위해서는 영점을 중심으로 두 시퀀스의 길이를 일치시켜야 한다.

$$x(n) = x_1(n) + x_2(n)$$

다음의 두 시퀀스 $x_1(n)$과 $x_2(n)$의 경우에, 영점과 길이가 모두 다르다.

$$x_1(n) = \{\underset{\uparrow}{0}, 0, 0, 1, 2, 3, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1\}$$

$$x_2(n) = \{1, 2, 3, 4, 5, \underset{\uparrow}{6}, 7, 6, 5, 4, 3, 2, 1\}$$

두 시퀀스의 순서시퀀스 n_1과 n_2는 다음과 같다.

$$n_1 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$$
$$n_2 = \{-5,\ -4,\ -3,\ -2,\ -1, 0, 1, 2, 3, 4, 5, 6, 7\}$$

3.3 이산신호의 연산

출력 시퀀스를 구하는 연산 순서는 다음과 같다.

(1) 출력 시퀀스 $y(n)$의 순서 시퀀스 n과 길이 N을 구한다. 시퀀스의 시작 위치 b와 끝 위치 s를 다음과 같이 구한다. 따라서 순서 시퀀스 n은 -5부터 15까지가 되고, 길이 N은 21이 된다.

$$b = \min(\min(n_1),\ \min(n_2)) = \min(0,\ -5) = -5$$
$$s = \max(\max(n_1),\ \max(n_2)) = \max(15, 7) = 15$$

$$N = s - b + 1 = 15 - (-5) + 1 = 21$$

(2) 길이가 N인 세 개의 영 시퀀스(zero sequence)를 할당하고, 입력 시퀀스 $x_1(n)$과 $x_2(n)$을 길이 N으로 확장한 다음, 같은 순서에 해당하는 샘플 값들을 서로 더해서 출력 시퀀스 $y(n)$을 구한다.

$$zx_1(n) = \{0,0,0,0,0,\overbrace{\boxed{\underset{\uparrow}{0},0,0,1,2,3,4,5,6,7,6,5,4,3,2,1}}^{x_1(n)}\}$$

$$zx_2(n) = \{\overbrace{\boxed{1,2,3,4,5,\underset{\uparrow}{6},7,6,5,4,3,2,1}}^{x_2(n)},0,0,0,0,0,0,0,0\}$$

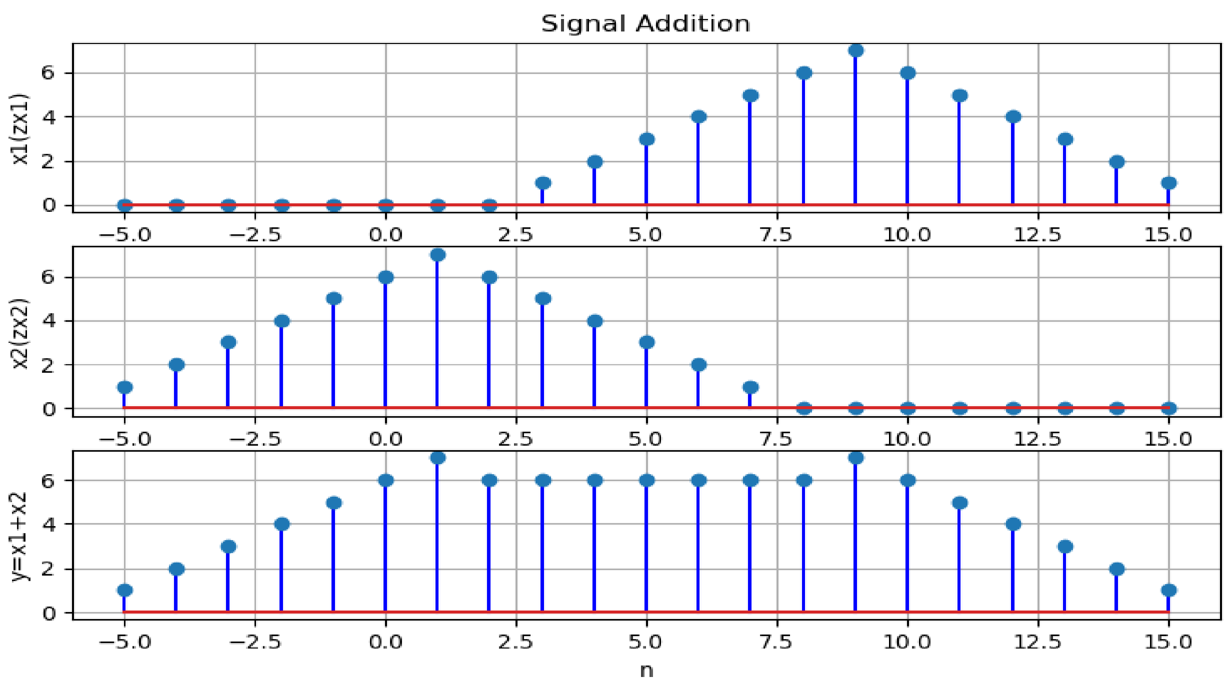
$$y(n) = \{1,2,3,4,5,\underset{\uparrow}{6},7,6,6,6,6,6,6,6,7,6,5,4,3,2,1\}$$

