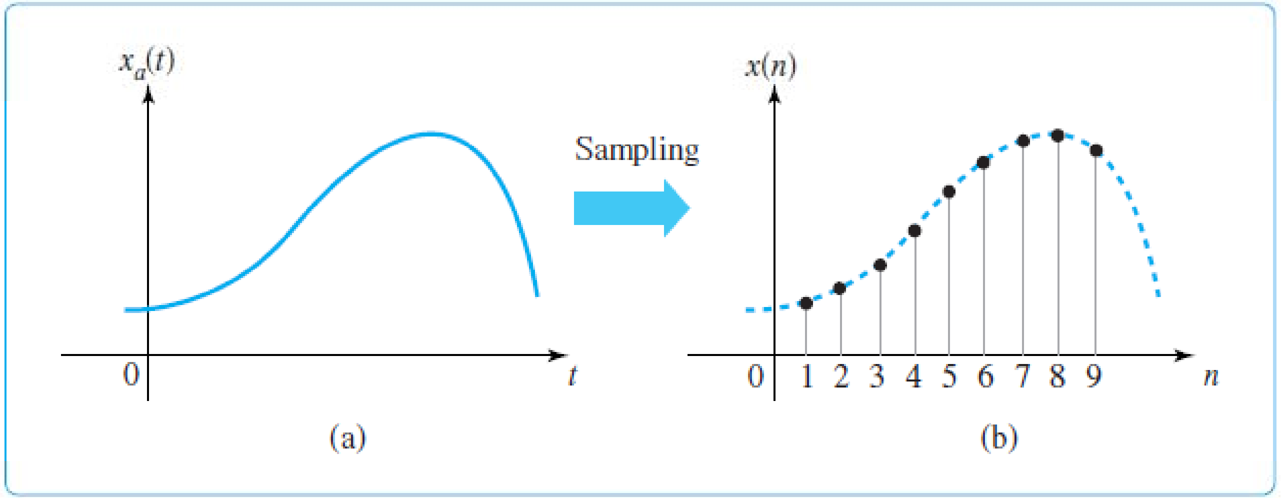
# 2. 이산 신호 및 시스템

# 목차


- 2.1 이산 신호의 정의
- 2.2 이산 시스템
- 2.3 이산 선형 시불변 시스템의 해석
- 2.4 차분 방정식으로 표현되는 시스템

# 2.1 이산 신호의 정의

- 이산 신호(discrete-time signal)
  - 정수의 독립 변수를 갖는 함수로 정의
  - 샘플링 간격 *T*와는 관계없이 항상 정수의 독립 변수를 가짐



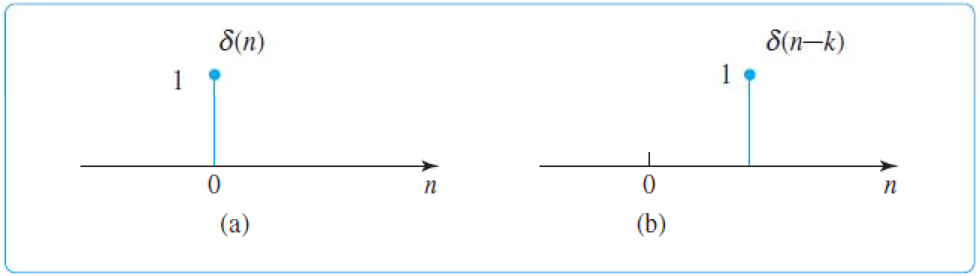

| 그림 2.1 | 샘플링하여 얻은 이산 신호

# 2.1 이산 신호의 정의

## 2.1.1 기본적인 이산 신호

- (1) 단위 임펄스 함수(unit impulse function)
  - 델타 함수 $\delta[n]$

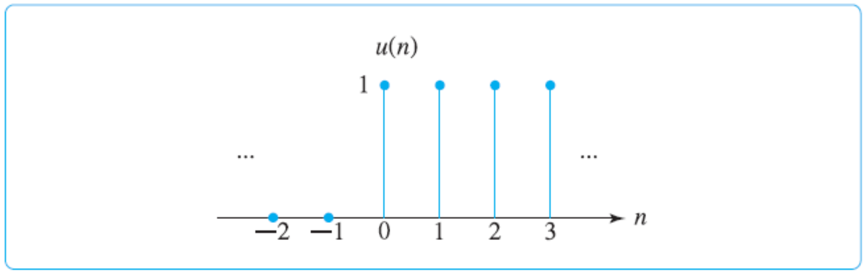
$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$




| 그림 2.2 |  단위 임펄스 함수

- (2) 단위 계단 함수(unit step function)

$$u(n) = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$$

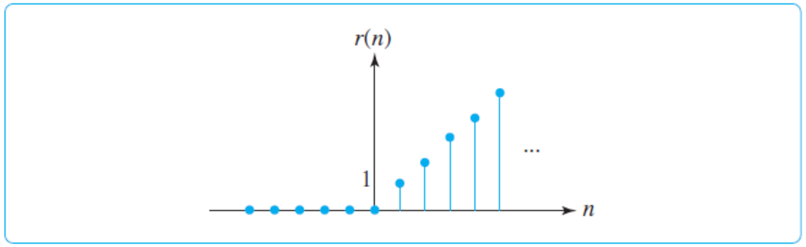



| 그림 2.4 |  단위 계단 함수

# 2.1 이산 신호의 정의

◦ (3) 단위 램프 함수(unit ramp function)

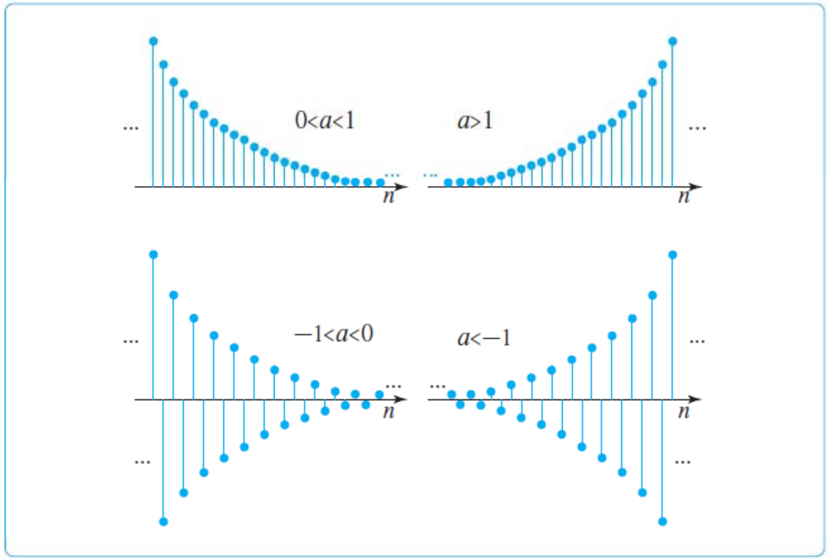
$$r(n) = \begin{cases} n & n \ge 0 \\ 0 & n < 0 \end{cases}$$




| 그림 2.5 | 단위 램프 함수

◦ (4) 지수 함수

$$x(n) = r^n e^{j\theta n}$$
$$= r^n[\cos\theta n + j\sin\theta n]$$




| 그림 2.6 | a 값에 따른 지수 신호의 형태

# 2.1 이산 신호의 정의

## 2.1.2 이산 신호의 분류

- (1) 에너지 신호와 전력 신호
  - 에너지 신호(energy signal)
    - 에너지가 유한한 값을 가지는 신호

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

  - 전력 신호(power signal)
    - 평균 전력이 유한한 값을 가지는 신호

$$P \equiv \lim_{N \to \infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x(n)|^2$$

