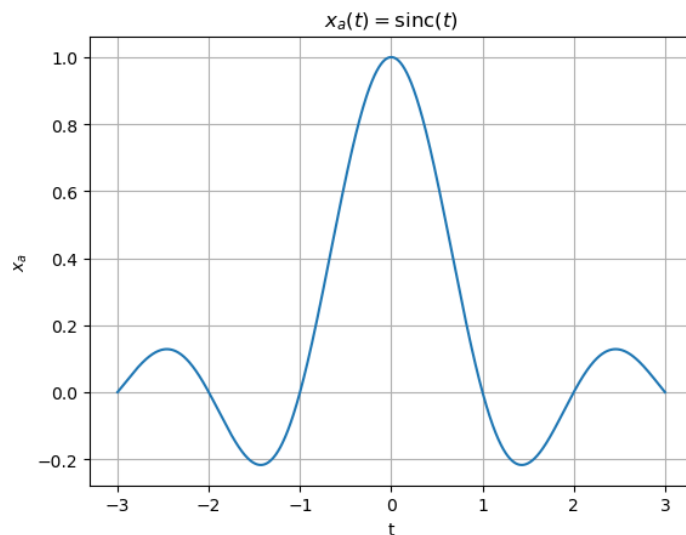


Lecture 04

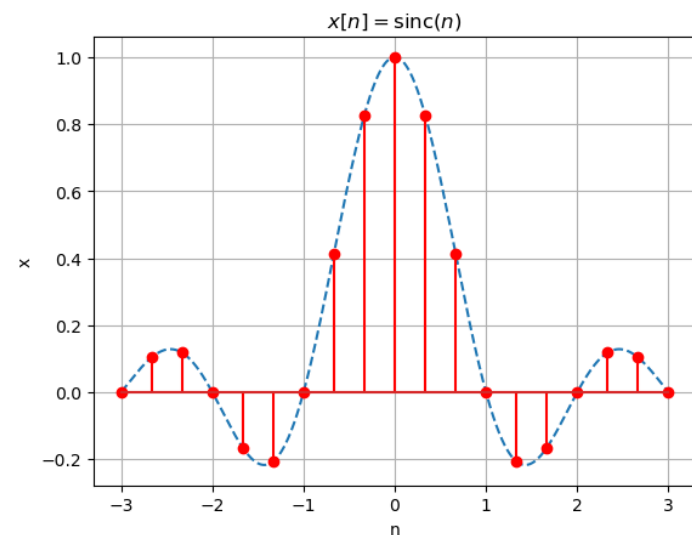
이산 신호 및 시스템

이산 신호

- 특정한 시각에서만 값을 갖는 신호로 정의됨
- 연속 신호의 샘플링(**sampling**) 과정을 거쳐 얻게 됨
 - $x[n] = x_a(nT_s)$
 - $x[n]$ 은 $x_a(t)$ 를 T_s 간격으로 샘플링 했을 때 n 번째 샘플을 의미함



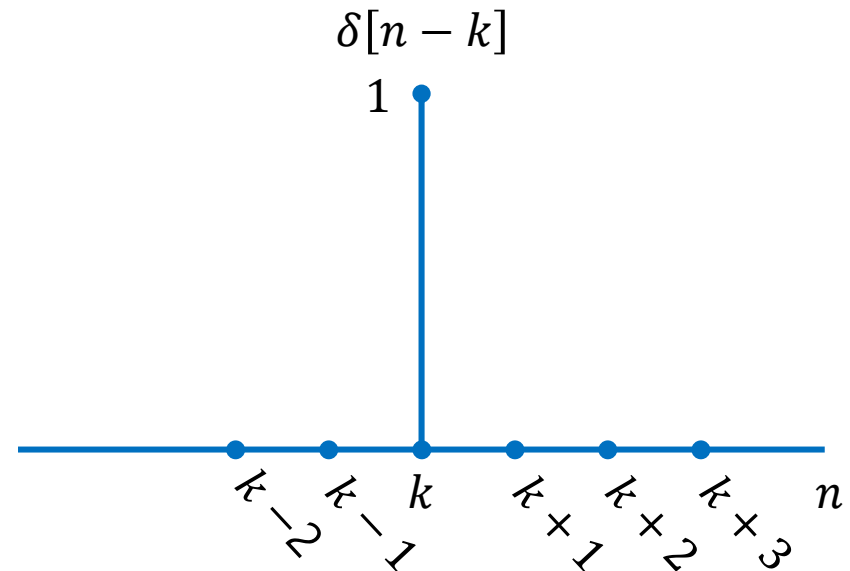
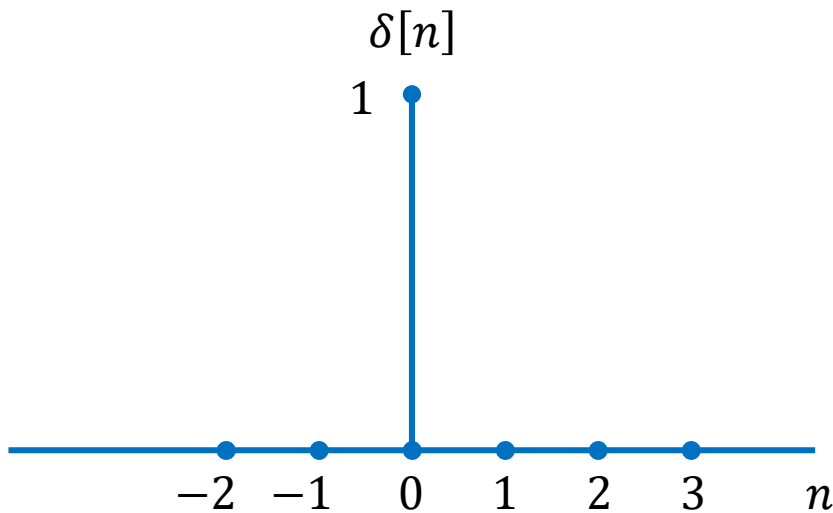
샘플링
→



기본 이산 신호

- 단위 임펄스 함수(**Unit Impulse Function**)

- 정의 : $\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$
- n 이 정수가 아닌 경우 함수 값이 정의되지 않음



기본 이산 신호

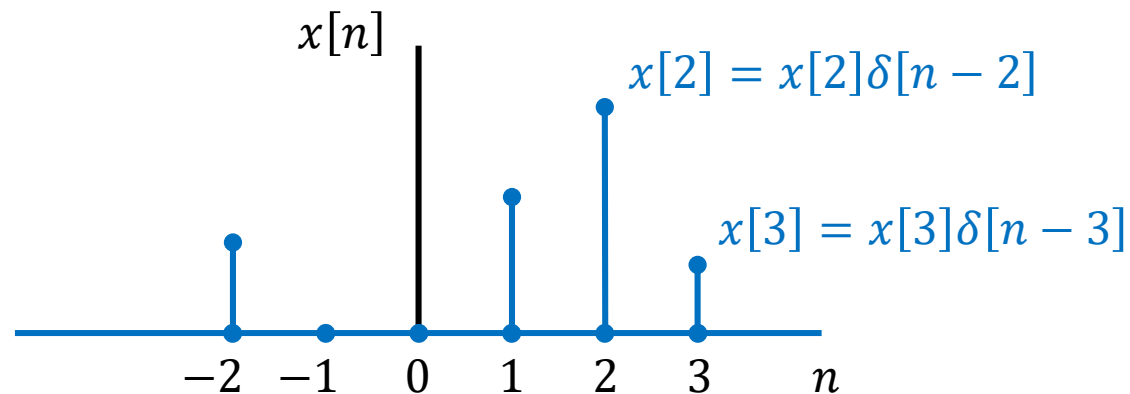
- 단위 임펄스 함수(**Unit Impulse Function**)

- 함수 $\delta[n]$ 는 임의의 이산 신호 $x[n]$ 에 대하여 다음과 같은 성질을 가짐

$$x[n]\delta[n - k] = x[k]\delta[n - k]$$

- 임의의 이산 신호 $x[n]$ 을 임펄스 함수를 가지고 표현할 수 있음

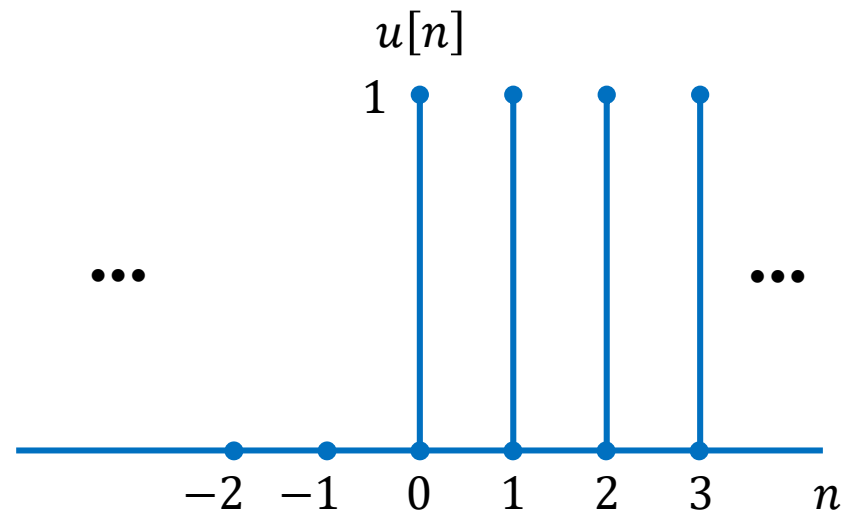
$$x[n] = \dots + x[-1]\delta[n + 1] + x[0]\delta[n] + x[1]\delta[n - 1] + \dots = \sum_{k=-\infty}^{\infty} x[k]\delta[n - k]$$