3.3 이산신호의 연산

앞의 파일을 실행시키면 다음과 같이 길이 N=21로 확장된 입력신호 시퀀스 $x_1(n)$과 $x_2(n)$, 신호 합 출력 시퀀스 $y(n)$, 그리고 순서 시퀀스 n이 출력된다.

```
y(n)= [1. 2. 3. 4. 5. 6. 7. 6. 6. 6. 6. 6. 6. 6. 7. 6. 5. 4. 3. 2. 1.]
x1(n)= [0. 0. 0. 0. 0. 0. 0. 0. 1. 2. 3. 4. 5. 6. 7. 6. 5. 4. 3. 2. 1.]
x2(n)= [1. 2. 3. 4. 5. 6. 7. 6. 5. 4. 3. 2. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
n= [-5 -4 -3 -2 -1  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15]
```

또한 신호 덧셈의 결과를 그림으로 볼 수 있다.



3.3 이산신호의 연산

2. 신호 곱셈(signal multiplication): 이산 신호 사이의 곱하기이다. 즉 샘플 시퀀스 사이의 곱 연산이다. 연산을 위해서는 신호 더하기에서와 마찬가지로영점을 중심으로 두 시퀀스의 길이를 일치시켜야 한다. 연산자는 "·"을 사용하며, 파이썬 프로그램에서는 "*"를 사용한다.

$$y(n) = x_1(n) \cdot x_2(n)$$

- 연산 순서는 신호 덧셈 때와 동일하게 순서 시퀀스 n과 길이 N을 구한 다음, 같은 순서에 있는 샘플 값들을 서로 곱해서 출력을 구한다.

$$b = \min(\min(n_1),\ \min(n_2)) = \min(0,\ -5) = -5$$
$$s = \max(\max(n_1),\ \max(n_2)) = \max(15, 7) = 15$$