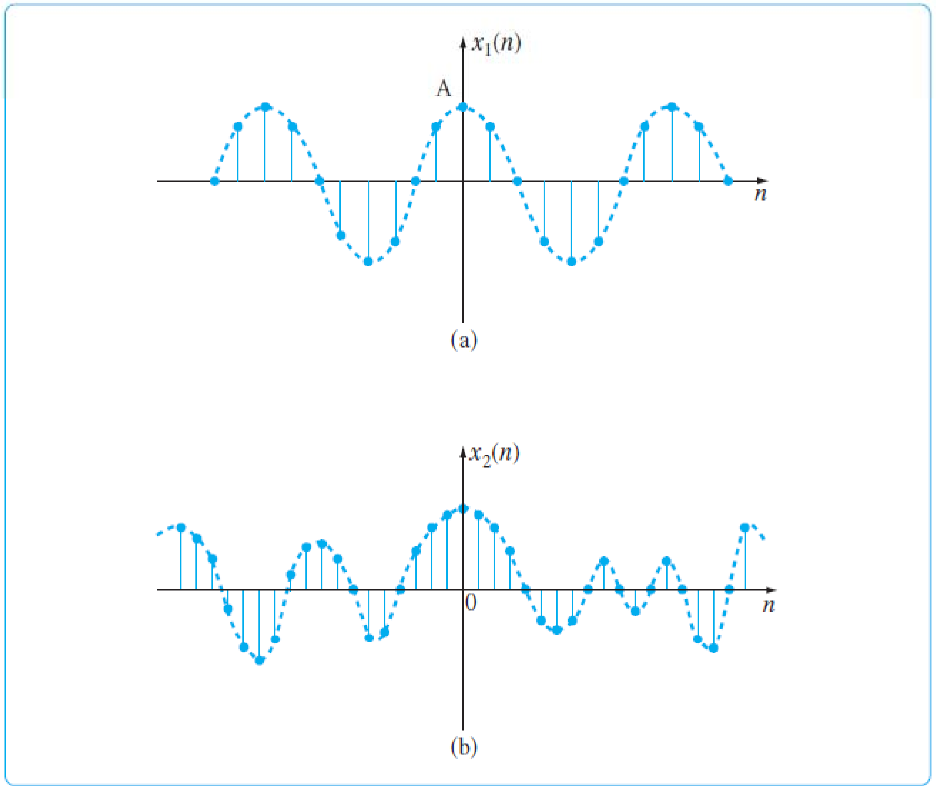
# 2.1 이산 신호의 정의

- (2) 주기 신호와 비주기 신호
  - 주기 신호(periodic signal)