기본 이산 신호

- 단위 계단 함수(**Unit Step Function**)

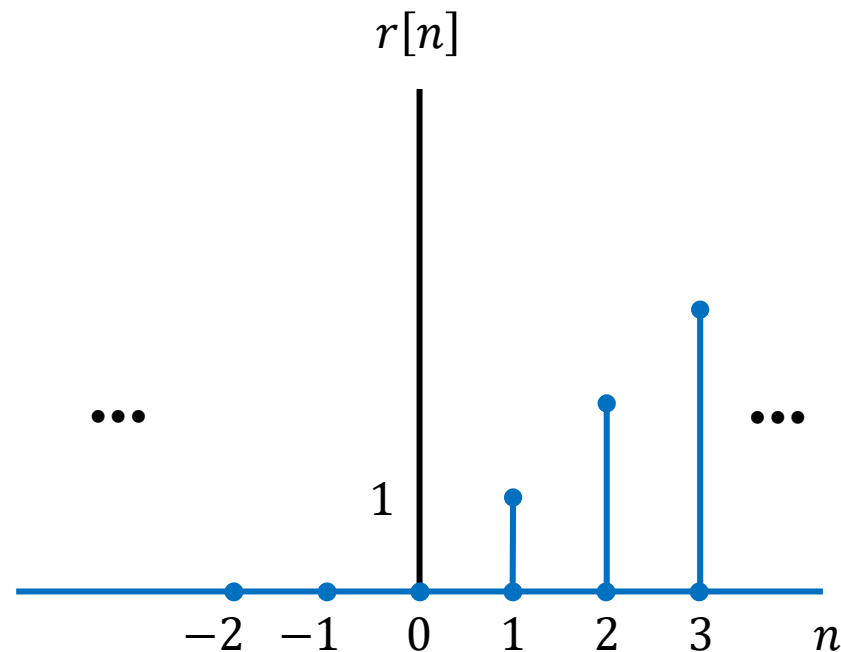
- 정의 : $u[n] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$
 - $u[n] - u[n - 1] = \delta[n]$
 - $\sum_{k=0}^{\infty} \delta[n - k] = u[n]$



기본 이산 신호

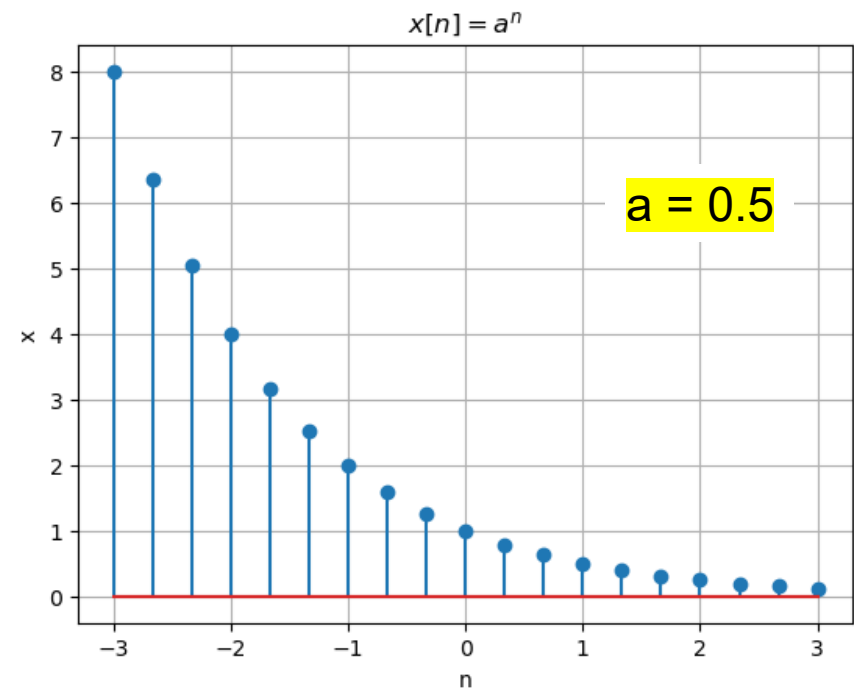
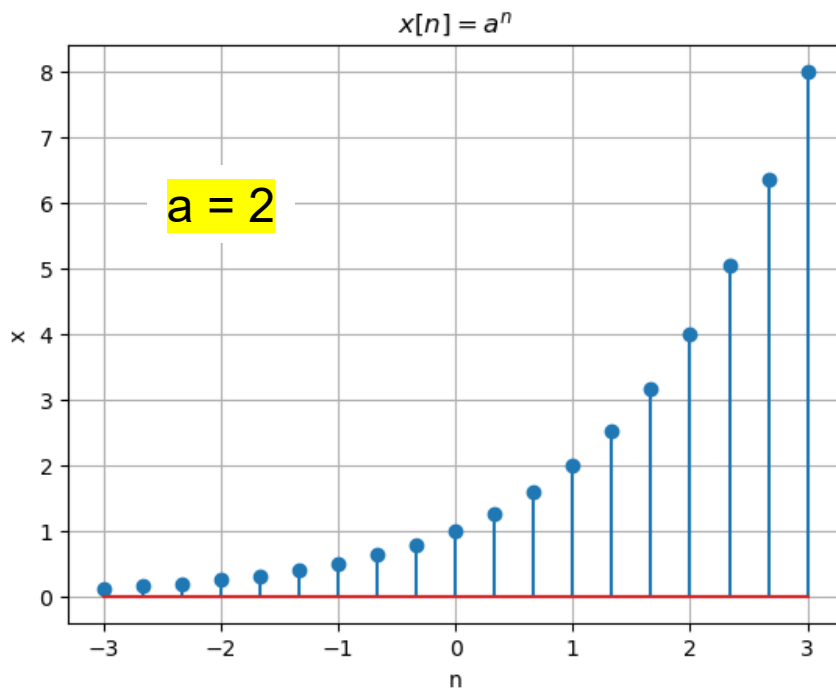
- 단위 램프 함수(**Unit Ramp Function**)

- 정의 : $r[n] = \begin{cases} n & n \geq 0 \\ 0 & n < 0 \end{cases}$



기본 이산 신호

- 지수 함수(**Exponential Function**)
 - 정의 : $x[n] = a^n, a > 0$



이산 신호 분류

- 에너지(**energy**) 신호와 전력(**power**) 신호

- 에너지 : 신호의 크기를 제공한 후 모두 정수 n 에 대하여 더한 값

$$E = \sum_{n=-\infty}^{\infty} |x[n]|^2$$

- 에너지 신호 : **유한** E 값을 갖고 있음
- 신호의 **평균 전력**은 다음과 같이 정의됨

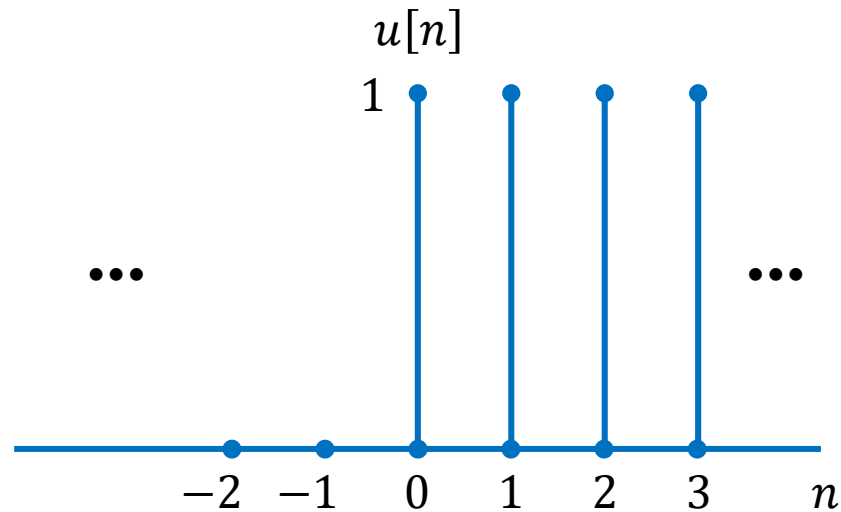
$$P = \lim_{N \rightarrow \infty} \frac{1}{2N + 1} \sum_{n=-N}^N |x[n]|^2$$