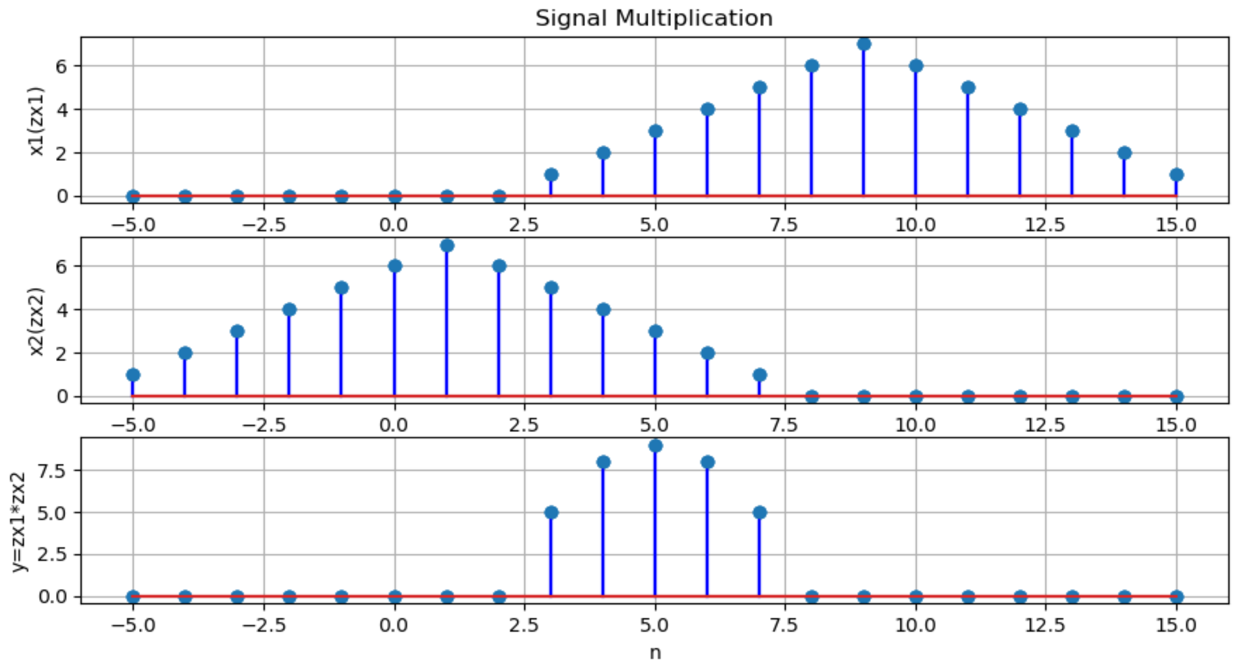
$$N = s - b + 1 = 15 - (-5) + 1 = 21$$

$$zx_1(n) = \{0,0,0,0,0,\overbrace{\boxed{\underset{\uparrow}{0},0,0,1,2,3,4,5,6,7,6,5,4,3,2,1}}^{x_1(n)}\}$$

$$zx_2(n) = \{\overbrace{\boxed{1,2,3,4,5,\underset{\uparrow}{6},7,6,5,4,3,2,1}}^{x_2(n)},0,0,0,0,0,0,0,0\}$$

$$y(n) = \{0,0,0,0,0,\underset{\uparrow}{0},0,0,5,8,9,8,5,0,0,0,0,0,0,0,0\}$$

3.3 이산신호의 연산



3. 크기 조정(scaling): 크기 조정은 단순하게 이산 신호의 샘플 값에 일정한 스칼라 값을 곱하여 신호의 크기를 크게 증폭시키거나 작게 감소시키는 연산이다.

$$y(n) = ax_1(n)$$

$$x_1(n) = \{0,0,0,1,2,3,4,5,6,7,6,5,4,3,2,1\}$$
$$\uparrow$$

$$y(n) = 3x_1(n) = \{0,0,0,3,6,9,12,15,18,21,18,15,12,9,6,3\}$$
$$\uparrow$$

3.3 이산신호의 연산

4. 이동(shifting): 이동 연산은 시퀀스 $x_1(n)$의 각 샘플 값에 대해 그 순서의 위치를 k만큼 이동시키는 것이며, 샘플의 크기 값에는 영향을 주지 않는다. $x_1(n-k)$에서 $n-k=0$으로 두면 $\mathrm{n}=\mathrm{k}$가 되어, $k>0$이면 양의 방향, $k<0$이면 음의 방향으로 이동하는 것을 의미한다.
예를 들어, $x_1(n-4)$이면 양의 방향으로 4 이동, $x_1(n+4)$이면 음의 방향으로 4 이동을 의미한다.

$$y(n)=x_1(n-k)$$

$$x_1(n)=\{\underset{\uparrow}{0},0,0,1,2,3,4,5,6,7,6,5,4,3,2,1\}$$

$$n_1=\{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]$$

$k=4$ 만큼 이동할 경우, 순서시퀀스는 4만큼 늘어나고 샘플 값은 $k=4$ 부터 동일하게 배열된다.

$$y(n)=\{\underset{\uparrow}{0},0,0,0,0,0,0,1,2,3,4,5,6,7,6,5,4,3,2,1\}$$

$$n=\{\underset{\uparrow}{0},1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19\}$$

3.3 이산신호의 연산

5. 반전(folding): 반전 연산은 시퀀스 $x(n)$의 각 샘플 값들에 대해 $n = 0$을 중심으로 양과 음 방향의 샘플들을 반대방향으로 서로 교환하는 연산이다.