$$x(n+N) = x(n)$$

  - 비주기 신호

$$x(n+N) \neq x(n)$$

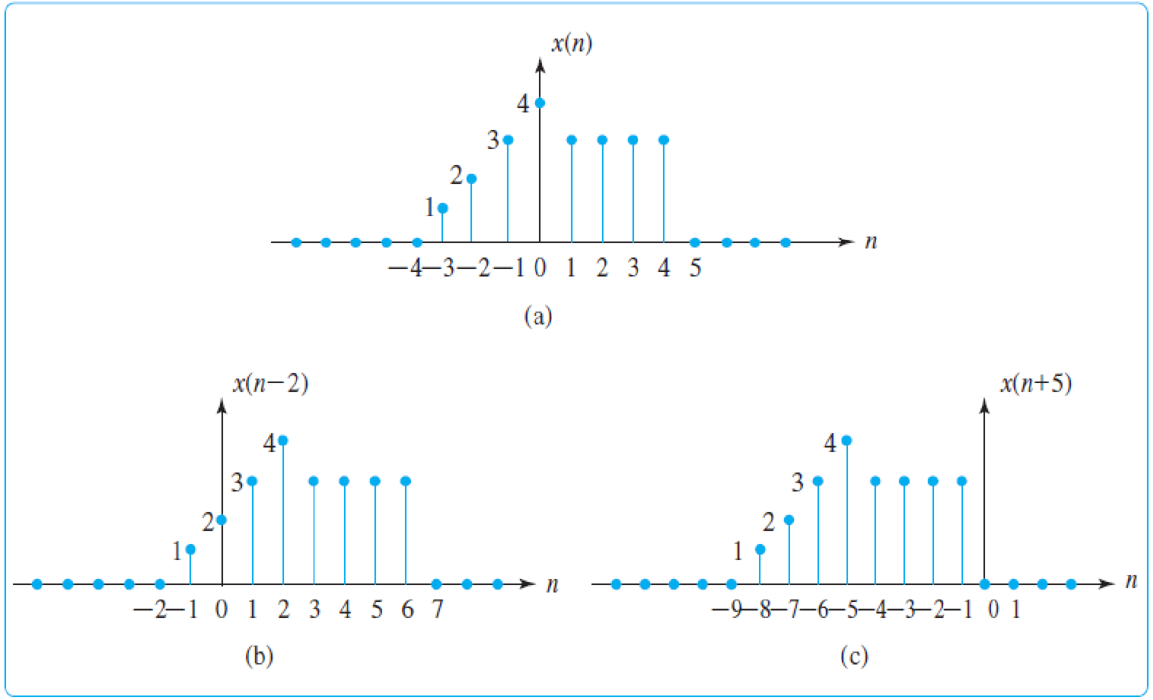



| 그림 2.7 | (a) 주기 신호 (b) 비주기 신호

# 2.1 이산 신호의 정의

## 2.1.3 이산 신호의 변형

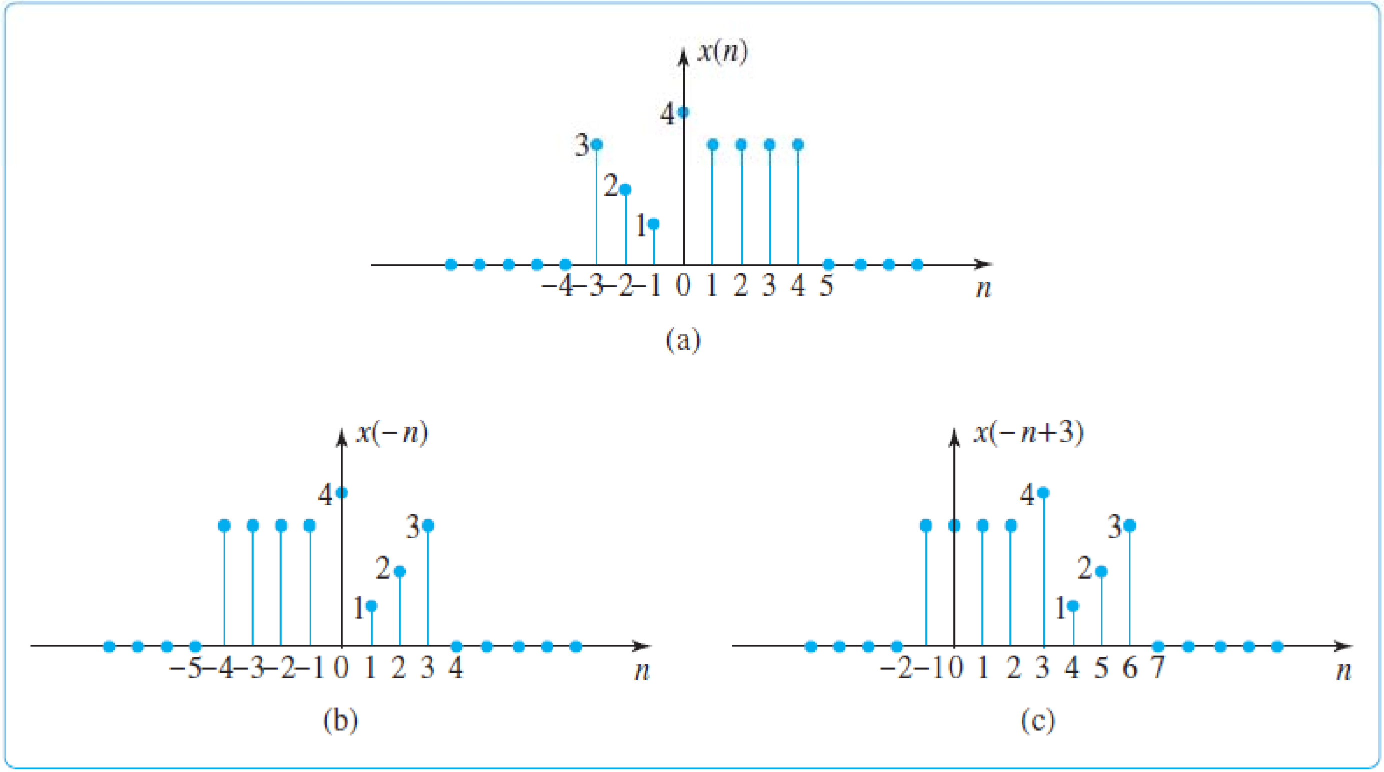
- (1) 독립 변수의 변환
  - 이산 신호의 지연과 선행




| 그림 2.8 | 이산 신호의 지연과 선행

# 2.1 이산 신호의 정의

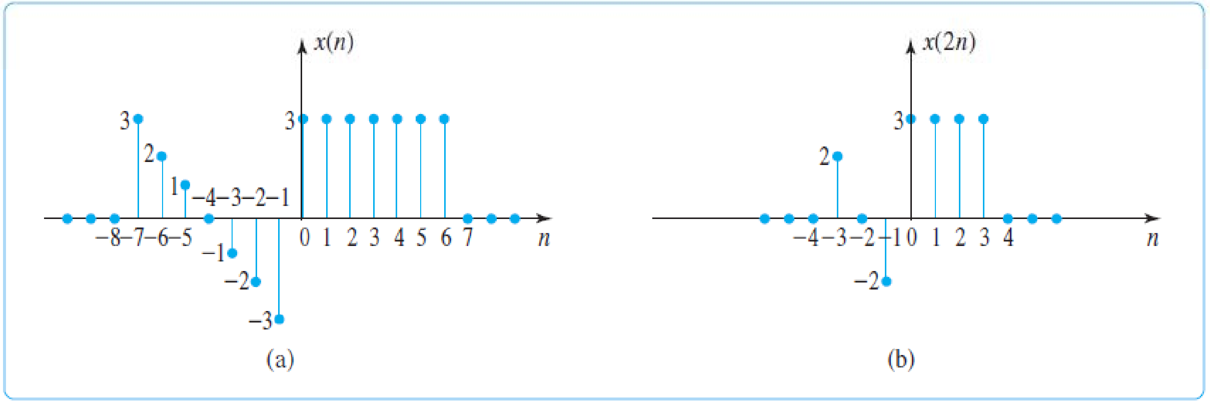
- 신호의 대칭 및 이동




| 그림 2.9 | 신호의 대칭 및 이동

# 2.1 이산 신호의 정의

- 신호의 다운 샘플링




| 그림 2.10 | 신호의 다운 샘플링

- (2) 신호의 더하기, 곱하기, 스케일링 동작
  - 더하기

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

  - 곱하기

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

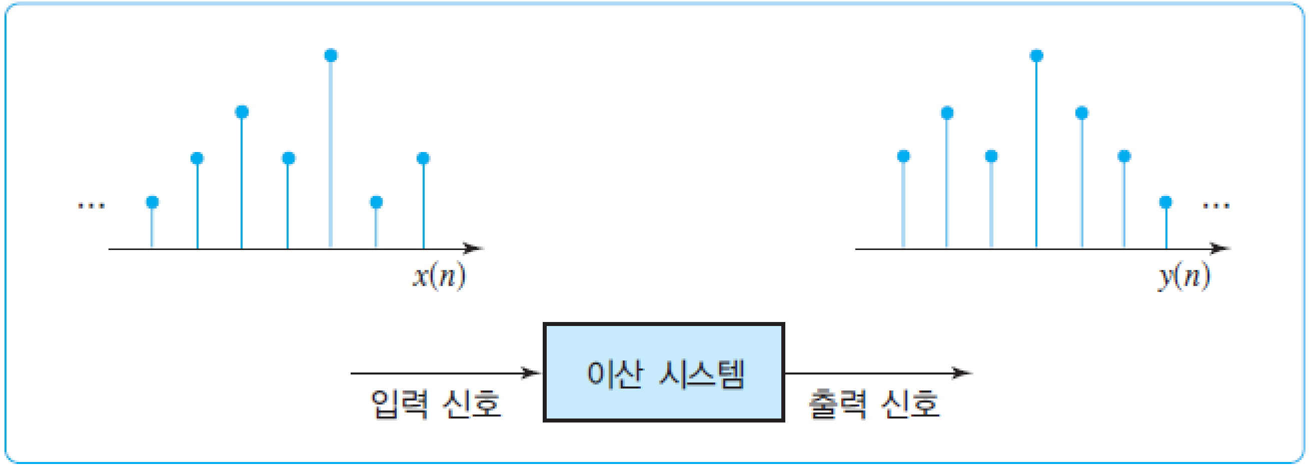
  - 스케일링

    $$y(n) = Ax\ (n)$$

# 2.2 이산 시스템

- 이산 시스템(discrete-time system)
  - 입력되는 이산 신호에 어떤 동작을 가하거나 연산을 수행하여 출력이라는 또 다른 이산 신호를 생성하는 모든 장치나 알고리즘

$$y(n) \equiv S\{x(n)\}$$

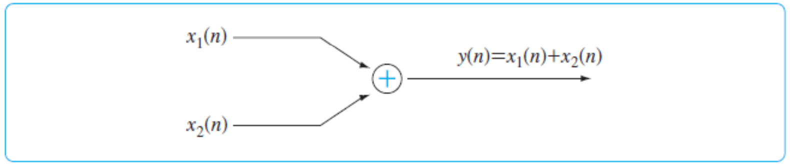



| 그림 2.11 | 이산 시스템의 개요도

# 2.2 이산 시스템

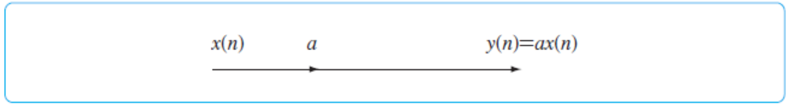
- 2.2.1 이산 시스템의 입출력 관계
  - 입출력 관계를 알기 위해서 반드시 시스템의 내부가 어떻게 구성되었는지를 알 필요는 없음