- 전력 신호 : **유한** P 값을 갖고 있음

에너지 E	전력 P
유한	0
무한	유한 혹은 무한

이산 신호 분류

- 에너지(**energy**) 신호와 전력(**power**) 신호
 - 예, $x[n] = u[n]$



이산 신호 분류

- 에너지(**energy**) 신호와 전력(**power**) 신호

- 예, $x[n] = u[n]$

- 에너지

$$E = \sum_{n=-\infty}^{\infty} |x[n]|^2 = \sum_{n=0}^{\infty} |1|^2 = \infty$$

- $x[n]$ 는 에너지 신호 아님

- 전력

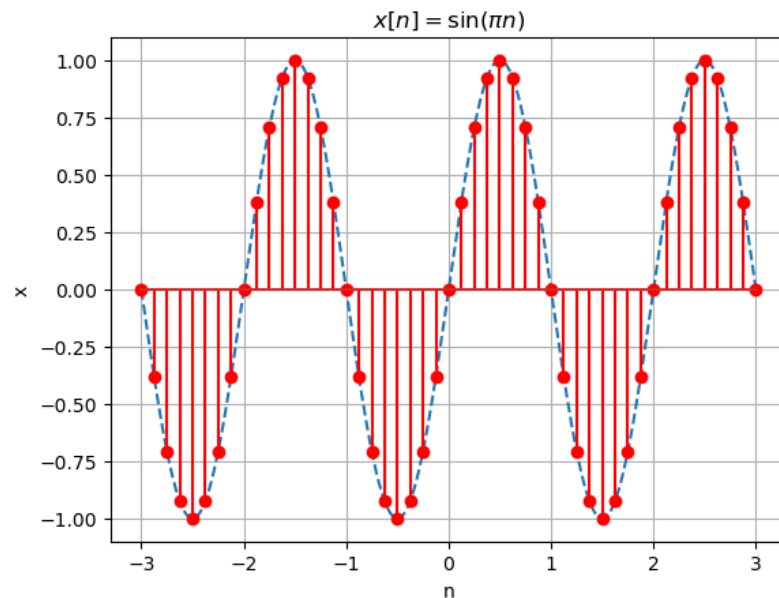
$$P = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N |x[n]|^2 = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=0}^N |1|^2 = \lim_{N \rightarrow \infty} \frac{N+1}{2N+1} = \frac{1}{2}$$

- $x[n]$ 는 전력 신호임

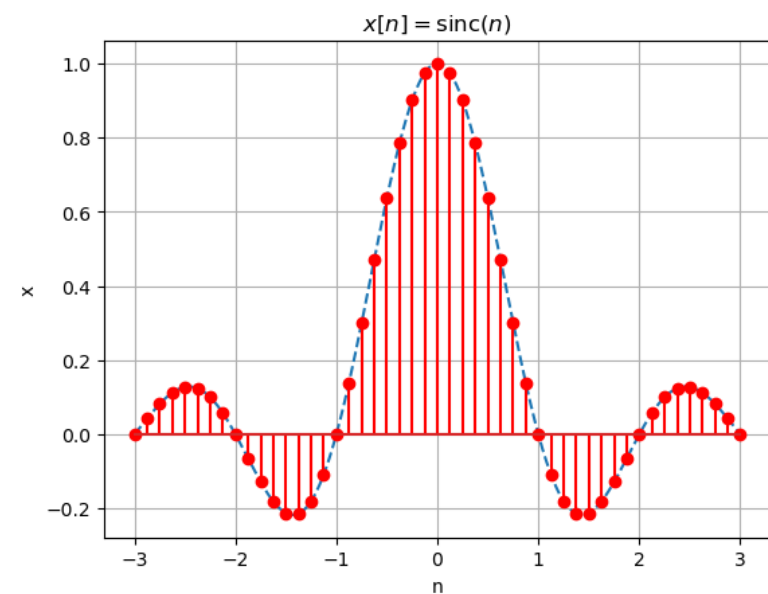
이산 신호 분류

- 주기 신호와 비주기 신호
 - 주기 신호 정의 : $x[n] = x[n + N]$

주기 신호

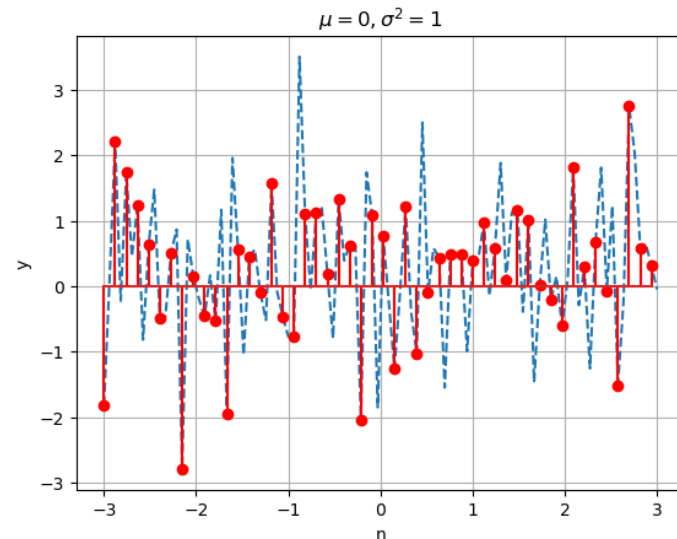
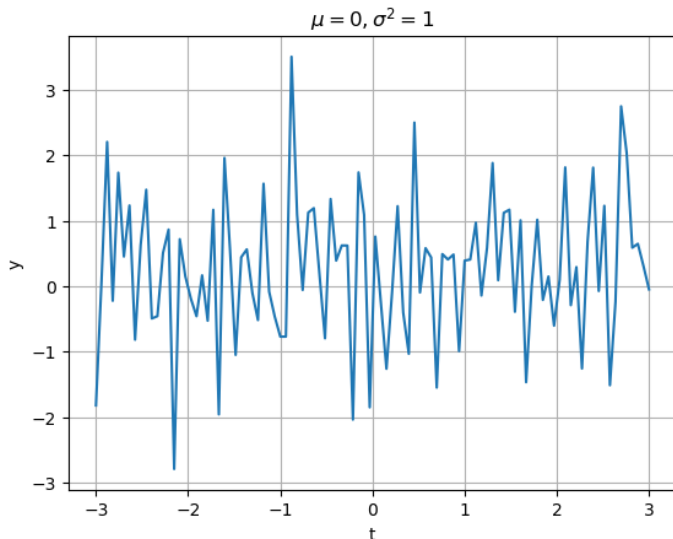


비주기 신호



이산 신호 분류

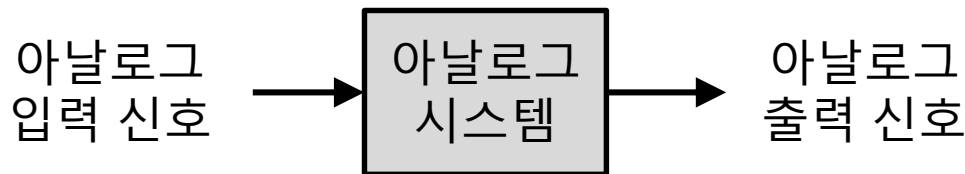
- 결정적(**deterministic**) 신호와 불규칙(**random**) 신호
 - 결정적 신호 : 수학식으로 명확하게 표현되거나 신호의 값을 **완전히** 예측할 수 있음
 - 불규칙 신호 : 신호의 값을 예측할 수 없음
 - 예, 백색 가우스 잡음(white Gaussian noise)



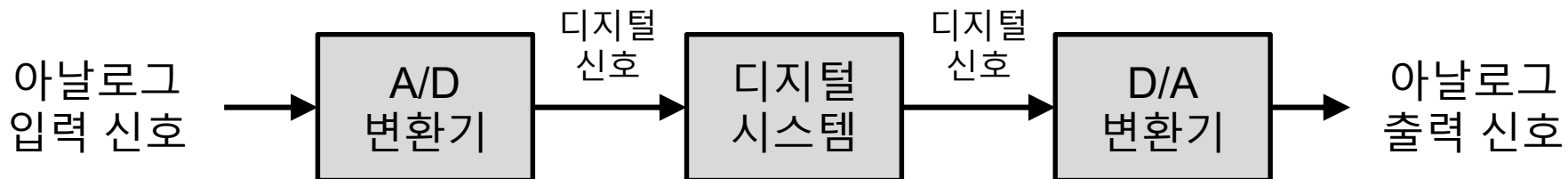
디지털 신호와 시스템

- 디지털 신호는 자연적으로 존재하는 신호가 아니라 아날로그 신호, 즉 **연속 신호**로부터 인위적인 작업을 통해 만들어짐

아날로그(연속) 시스템



디지털 처리를 위한 아날로그 신호의 변환



디지털 신호와 시스템

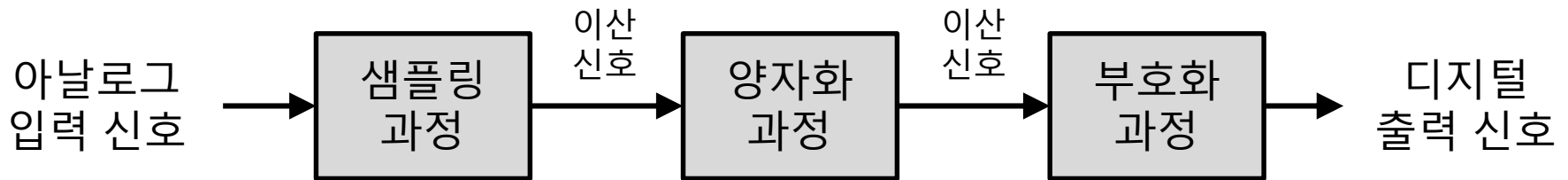
- 디지털 신호 처리 시스템이 필요한 이유
 - 컴퓨터 프로그램에서 처리가 가능함
 - 손쉽게 수정이 가능함
 - 다양한 저장 매체에 쉽게 저장할 수 있음
 - 이동이 용이함
 - 복잡한 시스템을 쉽게 구현할 수 있음
 - ...
- 디지털 시스템은 **음성** 신호 처리, **영상** 신호 처리 등 여러 분야에 널리 쓰임