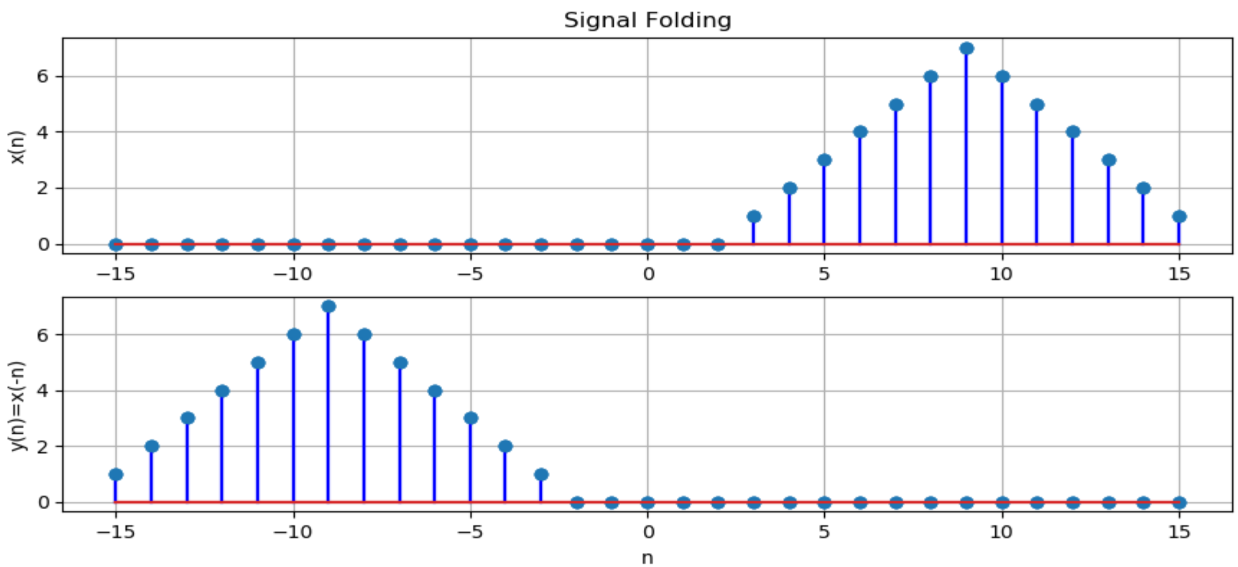
$$y(n) = x(-n)$$

$$x(n) = \{\underset{\uparrow}{0}, 0, 0, 1, 2, 3, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1\}$$

$$nx = \{\underset{\uparrow}{0}, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$$

$$y(n) = \{\underset{\uparrow}{1}, 2, 3, 4, 5, 6, 7, 6, 5, 4, 3, 2, 1, 0, 0, 0\}$$

$$ny = \{\underset{\uparrow}{-15}, -14, -13, -12, -11, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0\}$$



3.3 이산신호의 연산

6. 샘플 합(sample summation): 샘플 합 연산은 샘플 시퀀스 내에서 특정 구간 n_1에서 n_2까지의 모든 샘플 값들을 단순하게 합하는 연산이다. 넘파이의 "np.sum()" 함수를 사용하면 된다.

$$y(n) = \sum_{n=n_1}^{n_2} x(n) = x(n_1) + , \cdots, + x(n_2)$$

만약 순서 -2부터 7까지의 샘플 합을 구한다면, 사각형 구간의 샘플 값들의 단순 합을 구하는 것이다.

$$x(n) = \{0, 0, 0, 1, \boxed{2, 3, 4, 5, 6, 7, 6, 5, 4, 3}, 2, 1\}$$

$$n = \{-6, -5, -4, -3, \boxed{-2, -1, 0, 1, 2, 3, 4, 5, 6, 7}, 8, 9\}$$

7. 샘플 곱(sample products): 샘플 합 연산은 샘플 시퀀스 내에서 특정 구간 n_1에서 n_2까지의 모든 샘플 값들을 단순하게 곱하는 연산이다. 넘파이의 "np.sum()" 함수를 사용하면 된다.

$$y(n) = \prod_{n=n_1}^{n_2} x(n) = x(n_1) \times , \cdots, \times x(n_2)$$

만약 순서 -2부터 7까지의 샘플 곱을 구한다면, 사각형 구간의 샘플 값들의 단순 곱을 구하는 것이다.

$$x(n) = \{0, 0, 0, 1, \boxed{2, 3, 4, 5, 6, 7, 6, 5, 4, 3}, 2, 1\}$$

$$n = \{-6, -5, -4, -3, \boxed{-2, -1, 0, 1, 2, 3, 4, 5, 6, 7}, 8, 9\}$$

3.3 이산신호의 연산

8. 신호에너지(signal energy): 신호 에너지는 시퀀스 $x(n)$의 에너지를 말하며, $x(n)$과 $x(n)$의 공액(conjugate) 사이의 곱으로 계산되며, 샘플 값들의 제곱의 합으로도 계산된다.