- 2.2.2 이산 시스템의 구성 요소
  - (1) 신호 덧셈기




┃ 그림 2.12 ┃ 덧셈기

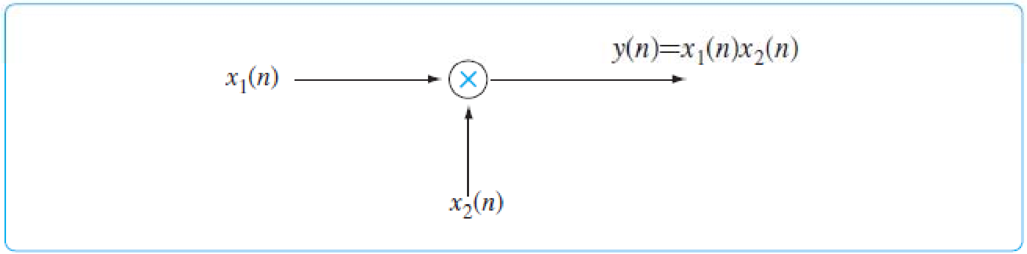
  - (2) 상수 곱셈기




┃ 그림 2.13 ┃ 상수 곱셈기

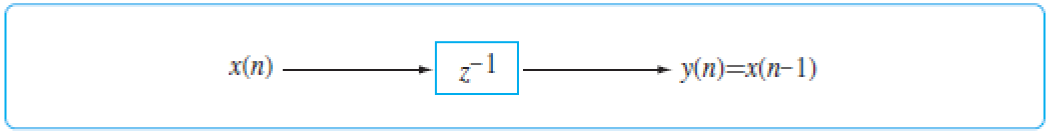
# 2.2 이산 시스템

◦ (3) 신호 곱셈기



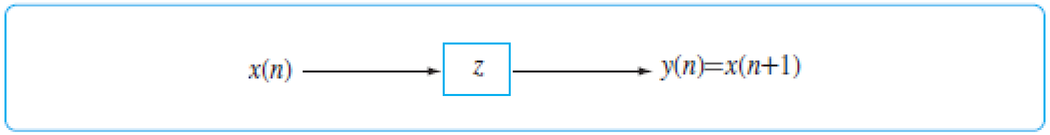

| 그림 2.14 | 신호 곱셈기

◦ (4) 단위 시간 지연기




| 그림 2.15 | 단위 시간 지연기

◦ (5) 단위 시간 선행기

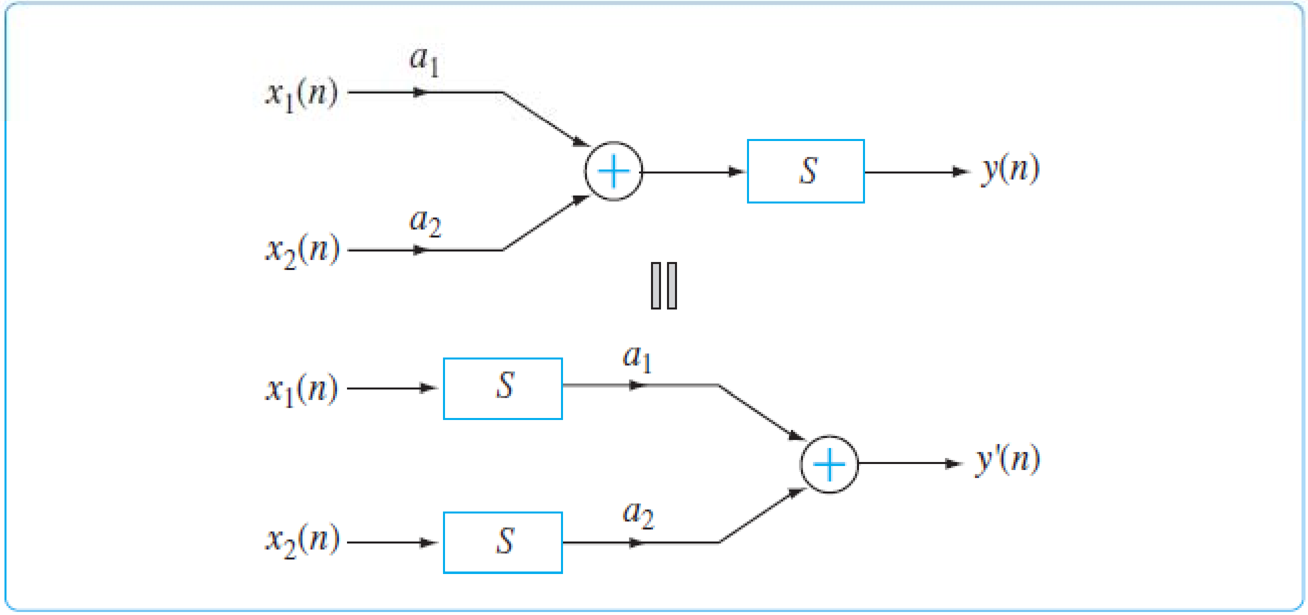



| 그림 2.16 | 단위 시간 선행기

# 2.2 이산 시스템

- 2.2.3 이산 시스템의 분류
  - (1) 선형(linear)과 비선형(nonlinear system) 시스템
    - 중첩 원리(superposition property)

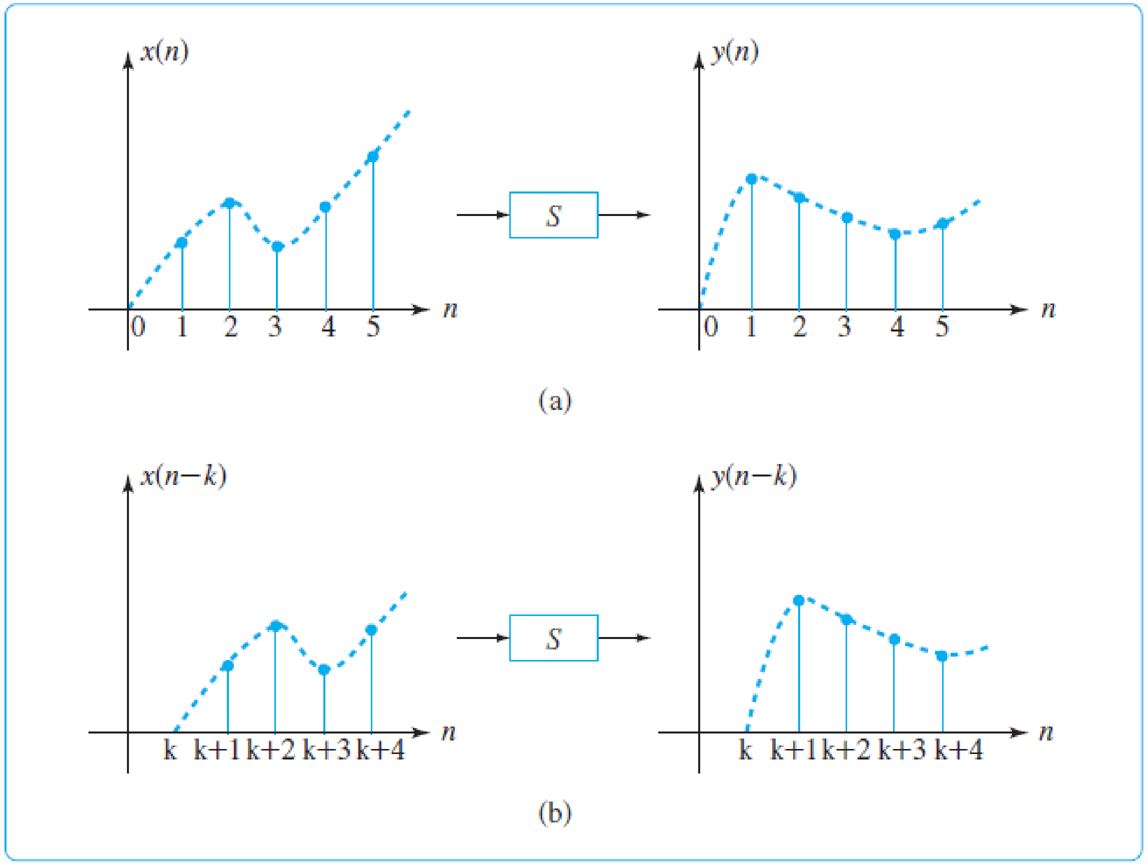
$$S\{a_1x_1(n) + a_2x_2(n)\} = a_1S\{x_1(n)\} + a_2S\{x_2(n)\}$$




| 그림 2.18 | 중첩의 원리(만일 $y(n) = y'(n)$이면 시스템은 선형)

# 2.2 이산 시스템

◦ (2) 시변(time-variant)과 시불변(time-invariant) 시스템




| 그림 2.19 | 시불변 시스템 입출력 관계

# 2.2 이산 시스템

- (3) 기억 시스템과 무기억 시스템
  - 기억 시스템(memory system)
    - 출력을 구하는데 현재의 입력 값뿐 아니라 과거나 미래의 입력 값이 필요