디지털 신호와 시스템



- 아날로그-디지털 변환(**ADC**: analog-to-digital conversion)
 - 샘플링
 - 양자화
 - 부호화

아날로그-디지털 변환기

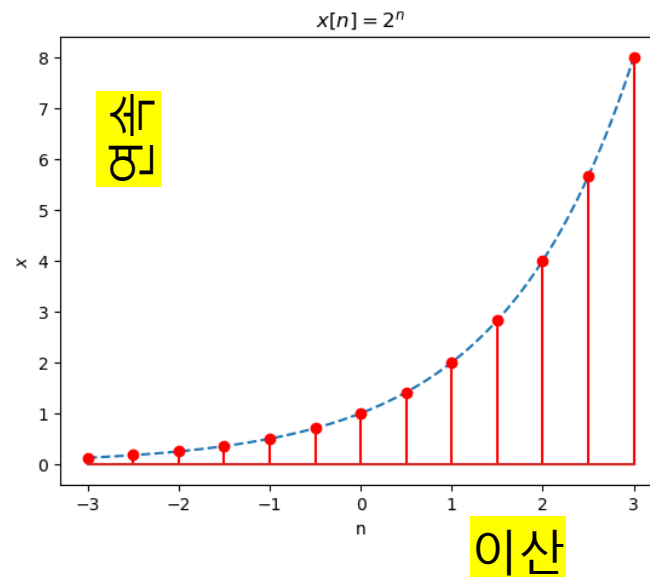
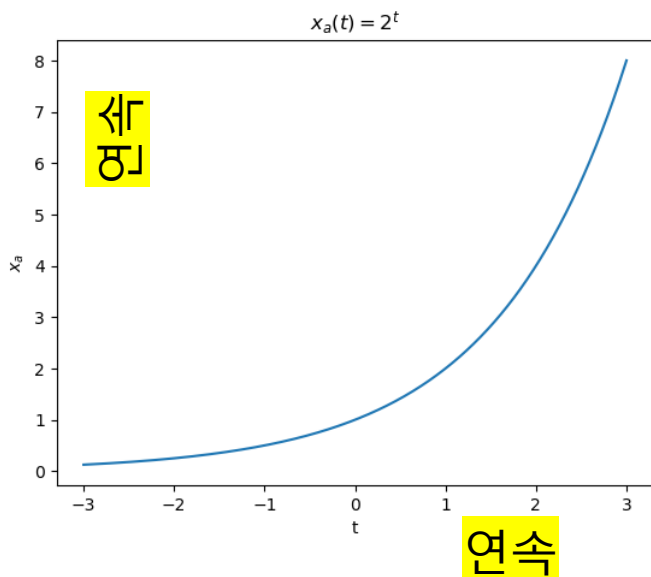


디지털 신호와 시스템

- 아날로그-디지털 변환(**ADC**: analog-to-digital conversion)

- **샘플링** : 일정한 간격을 가지고 규칙적으로 연속 신호로부터 샘플을 취함으로써 **연속 신호** $x_a(t)$ 를 **이산 신호** $x[n] = x_a(nT)$ 로 변환하는 과정임

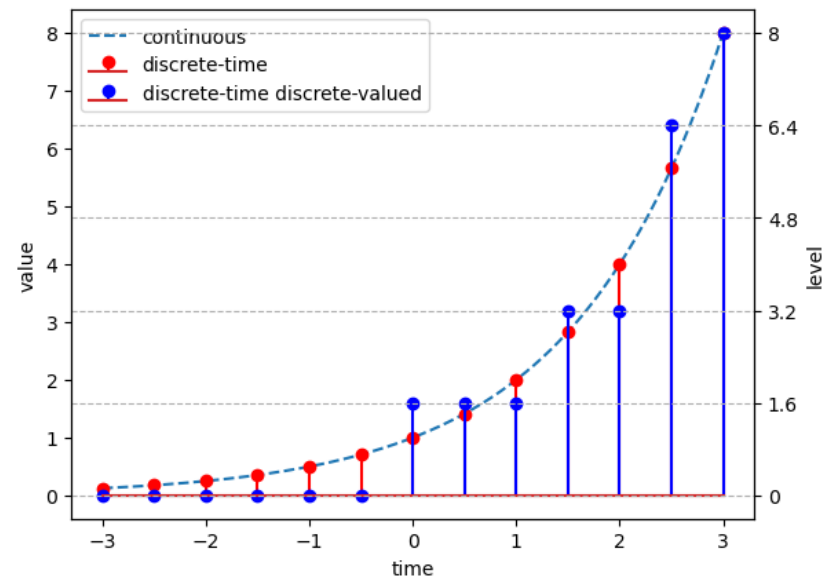
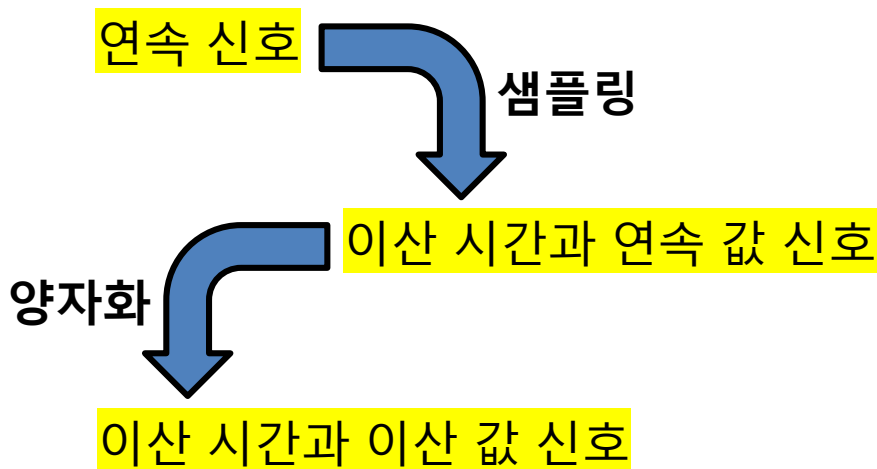
discrete-time signal



디지털 신호와 시스템

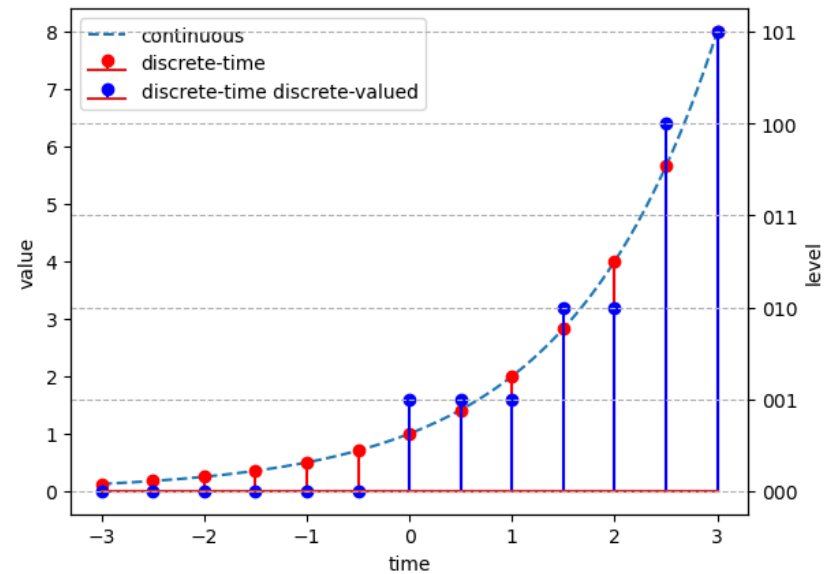
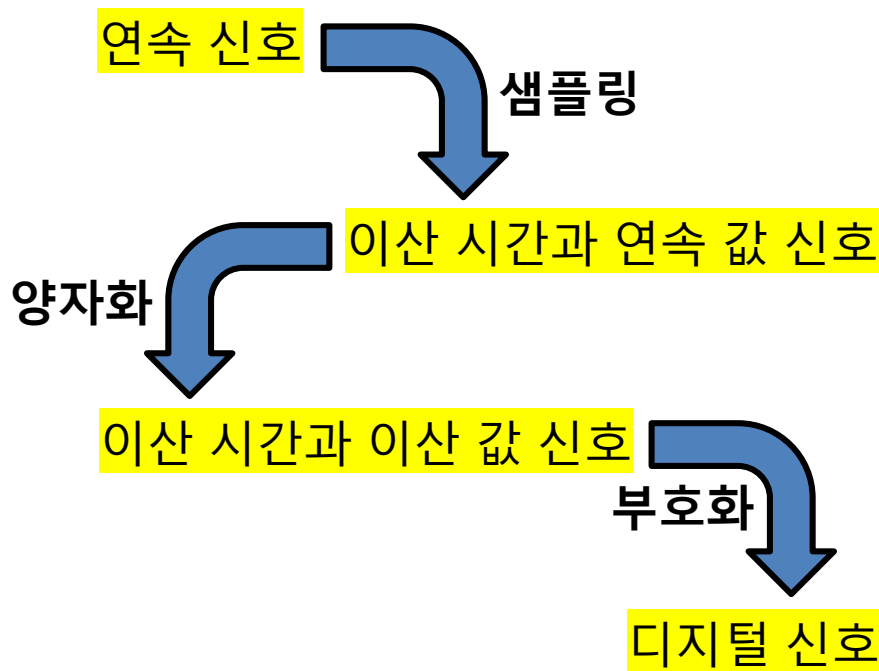
- 아날로그-디지털 변환(**ADC**: analog-to-digital conversion)
 - **양자화** : 유한한 데이터 처리를 위한 디지털 처리를 위해서는 이산 신호를 **이산 시간과 이산 값**을 갖는 신호로 다시 변환하는 과정임