$$E = \sum_{n=-\infty}^{\infty} x(n)x^*(n) = \sum_{n=-\infty}^{\infty} |x(n)|^2$$

예제 3-1 다음의 시퀀스들을 각각의 지정된 구간에 대해 발생시키고 그래프로 나타내시오.

a. $x(n) = 2\delta(n+2) - \delta(n-4), \quad -5 \leq n \leq 5$

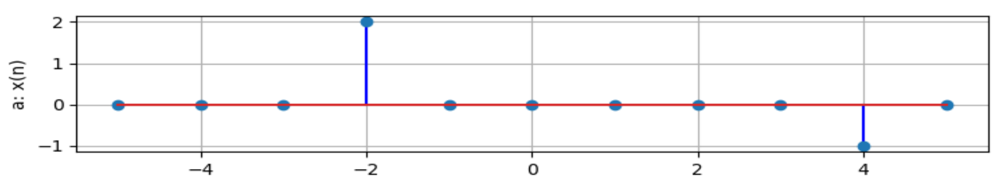
b. $x(n) = n[u(n) - u(n-10)] + 10e^{-0.3(n-10)}[u(n-10) - u(n-20)], \quad 0 \leq n \leq 20$

c. $x(n) = \cos(0.04\pi n) + 0.2\omega(n), \quad 0 \leq n \leq 20$

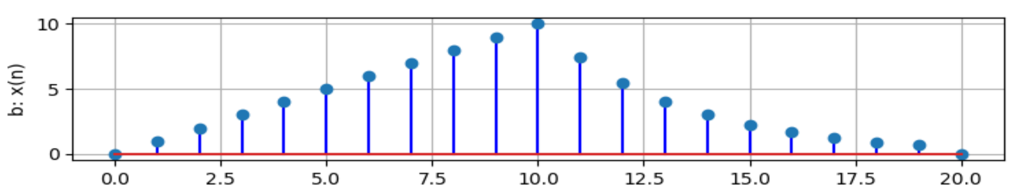
여기서 $\omega(n)$은 평균이 0이고 분산이 1인 가우시안 랜덤 시퀀스이다.

d. $\tilde{x}(n) = \{\cdots, 5, 4, 3, 2, 1, 5, 4, 3, 2, 1, 5, 4, 3, 2, 1, \cdots\}, \quad -10 \leq n \leq 9$

a. 신호 $x(n)$은 $\mathrm{n} = -2$ 에 크기가 2인 임펄스와 $\mathrm{n} = 4$ 에 크기가 1인 임펄스가 있는 신호이다.

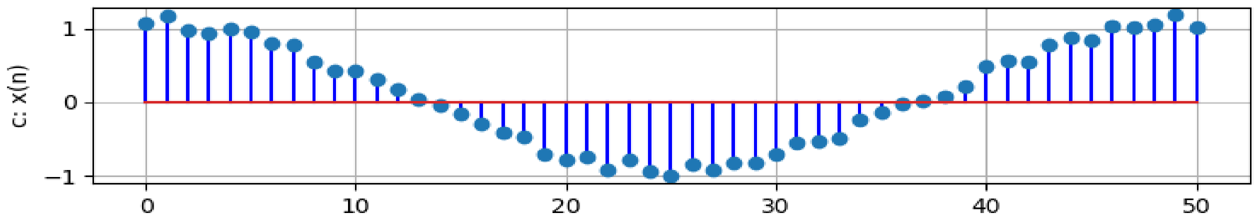


b. 신호 $x(n)$ 은 $\mathrm{n} = 0$에서 시작하여 $\mathrm{n} = 10$까지는 단조 증가하고, 다시 $\mathrm{n} = 10$부터 $n = 20$까지는 기하급수적으로 감소하는 형태의 신호이다.



3.3 이산신호의 연산

c. 입력신호 $x(n)$ 의 코사인 항을 보면, $\cos(0.04\pi n) = \cos(2\pi \cdot \frac{1}{50} \cdot n)$이다. 주파수가 $\frac{1}{50}$이므로 주기는 $50[samples]$가 된다. 따라서 이 신호의 한 주기는 50 샘플로 구성된다. 그러므로 이 신호는 $n = 0$에서 시작하여 $n = 49$ 일 때 1주기가 되는 정현파 시퀀스이며, 0에서 0.2 크기의 잡음이 동반되는 신호이다.



d. 단조 감소하는 신호가 $T = 5\ [samples]$ 마다 반복하는 주기신호이다.