$$y(n) = 3x(n)$$

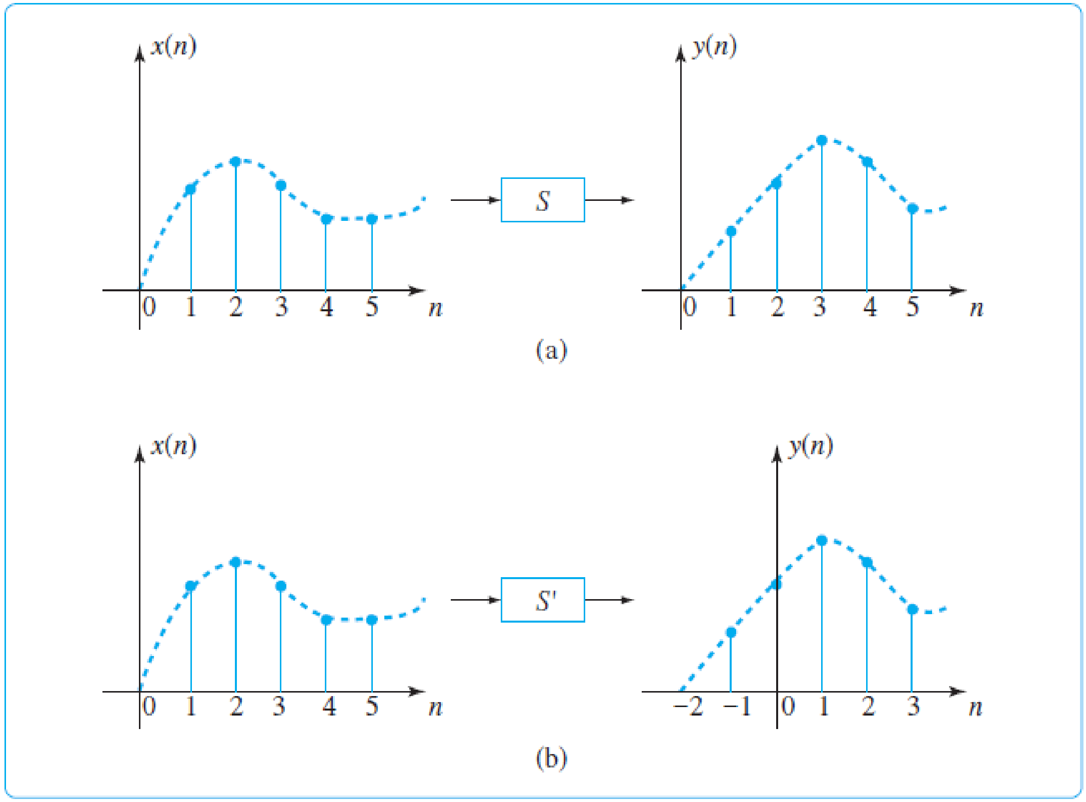
$$y(n) = nx(n) + 2x^2(n)$$

  - 무기억 시스템(memoryless system)
    - 이산 시스템의 출력 y(n)이 현재 시각 n에서의 입력신호 x(n)으로만 결정됨

$$y(n) = x(n-2) + 3x(n-5) + 2x(n+2)$$

# 2.2 이산 시스템

- (4) 인과(causal) 시스템과 비인과(noncausal) 시스템
  - 인과 시스템은 현재의 출력을 결정하기 위해 과거나 현재의 입력 값을 필요

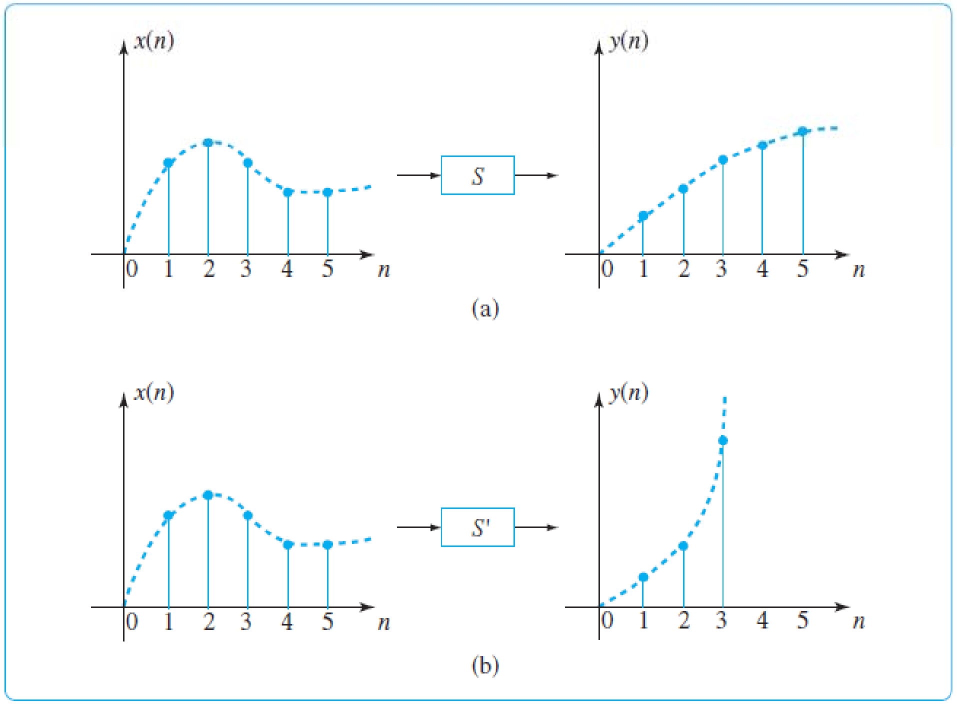



| 그림 2.20 | (a) 인과 시스템 (b) 비인과 시스템
(인과 시스템은 입력이 가해진 후에만 출력이 발생함)

# 2.2 이산 시스템

- (5) 안정(stable) 시스템과 비안정(unstable) 시스템
  - BIBO(bounded input bounded output stable system)
    - 입력의 값이 유한할 때 출력의 값도 항상 유한하다면 시스템은 BIBO 안정

$$|x(n)| \le M_x < \infty, |y(n)| \le M_y < \infty \quad \text{for all } n$$




| 그림 2.21 | (a) 안정 시스템 (b) 비안정 시스템

# 2.3 이산 선형 시불변 시스템의 해석

- 2.3.1 선형시스템의 해석
  - 방법 1: 차분 방정식(difference equation)
    - 이산 시스템의 입출력 관계를 수학식으로 모델링하면 차분 방정식으로 표현

$$y(n) = G\{y(n-1), y(n-2)\cdots,\ \cdots y(n-N), x(n), x(n-1), \cdots, x(n-M)\}$$

    - 선형 시불변 시스템(LTI)에서는 다음과 같은 N차 차분 방정식으로 표현됨
      - $a_k$ 와 $b_k$는 시스템의 특성에 따라 결정되는 매개 변수

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

# 2.3 이산 선형 시불변 시스템의 해석

- 방법 2: 입력을 기본 신호의 합으로 분해
  - 입력 신호 $x(n)$을 기본 신호 $x_k(n)$ 로 다음과 같이 분해

$$x(n) = \sum_k c_k x_k(n)$$

  - 기본 신호 $x_k(n)$의 시스템 응답 $y_k(n)$

$$y_k(n) = S\{x_k(n)\}$$

  - 시스템이 선형이기 때문에 다음의 관계식을 얻을 수 있음

$$\begin{aligned} y(n) &= S\{x(n)\} = S\{\sum_k c_k x_k(n)\} \\ &= \sum_k c_k S\{x_k(n)\} \\ &= \sum_k c_k y_k(n) \end{aligned}$$

  - 단위 임펄스 함수 $\delta(n)$를 기본 신호로 선택하였을 때 그 출력을 시스템의 임펄스 응답(impulse response)이라고 정의하고 $h(n)$으로 표현함

# 2.3 이산 선형 시불변 시스템의 해석

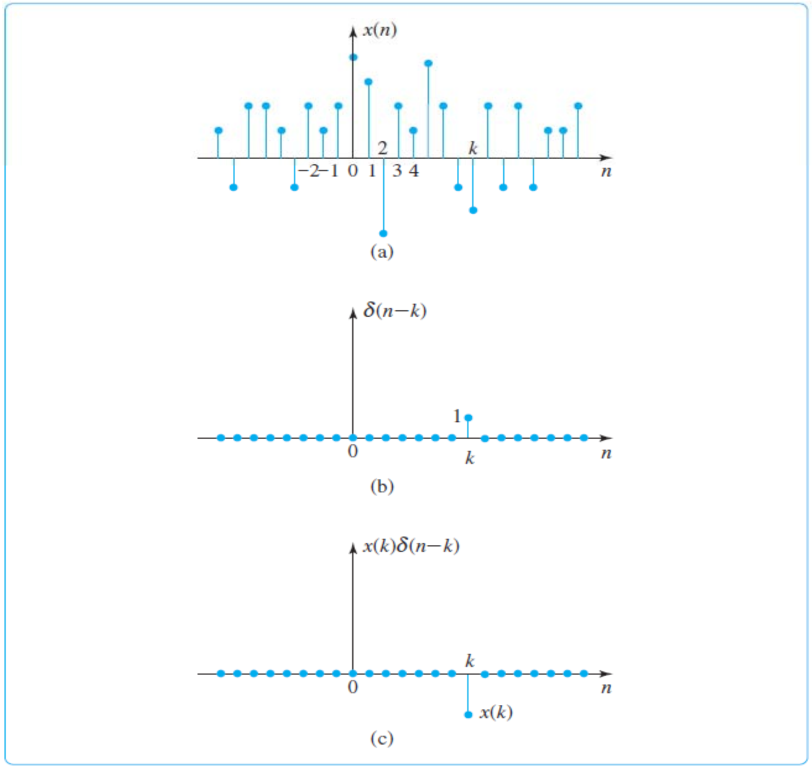
## 2.3.2 컨벌루션 합의 공식

- 단위 임펄스 함수를 기본 신호로 한 입력 신호 $x(n)$ 의 분해

$$x(n)\delta(n-k) = x(k)\delta(n-k)$$

$$x(n) = \cdots + x(-1)\delta(n+1) + x(0)\delta(n) + x(1)\delta(n-1) + \cdots$$

$$= \sum_{k=-\infty}^{\infty} x(k)\delta(n-k)$$




| 그림 2.22 | $x(n)$과 $\delta(n-k)$의 곱

# 2.3 이산 선형 시불변 시스템의 해석

- 컨벌루션 합
  - $\delta(n-k)$에 대한 시스템 응답을 $h(n,k)$라고 정의
  - 시스템이 시불변이라고 가정하였기 때문에 다음 식이 성립