discrete-time discrete-valued signal



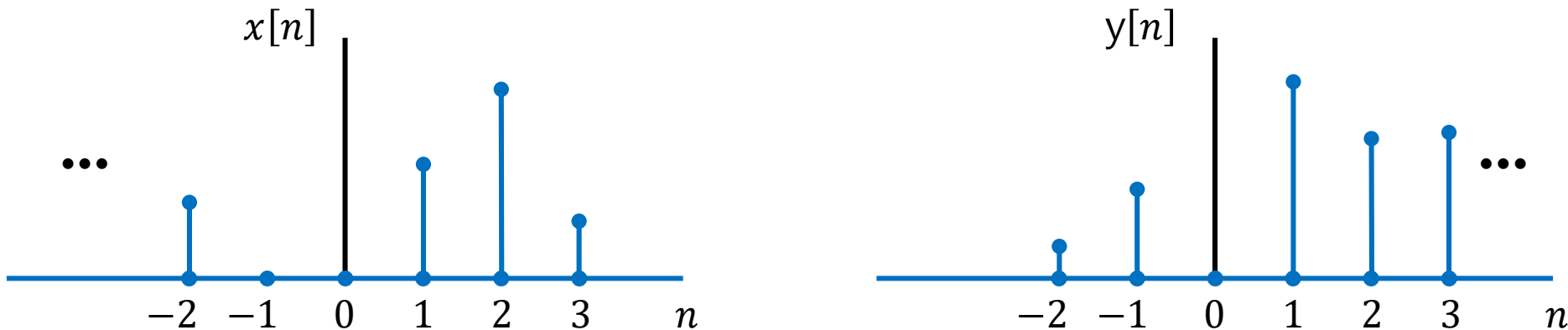
디지털 신호와 시스템

- 아날로그-디지털 변환(**ADC**: analog-to-digital conversion)
 - **부호화** : 각 양화자 구간에 하나의 이진수를 대응시키는 과정임



이산 시스템

- 입력 이산 신호에 어떤 동작이나 연산을 수행하여 다른 이산 신호를 만들어 내는 장치나 알고리즘임
 - 정의 : $y[n] = S\{x[n]\}$

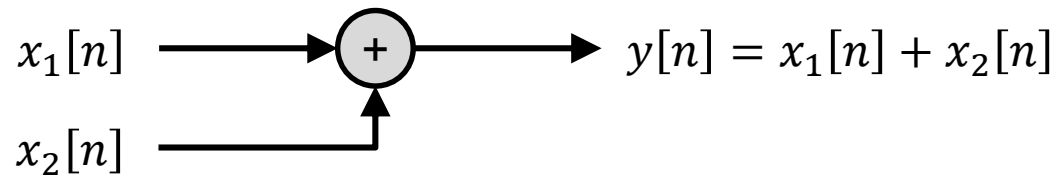


이산 시스템

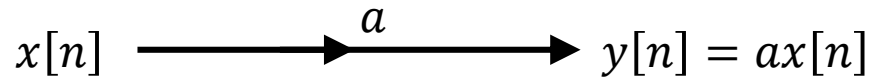
- 예, $x[n] = \begin{cases} |n| & -2 \leq n \leq 2 \\ 0 & \text{다른 경우} \end{cases}$
 - $y[n] = x[n]$
 - 시스템의 출력은 시스템의 입력과 같음
 - $y[n] = x[n-2] = \begin{cases} |n-2| & -2 \leq n-2 \leq 2 \\ 0 & \text{다른 경우} \end{cases} = \begin{cases} |n-2| & 0 \leq n \leq 4 \\ 0 & \text{다른 경우} \end{cases}$
 - 입력 $x[n] = \{\dots, 0, 0, 2, 1, \mathbf{0}, 1, 2, 0, 0, \dots\}$
 - 출력 $y[n] = \{\dots, 0, 0, 0, 0, \mathbf{2}, 1, 0, 1, 2, 0, 0, \dots\}$
 - 입력 신호를 두 샘플만큼 지연시켜 출력됨
 - $y[n] = \frac{1}{2}\{x[n-1] + x[n]\} = \{\dots, 0, 0, 1, \frac{3}{2}, \mathbf{\frac{1}{2}}, \frac{1}{2}, \frac{3}{2}, 1, 0, \dots\}$
 - 입력 신호의 적전 값과 현재 값의 평균값을 출력됨