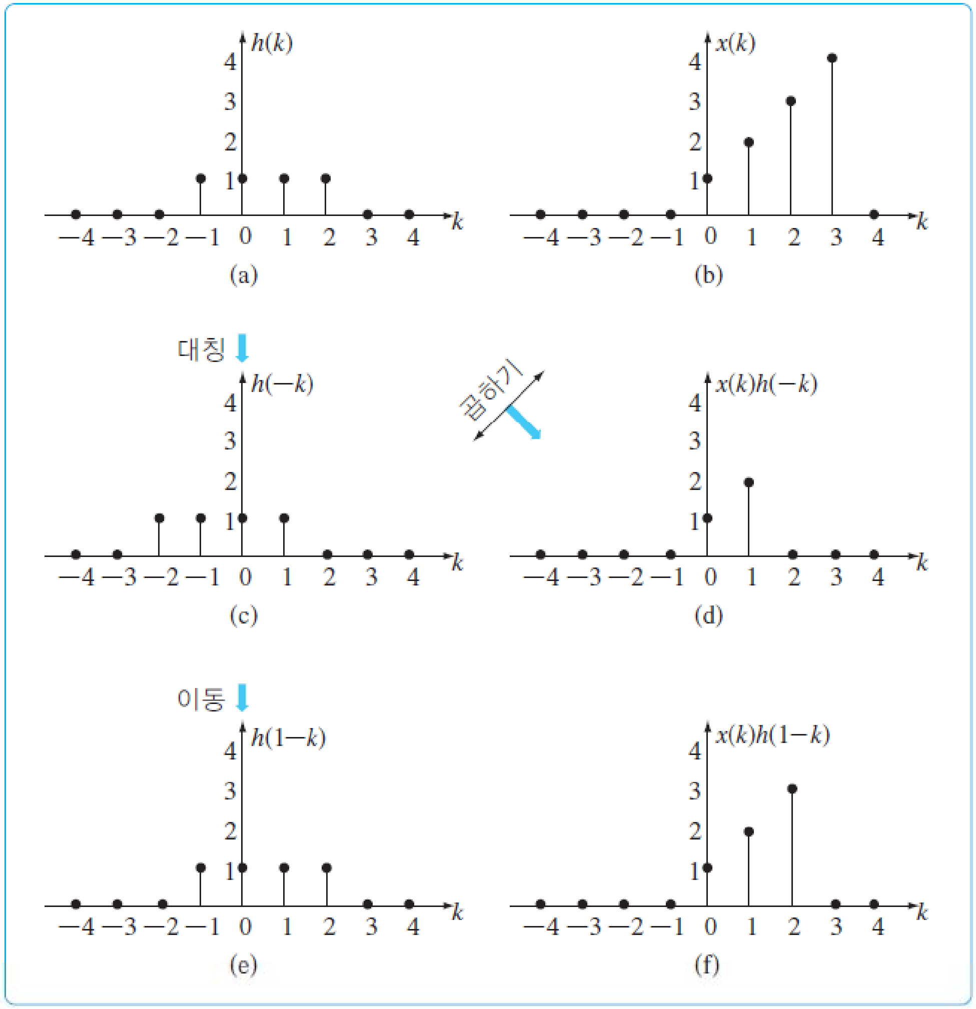
$$h(n,k) = S\{\delta(n-k)\} = h(n-k)$$

  - 입력 $x(n)$이 단위 임펄스 함수의 선형 조합으로 표현된다면 또한 시스템의 선형, 시불변 특성에 의하여

$$\begin{aligned} y_k(n) &= S\{x(n)\} = S\{\sum_{k=-\infty}^{\infty} x(k)\delta(n-k)\} \\ &= \sum_{k=-\infty}^{\infty} x(k) S\{\delta(n-k)\} \\ &= \sum_{k=-\infty}^{\infty} x(k) h(n-k) \\ &= x(n) * h(n) \end{aligned}$$

# 2.3 이산 선형 시불변 시스템의 해석

◦ 컨벌루션 연산의 예



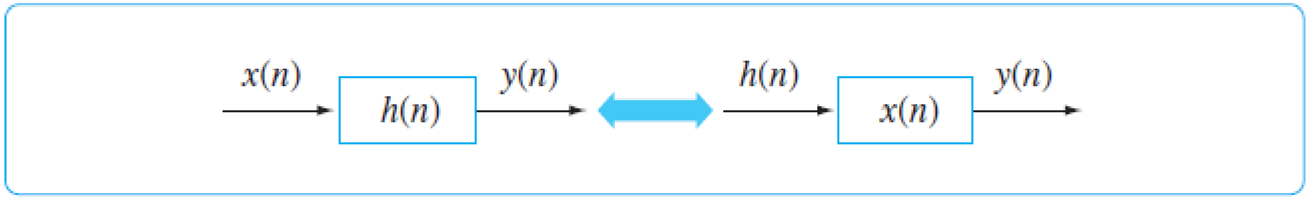

❙ 그림 2.23 ❙ 컨벌루션 연산의 예

# 2.3 이산 선형 시불변 시스템의 해석

## 2.3.3 컨벌루션의 성질

- (1) 교환 법칙(commutative law)

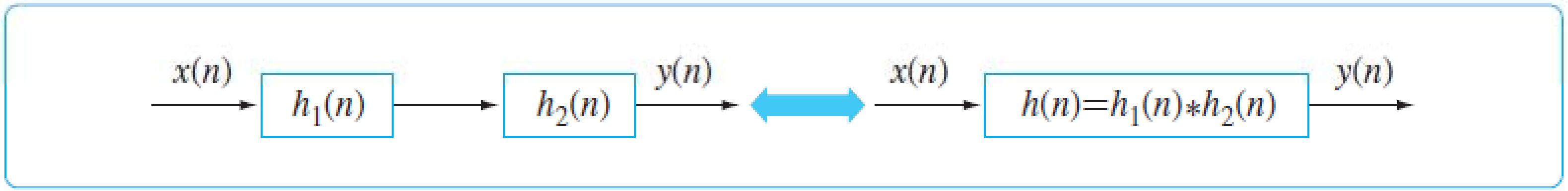
$$x(n) * h(n) = h(n) * x(n)$$




**| 그림 2.24 |** 컨벌루션의 교환 법칙

- (2) 결합 법칙(associative law)

$$\{x(n) * h_1(n)\} * h_2 = x(n) * \{h_1(n) * h_2(n)\}$$

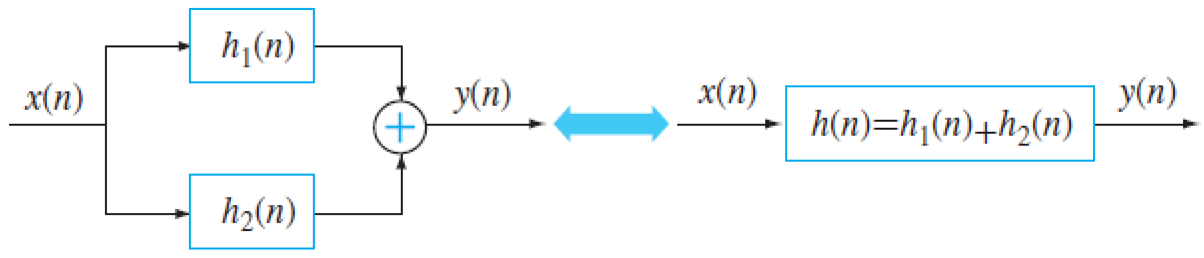



**| 그림 2.25 |** 컨벌루션의 결합 법칙

# 2.3 이산 선형 시불변 시스템의 해석

- (3) 배분 법칙(distributed law)