이산 시스템의 구성 요소

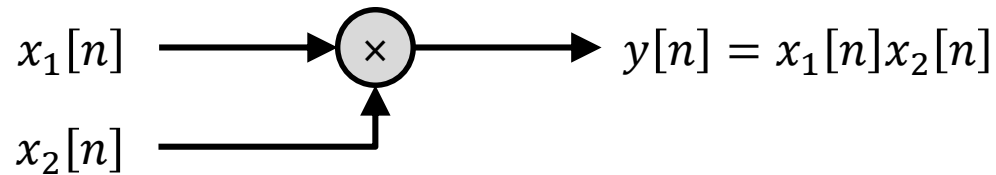
덧셈기



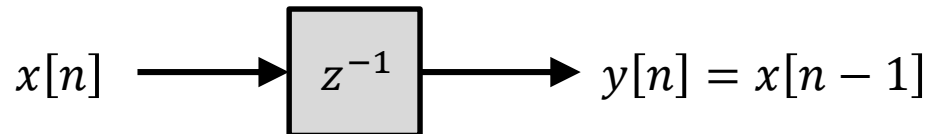
상수 곱셈기



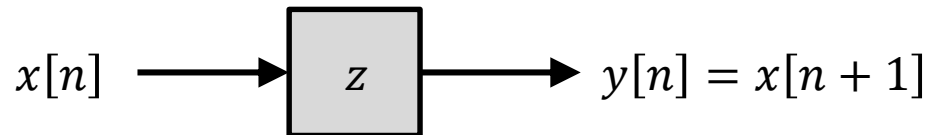
신호 곱셈기



단위 시간 지연기

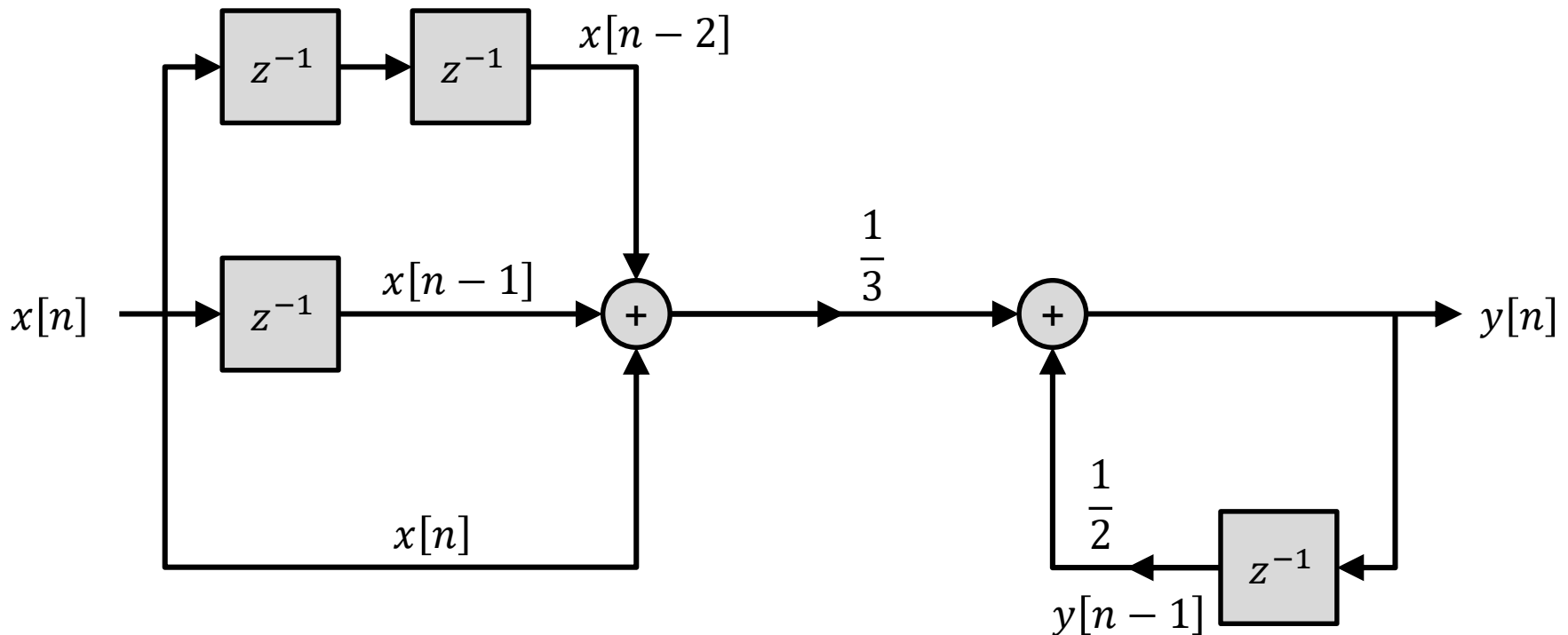


단위 시간 선행기



이산 시스템의 구성 요소

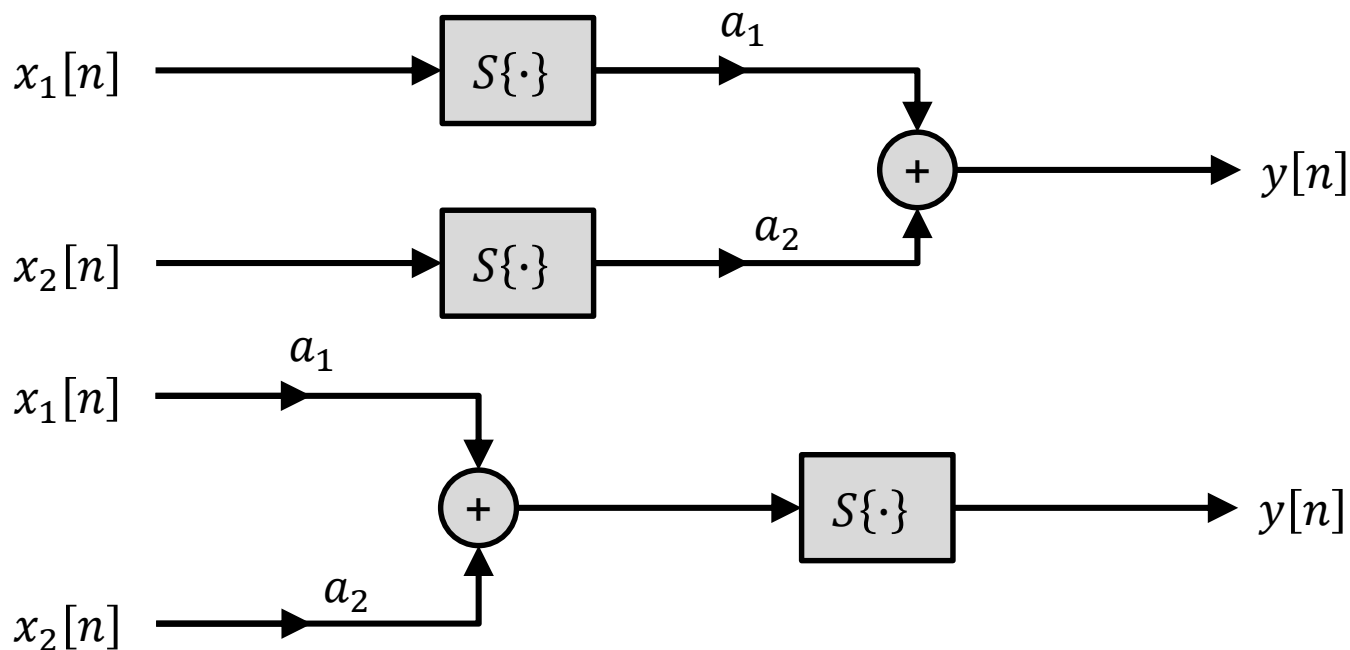
- 예, $y[n] = \frac{1}{3}\{x[n-2] + x[n-1] + x[n]\} + \frac{1}{2}y[n]$



이산 시스템 분류

- 선형(**linear**)과 비선형(**nonlinear**) 시스템
 - 선형 시스템 : 중첩의 원리(**superposition principle**)를 만족하는 시스템임

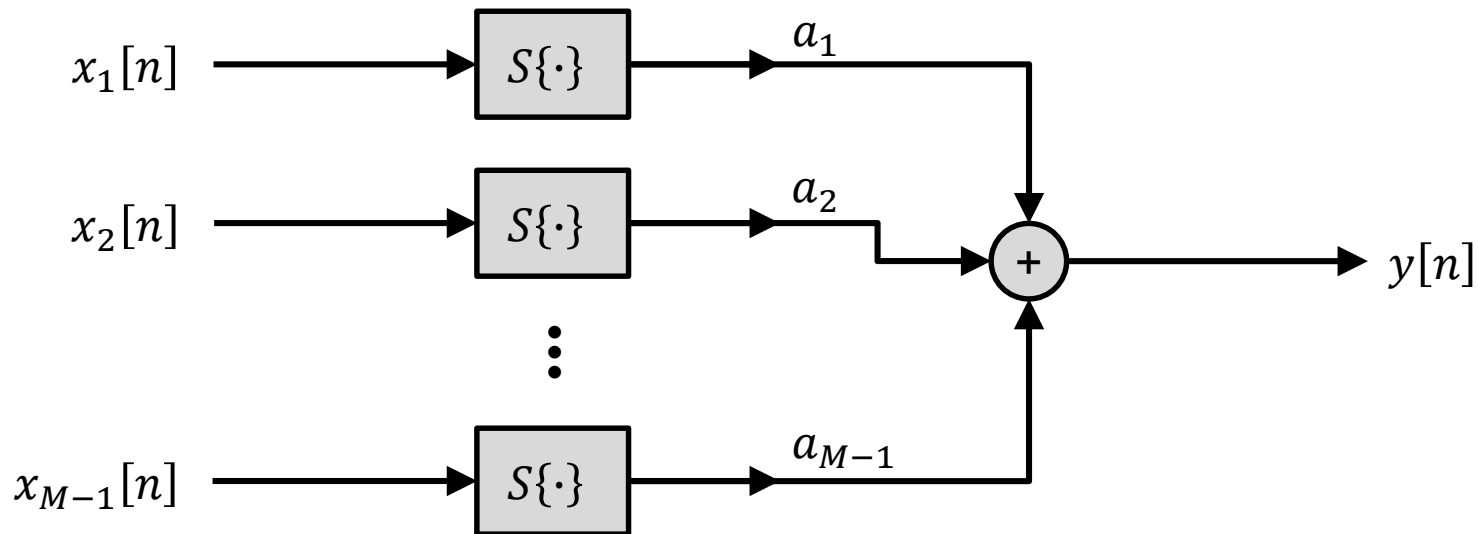
$$S\{a_1x_1[n] + a_2x_2[n]\} = a_1S\{x_1[n]\} + a_2S\{x_2[n]\}$$



이산 시스템 분류

- 선형(**linear**)과 비선형(**nonlinear**) 시스템
 - 선형 시스템을 이용하여 복잡한 신호 $x[n]$ 을 분석할 수 있음

$$x[n] = \sum_{k=1}^{M-1} a_k x_k[n] \rightarrow y[n] = \sum_{k=1}^{M-1} a_k y_k[n]$$



이산 시스템 분류

- 선형(**linear**)과 비선형(**nonlinear**) 시스템
 - 예, $y[n] = 2x[n]$

$$y_1[n] = 2x_1[n], y_2[n] = 2x_2[n]$$

$$S\{a_1x_1[n] + a_2x_2[n]\} = 2\{a_1x_1[n] + a_2x_2[n]\} = 2a_1x_1[n] + 2a_2x_2[n]$$

$$a_1y_1[n] + a_2y_2[n] = a_1\{2x_1[n]\} + a_2\{2x_2[n]\} = 2a_1x_1[n] + 2a_2x_2[n]$$

선형 시스템

이산 시스템 분류

- 선형(**linear**)과 비선형(**nonlinear**) 시스템
 - 예, $y[n] = 2x[n] + 3$

$$y_1[n] = 2x_1[n] + 3, y_2[n] = 2x_2[n] + 3$$

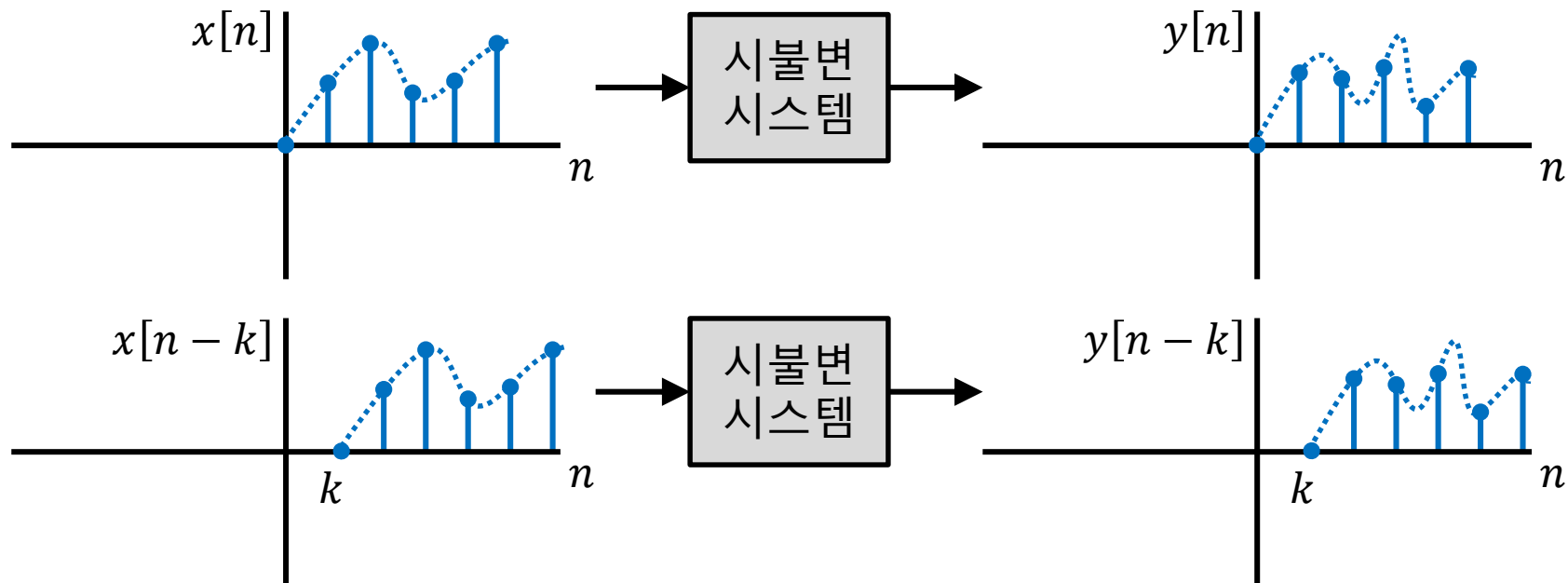
$$S\{a_1x_1[n] + a_2x_2[n]\} = 2\{a_1x_1[n] + a_2x_2[n]\} + 3 = 2a_1x_1[n] + 2a_2x_2[n] + 3$$