$$x(n) * \{h_1(n) + h_2(n)\} = x(n) * h_1(n) + x(n) * h_2(n)$$




| 그림 2.26 | 컨벌루션의 배분 법칙

# 2.3 이산 선형 시불변 시스템의 해석

## 2.3.4 인과적 선형 시불변 시스템

- $n=n_0$인 경우 시스템의 출력

$$y(n_0) = \sum_{k=-\infty}^{\infty} h(k)x(n_0 - k)$$

- $n_0$를 기준으로 미래, 과거와 현재의 입력 성분으로 나누어 정리하면

$$\begin{aligned} y(n_0) &= \sum_{k=-\infty}^{-1} h(k)x(n_0 - k) + \sum_{k=0}^{\infty} h(k)x(n_0 - k) \\ &= \{\cdots + h(-2)x(n_0 + 2) + h(-1)x(n_0 + 1)\} \\ &\quad + \{h(0)x(n_0) + h(1)x(n_0 - 1) + h(2)x(n_0 - 2) + \cdots\} \end{aligned}$$

- 시스템이 인과적이기 위해서는 다음을 만족하여야 함

$$h(n) = 0 \qquad n < 0$$

- 시스템이 인과적인 경우에 컨벌루션 합의 공식

$$\begin{aligned} y(n) &= \sum_{k=0}^{\infty} h(k)x(n-k) \\ &= \sum_{k=-\infty}^{n} x(k)h(n-k) \end{aligned}$$

# 2.3 이산 선형 시불변 시스템의 해석

- 2.3.5 안정 LTI 시스템
  - 선형 시불변 시스템인 경우 BIBO 안정하기 위한 조건

$$\sum_{k=-\infty}^{\infty} |h(k)| < \infty$$

  - 증명: 양변에 절대값을 씌우면

$$\begin{aligned} |y(n)| &= \left| \sum_{k=-\infty}^{\infty} h(k)x(n-k) \right| \\ &\le \sum_{k=-\infty}^{\infty} |h(k)||x(n-k)| \\ &\le M_x \sum_{k=-\infty}^{\infty} |h(k)| \end{aligned}$$

# 2.3 이산 선형 시불변 시스템의 해석

## 2.3.6 FIR 시스템과 IIR 시스템

- 유한 임펄스 응답(finite-duration impulse response)
  - 어느 유한한 구간 외에는 임펄스 응답이 모두 0의 값을 갖는 시스템
  - 시스템이 인과적이라면 h(n)=0, n<0 과 n≥M으로 정의됨

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k)$$

- 무한 임펄스 응답(infinite-duration impulse response)
  - 무한한 구간에서 임펄스 응답이 존재하는 시스템
  - IIR 선형 시불변 시스템

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k)$$

# 2.4 차분 방정식으로 표현되는 시스템

## 2.4.1 차분 방정식의 표현법

- 재귀 시스템(recursive system)
  - 과거의 출력 값이 현재의 출력 값을 구하는 데 이용되는 시스템

$$y(n) = G\{y(n-1), y(n-2), \ldots, y(n-N), x(n), x(n-1), \ldots, x(n-M)\}$$

  - 입력 신호 x(n)의 n+1개의 평균 값을 구하는 시스템

$$y(n) = \frac{1}{n+1}\sum_{k=0}^{n} x(k) \qquad n = 0,1,\ldots$$

$$(n+1)y(n) = \sum_{k=0}^{n-1} x(k) + x(n)$$

$$= ny(n-1) + x(n)$$

$$y(n) = \frac{n}{n+1}y(n-1) + \frac{1}{n+1}x(n)$$

# 2.4 차분 방정식으로 표현되는 시스템

- 비재귀 시스템(nonrecursive system)
  - 과거의 출력 값이 아닌 입력 값에만 의존하는 시스템

$$y(n) = G\{x(n), x(n-1), ..., x(n-M)\}$$

  - 선형 시불변 FIR 시스템은 비재귀 시스템이 됨

$$\begin{aligned} y(n) &= \sum_{k=0}^{M} h(k)\, x(n-k) \\ &= h(0)x(n) + h(1)x(n-1) + \cdots + h(n)x(n-M) \\ &= G\{x(n), x(n-1), \cdots, x(n-M)\} \end{aligned}$$

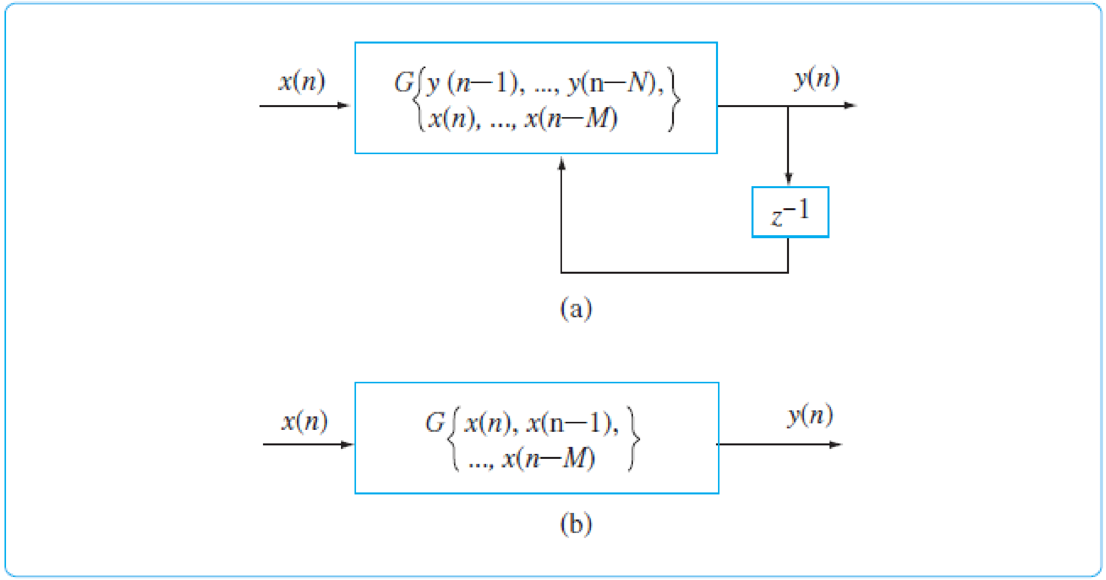
  - 선형 시불변 IIR 시스템은 재귀 시스템이 됨

# 2.4 차분 방정식으로 표현되는 시스템

- 모든 선형 시불변 시스템은 아래와 같은 차분 방정식 형태로 표현 가능
  - $a_k$와 $b_k$는 시스템의 특성에 의해서 결정되는 임의의 상수