$$\begin{aligned} a_1y_1[n] + a_2y_2[n] &= a_1\{2x_1[n] + 3\} + a_2\{2x_2[n] + 3\} \\ &= 2a_1x_1[n] + 2a_2x_2[n] + 3a_1 + 3a_2 \end{aligned}$$

비선형 시스템

이산 시스템 분류

- 시변(**time-varying**)과 시불변(**time-invariant**) 시스템
 - 시불변 : 시스템의 특성이 시간에 따라 변하지 않음
 - 시변 : 시스템의 특성이 시간에 따라 변함



이산 시스템 분류

- 시변(**time-varying**)과 시불변(**time-invariant**) 시스템
 - 예, $y[n] = x[n] - x[n - 2]$

$$y[n - k] = x[n - k] - x[n - k - 2]$$

$$S\{x[n - k]\} = x[n - k] - x[n - k - 2]$$

시불변 시스템

이산 시스템 분류

- 시변(**time-varying**)과 시불변(**time-invariant**) 시스템
 - 예, $y[n] = nx[n]$

$$y[n - k] = (n - k)x[n - k]$$

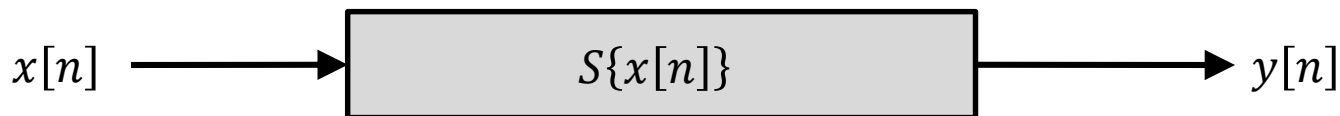
$$S\{x[n - k]\} = nx[n - k]$$

시변 시스템

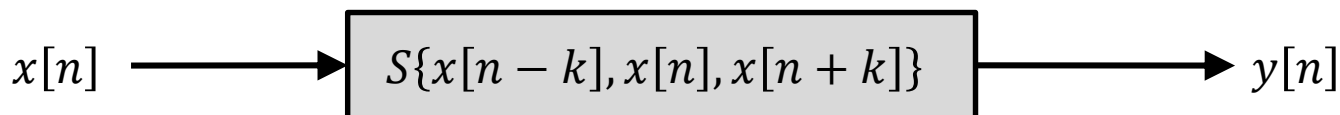
이산 시스템 분류

- 기억(**memory**)과 무기억(**memoryless**) 시스템
 - 무기억 : 현재의 출력 값이 오직 현재의 입력 값에만 의존함
 - 예, $y[n] = 2x[n]$
 - 기억 : 현재의 출력 값이 과거나 미래의 입력 값에 의존함
 - 예, $y[n] = x[n] + 2x[n + 1] + x[n - 3]$

무기억 시스템

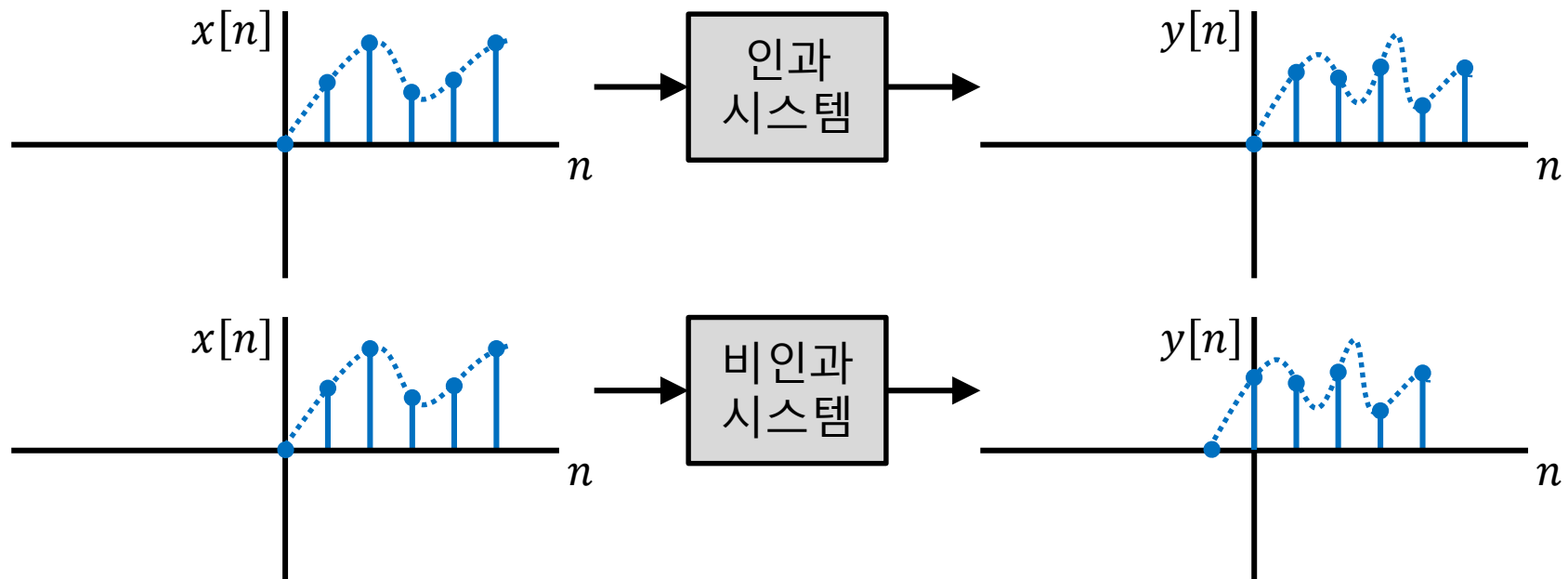


기억 시스템



이산 시스템 분류

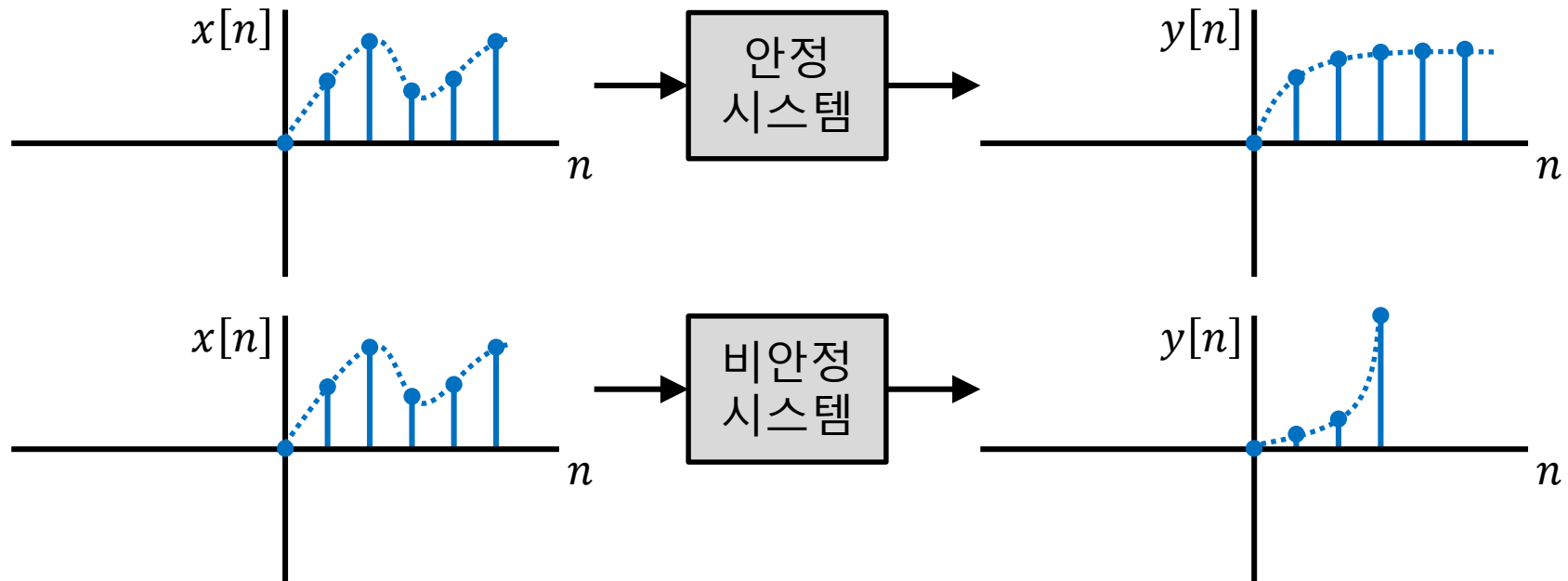
- 인과(**causal**)과 비인과(**noncausal**) 시스템
 - 인과 : 어떤 시각 n 에서의 출력이 과거나 현재의 입력 값에만 의존하고 미래의 값과는 무관함
 - 비인과 : 입력 값이 존재하지 않은 구간에 출력을 내보냄