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{k=0}^{M} b_k x(n-k), \; a_0 = 1$$




| 그림 2.27 | (a) 재귀 시스템과 (b) 비재귀 시스템의 기본 형태

# 2.4 차분 방정식으로 표현되는 시스템

- 2.4.2 상계수 차분 방정식으로 표현되는 선형 시불변 시스템
  - 재귀 시스템을 표현하는 간단한 1차 차분 방정식
    - 신호 x(n)이 n>0일 때 시스템에 입력되고, 초기 조건은 y(-1)이라고 가정

$$y(n) = 2y(n-1) + x(n)$$

$$y(0) = 2y(-1) + x(0)$$

$$y(1) = 2y(0) + x(1) = 2^2 y(-1) + 2x(0) + x(1)$$

$$y(2) = 2y(1) + x(2) = 2^3 y(-1) + 2^2 x(0) + 2x(1) + x(2)$$

$$\vdots$$

$$\begin{aligned} y(n) &= 2y(n-1) + x(n) \\ &= 2^{n+1} y(-1) + 2^n x(0) + 2^{n-1} x(1) + \cdots + 2x(n-1) + x(n) \end{aligned}$$

$$y(n) = 2^{n+1} y(-1) + \sum_{k=0}^{n} 2^k x(n-k), \ n \geq 0$$

# 2.4 차분 방정식으로 표현되는 시스템

- 제로 상태 응답(zero state response)
  - 입력 x(n)에 의해 생성되는 출력 성분
  - 시스템의 초기 조건이 모두 0이라면 첫 번째 항이 없어짐

$$y_{zs}(n) = \sum_{k=0}^{n} 2^k x(n-k),\ n \geq 0$$

- 제로 입력 응답(zero input response)
  - 시스템의 초기 조건에 의해서 생성되는 출력 성분
  - 신호 x(n)이 시스템에 입력되지 않는다면, 즉 입력 신호가 모두 0일 때

$$y_{zi}(n) = 2^{n+1} y(-1),\ n \geq 0$$

- 차분 방정식에 의해 표현되는 시스템이 선형 시스템이 되기 위해서는 전체해(total solution)가 제로 상태 응답과 제로 입력 응답의 합으로 이루어 져야 함

$$y(n) = y_{zi}(n) + y_{zs}(n)$$

# 2.4 차분 방정식으로 표현되는 시스템

- 2.4.3 선형 상계수 차분 방정식의 해
  - 상수 계수를 갖는 선형 차분 방정식의 해
    - 차분 방정식의 해는 균일해 $y_h(n)$과 특수해 $y_p(n)$의 합으로 이루어짐

$$y(n) = y_h(n) + y_p(n)$$

    - 균일해 $y_h(n)$는 입력 $x(n)=0$(입력이 없는 경우)이라고 가정하여 구함
    - 특수해 $y_p(n)$는 주어진 입력 $x(n)$에 대한 출력 성분임

# 2.4 차분 방정식으로 표현되는 시스템

- 균일해(homogeneous solution)
  - 입력 신호 x(n)이 0이라고 가정하면

$$\sum_{k=0}^{N} a_k y(n-k) = 0$$

  - $y_h(n) = \lambda^n$이라 가정하면 다음과 같음

$$\sum_{k=0}^{N} a_k \lambda^{n-k} = 0$$

$$\lambda^{n-N}(a_0\lambda^N + a_1\lambda^{N-1} + a_2\lambda^{N-2} + \cdots + a_{N-1}\lambda + a_N) = 0$$

  - **특성 방정식**(characteristic equation)
    - 위의 식의 괄호 안의 다항식
    - 일반적으로 N개의 근 $\lambda_1, \lambda_2, \ldots, \lambda_N$이 존재
    - N개의 근이 서로 다른 실수일 때

$$y_h(n) = C_1\lambda_1^n + C_2\lambda_2^n + \ldots + C_N\lambda_N^n$$

여기서 $C_1, C_2, \cdots, C_N$는 가중 계수이며 시스템의 초기 조건에 의해 구해짐

    - 중복근일 경우

$$y_h(n) = C_1\lambda_1^n + C_2 n\lambda_1^n + \cdots + C_{P_1} n^{P_1-1}\lambda_1^n + C_{P_1+1}\lambda_{p_1+1}^n + \cdots + C_N\lambda_N$$

# 2.4 차분 방정식으로 표현되는 시스템

- [예제 2.11]

$$y(n)+0.5y(n-1)=0, \quad y(-1)=-1$$

- 균일해가 $y_k(n)=\lambda^n$이라고 가정

$$\lambda^n+0.5\lambda^{n-1}=0$$

$$\lambda^{n-1}(\lambda+0.5)=0$$

$$\lambda=-0.5$$

$$y_h(n)=C\lambda^n=C(-0.5)^n$$

- 이때 y(-1)=-1이므로 이를 만족하려면 C=0.5가 되어야 함

# 2.4 차분 방정식으로 표현되는 시스템

- 특수해(particular solution)
  - 입력 x(n)에 대한 차분 방정식의 출력 성분
  - [예제 2.12]

$$y(n)+0.5y(n-1)=x(n)$$

$$x(n)=u(n)$$

- 입력이 n≥0에서 상수이므로 특수해의 형태도 상수라고 가정(educated guess)

$$y_p(n)=Ku(n)$$ 표 2.1 참조

$$Ku(n)+0.5Ku(n-1)=u(n)$$

- n≥1인 경우 K+0.5K=1이 되고 따라서 K=1/1.5

$$y_p(n)=\frac{1}{1+a_1}u(n)$$

▶표 2.1 입력 신호에 대한 특수해의 예

| 입력신호 $x(n)$ | 특수해 $y_p(n)$ |
|---|---|
| $A$(상수) | $K$ |
| $AM^n$ | $KM^n$ |
| $An^M$ | $K_o n^M + K_1 n^{M-1} + \cdots + K_M$ |
| $A^n n^M$ | $A^n (K_0 n^M + K_1 n^{M-1} + \cdots + K_M)$ |
| $A\cos\omega_0 n$ | $K_1 \cos\omega_0 n + K_2 \sin\omega_0 n$ |
| $A\sin\omega_0 n$ | $K_1 \cos\omega_0 n + K_2 \sin\omega_0 n$ |

# 2.4 차분 방정식으로 표현되는 시스템

- 전체해(total solution)

$$y(n) = y_h(n) + y_p(n)$$

- [예제 2.13]

$$y(n) - 2y(n-1) = 2u(n), \quad y(-1) = 1$$

  - x(n)=0일 때 균일해가 $y_h(n)=\lambda^n$이라고 가정하고 차분 방정식이 대입

$$\lambda^n - 2\lambda^{n-1} = 0$$

$$\lambda^{n-1}(\lambda - 2) = 0$$

  - 특성 방정식은 λ–2=0, λ=2가 되므로 균일해는

$$y_h(n) = C(2)^n$$

# 2.4 차분 방정식으로 표현되는 시스템

- 단위 계단 함수 u(n)은 n≥0일 때 모두 1이므로 상수라 가정하면 표 2.1로부터 특수해는 다음과 같음

$$y_p(n) = Ku(n)$$

- 차분 방정식에 대입하여 n≥1일 경우 K값을 구하면

$$Ku(n) + 2Ku(n-1) = 2u(n)$$

$$K + 2K = 2, \quad \mathrm{K} = \frac{2}{3}$$

- 따라서 특수해는 다음과 같음

$$y_p(n) = \frac{2}{3}u(n)$$

# 2.4 차분 방정식으로 표현되는 시스템

- 따라서 균일해와 특수해를 더한 전체해는 다음과 같음

$$y(n) = \frac{10}{3}(2)^n + \frac{2}{3}, \quad n \ge 0 \qquad (1)$$

- 주어진 차분 방정식에 n=0을 대입하면, 초기 조건 y(−1)=1을 이용하면

$$y(0) - 2y(-1) = 2$$

- y(0)=4를 얻을 수 있고, C를 구하기 위해 위의 식 (1)에 n=0을 대입하면

$$y(0) = C + \frac{2}{3}$$

$$C = 4 - \frac{2}{3} = \frac{10}{3}$$

- 따라서 차분 방정식의 해는 다음과 같음

$$y(n) = C(2)^n + \frac{2}{3}, \quad n \ge 0$$

# 2.4 차분 방정식으로 표현되는 시스템

- 2.4.4 차분 방정식에서 시스템 임펄스 응답 구하기
  - 차분 방정식에서 입력 신호 x(n)을 단위 임펄스 함수 δ(n)으로 대체하고 시스템의 모든 초기 조건을 0이라 가정
  - 간단한 재귀 시스템을 예로 들어보면,

$$y(n) = 2y(n-1) + x(n)$$

    - 모든 초기 조건이 0이라고 가정했으므로 해는 제로 상태 응답만으로 구성됨

$$y(n) = \sum_{k=0}^{n} 2^k x(n-k), \quad n \geq 0$$

    - 여기에 입력 신호 x(n)을 δ(n)으로 대입하면 다음과 같이 됨

$$y(n) = \sum_{k=0}^{n} 2^k \delta(n-k)$$
$$= 2^n, \quad n \geq 0$$

    - 결국 임펄스 응답은 다음과 같음

$$h(n) = 2^n u(n)$$