이산 시스템 분류

안정(**stable**)과 불안정(**unstable**) 시스템

- 안정 : 유한 입력 유한 출력(**BIBO: Bounded Input Bounded Output**) 안정도를 만족함
- 불안정 : BIBO 안정도를 만족하지 않음



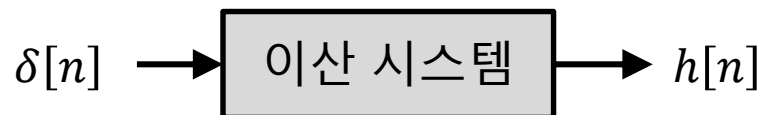
이산 선형 시불변 시스템

- LTI(**L**inear **T**ime-**I**nvariant) 시스템이라고 부름
 - 시간에 따라 변화하지 않은 시스템
 - 중첩의 원리(**superposition principle**)를 만족하는 시스템
 - 가산성(additivity) : $f(x_1 + x_2) = f(x_1) + f(x_2)$
 - 균일성(homogeneity) : $f(ax_1) = af(x_1)$



$$S\{a_1x_1[n] + a_2x_2[n]\} = a_1S\{x_1[n]\} + a_2S\{x_2[n]\}$$

- 시스템의 임펄스 응답(**impulse response**)
 - 정의 : $h[n] = S\{\delta[n]\}$



이산 선형 시불변 시스템

- LTI 시스템의 응답

$$y[n] = S\{x[n]\}$$

$$= S\{\sum_{k=-\infty}^{\infty} x[k]\delta[n-k]\}$$

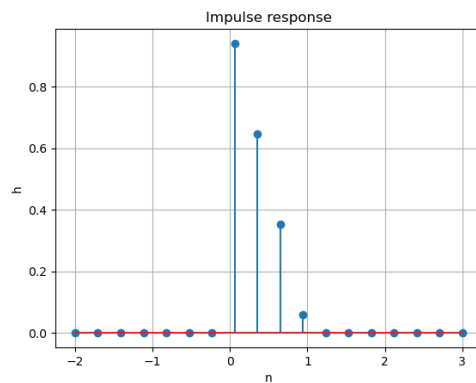
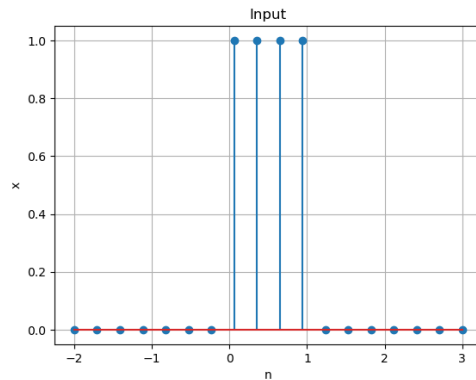
$$= \sum_{k=-\infty}^{\infty} x[k]S\{\delta[n-k]\}$$

$$= \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

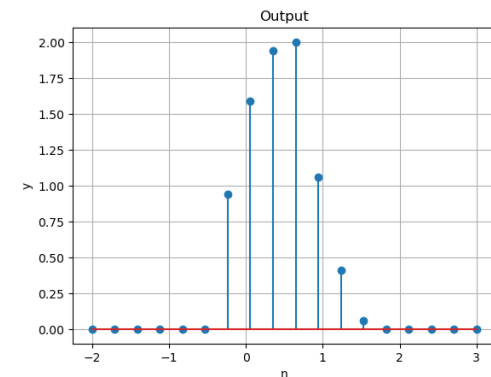
$$= x[n] * h[n] \leftarrow \bullet \text{컨벌루션 합(convolution sum)}$$

이산 선형 시불변 시스템

■ LTI 시스템의 응답



$$y[n] = x[n] * h[n]$$



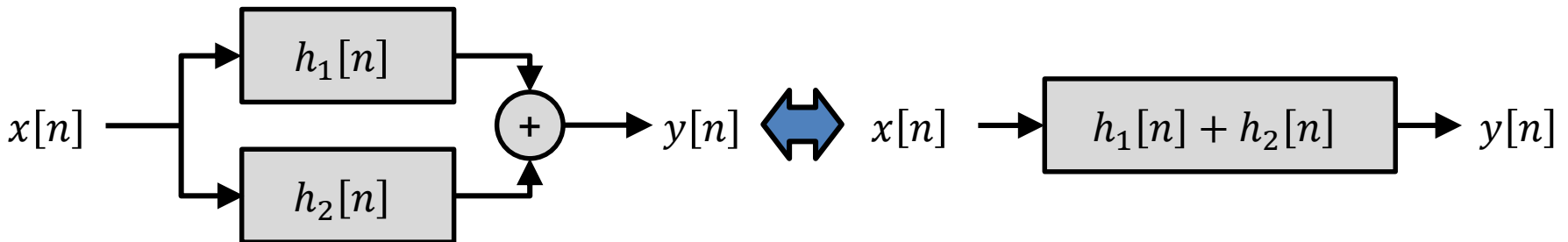
이산 선형 시불변 시스템

■ 컨벌루션의 성질

- 교환 법칙 : $x[n] * h[n] = h[n] * x[n]$
- 결합 법칙 : $\{x[n] * h_1[n]\} * h_2[n] = x[n] * \{h_1[n] * h_2[n]\}$



- 분배 법칙 : $x[n] * \{h_1[n] + h_2[n]\} = x[n] * h_1[n] + x[n] * h_2[n]$



이산 선형 시불변 시스템

- 컨벌루션의 성질

- 인과 LTI 시스템 : 어느 시각 $n = n_0$ 에서의 출력은 $n \leq n_0$ 의 입력 신호 값에 의해서만 결정됨

$$y[n_0] = \sum_{k=-\infty}^{\infty} h[k]x[n_0 - k]$$

$$= \sum_{k=0}^{\infty} h[k]x[n_0 - k] + \sum_{k=-\infty}^{-1} h[k]x[n_0 - k]$$

$$= \{h[0]x[n_0] + h[1]x[n_0 - 1] + h[2]x[n_0 - 2] + \dots\} + \{h[-1]x[n_0 + 1] + h[-2]x[n_0 + 2] + h[-3]x[n_0 + 3] + \dots\}$$

- 시스템이 인과 LTI 성질을 갖기 위해서 $\{h[-1]x[n_0 + 1] + h[-2]x[n_0 + 2] + h[-3]x[n_0 + 3] + \dots\} = 0$

$$h[n] = 0, n < 0$$

이산 선형 시불변 시스템

- 컨벌루션의 성질
 - 안정 LTI 시스템

$$|y[n]| = \left| \sum_{k=-\infty}^{\infty} h[k]x[n-k] \right|$$

$$\leq \sum_{k=-\infty}^{\infty} |h[k]| |x[n-k]|$$

←.....• $h[k]$ 와 $x[n-k]$ 를 곱하면 음수, 양수의 경우가 모두 발생함. 이 값을 더하는 값보다 절댓값을 취해서 모두 양수 값을 만들고 더하는 것이 당연히 큼

$$\leq M_x \sum_{k=-\infty}^{\infty} |h[k]|$$

←.....• 입력이 Bounded 되어 있으므로 유한한 값을 가져야 함

차분 방정식의 표현법

- 이산 시스템을 모델링할 때 많이 사용됨

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

- N 개의 계수 a_k 와 $(M+1)$ 개의 계수 b_k 는 상수이며 시스템의 특징에 의해 결정됨
 - 해를 구하기 위해서 다음과 같이 N 개의 초기 조건이 필요함
 $y[-1], y[-2], \dots, y[-N]$
 - 초기 조건을 이용하여 $y[0], y[1], y[2], \dots, y[n]$ 을 순서대로 구할 수 있으므로 $y[n]$ 의 하나의 식으로 표현하지 못함
- 차분 방정식의 **균일해**(homogeneous solution)와 **특수해**(particular solution)를 구하는 해석적인 방법이 필요함

$$y[n] = y_h[n] + y_p[n]$$

차분 방정식의 표현법

- 균일해

- 입력 신호 $x[n] = 0$ 이라고 가정하여 구함

$$y[n] + \sum_{k=1}^N a_k y[n-k] = 0$$

- 특수해

- 입력 신호 $x[n]$ 이 주어졌을 때 차분 방정식의 근이며 입력 $x[n]$ 의 형태에 따라 결정됨

$$y[n] + \sum_{k=1}^N a_k y[n-k] = x[n]$$