신호 및 시스템

제 *6* 장. 이산시간 선형 시불변 시스템의 시간 영역 해석

박섭형

한림대학교

2021학년도 1학기

배울 내용

- 이산시간 단위 임펄스 함수와 단위 계단 함수
- 이산시간 선형 시불변(LTI: linear time invariant) 시스템의 시간 영역 응답
- 이산시간 선형 시불변 시스템의 출력은 입력 신호와 시스템의 임펄스 응답(impulse response)의 콘볼루션 합(convolution sum)으로 나타난다
  참고: CNN(convolutional neural networks)
- 이산시간 선형 시불변 시스템의 상태 공간 표현

이산시간 단위 임펄스 함수

정의 *6.1 (*크로네커 델타*(Kronecker-delta)* 함수*)*

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} . \tag{6.1}$$

n	$\cdots$	-2	-1	0	1	2	3	4	5	6	$\cdots$
$\delta[n]$	0	0	0	1	0	0	0	0	0	0	0

단위 임펄스 신호를 이용한 일반적인 신호 표현

$$x[n] = \delta[n] + 3\delta[n-1] + 5\delta[n-2] + 4\delta[n-3] + 2\delta[n-4]. \tag{6.2}$$

n	$\cdots$	-2	-1	0	1	2	3	4	5	6	$\cdots$
$\delta[n]$	0	0	0	1	0	0	0	0	0	0	0
$3\delta[n-1]$	0	0	0	0	3	0	0	0	0	0	0
$5\delta[n-2]$	0	0	0	0	0	5	0	0	0	0	0
$4\delta[n-3]$	0	0	0	0	0	0	4	0	0	0	0
$2\delta[n-4]$	0	0	0	0	0	0	0	2	0	0	0
$x[n]$	0	0	0	1	3	5	4	2	0	0	0

이 표가 의미하는 것은 옆으로 이동한 임펄스들에 가중치를 준 후에 더한 가중합(weighted sum)으로 임의의 신호 열을 표현할 수 있다는 것이다.

이산 시간 임펄스 신호의 체질 성질

$$x[0] = 1, \quad x[1] = 3, \quad x[2] = 5, \quad x[3] = 4, \quad x[4] = 2. \tag{6.3}$$

$$\begin{aligned} x[n] &= x[0]\delta[n] + x[1]\delta[n-1] + x[2]\delta[n-2] + x[3]\delta[n-3] + x[4]\delta[n-4] \\ &= \sum_{k=0}^{4} x[k]\delta[n-k] \end{aligned} \tag{6.4}$$

이 개념을 확장하면 무한 길이 신호 $x[n]$ 을 다음과 같이 표현할 수 있다.

$$\begin{aligned} x[n] &= \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \\ &= \cdots + x[-1]\delta[n+1] + x[0]\delta[n] + x[1]\delta[n-1] + \cdots . \end{aligned} \tag{6.5}$$

이 식을 이산 시간 임펄스 신호의 체질 성질(sifting property)이라고 한다.

이산시간 단위 계단 함수

정의 6.2 (이산시간 단위 계단 *(unit step)* 함수)

$$u[n] = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}. \tag{6.6}$$

FIR 필터의 단위 임펄스 응답 *(1)*

FIR(finite impulse response, 유한 임펄스 응답) 필터의 입출력 관계식

$$y[n] = \sum_{k=0}^{M} b_k x[n-k]. \tag{6.7}$$

출력 $y[n]$은 입력 열 $x[n-M], x[n-M-1], \cdots, x[n-1], x[n]$에 각각 $b_M, b_{M-1}, \cdots, b_1, b_0$을 곱한 후에 더하는 가중 합으로 표현된다.

FIR 필터의 단위 임펄스 응답 *(2)*

모든 k에 대해서 $b_k = \frac{1}{M+1}$ 인 FIR 필터: $(M+1)$ 점 이동 평균 필터

$$y[n] = \frac{1}{M+1}\sum_{k=0}^{M} x[n-k]. \tag{6.8}$$

- $M = 1$: 2 점 이동 평균 필터

$$y[n] = \frac{1}{2}(x[n] + x[n-1]).$$

- $M = 2$: 3 점 이동 평균 필터

$$y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2]).$$

FIR 필터의 단위 임펄스 응답 *(3)*

FIR 필터의 입출력 관계식

$$y[n] = \sum_{k=0}^{M} b_k x[n-k].$$

FIR 필터의 임펄스 응답: $x[n] = \delta[n]$가 입력되었을 때의 출력 (또는 응답)

$$h[n] = \sum_{k=0}^{M} b_k \delta[n-k] = \begin{cases} b_n, & n = 0, 1, 2, \ldots M \\ 0, & \text{otherwise} \end{cases}. \tag{6.9}$$

참고:

$$x[n] = x[0]\delta[n] + x[1]\delta[n-1] + x[2]\delta[n-2] + x[3]\delta[n-3] + x[4]\delta[n-4]$$

$$= \sum_{k=0}^{4} x[k]\delta[n-k]$$

FIR 필터의 단위 임펄스 응답 (4)

$$
\begin{aligned}
h[-1] &= \sum_{k=0}^{M} b_k \delta[-1-k] = b_0\delta[-1] + b_1\delta[-2] + \cdots + b_M\delta[-1-M] = 0 \\
h[0] &= \sum_{k=0}^{M} b_k \delta[0-k] = b_0\delta[0] + b_1\delta[-1] + b_2\delta[-2] + \cdots + b_M\delta[-M] = b_0 \\
h[1] &= \sum_{k=0}^{M} b_k \delta[1-k] = b_0\delta[1] + b_1\delta[0] + b_2\delta[-1] + \cdots + b_M\delta[-M] = b_1 \\
&\vdots \\
h[M] &= \sum_{k=0}^{M} b_k \delta[M-k] = b_M \\
h[M+1] &= \sum_{k=0}^{M} b_k \delta[M+1-k] = 0
\end{aligned}
$$

FIR 필터의 단위 임펄스 응답 *(5)*

$$h[k] = b_k, \quad k = 0, 1, \cdots, M. \tag{6.10}$$

이산시간 선형 시불변 시스템의 출력을 구하는 콘볼루션 합

$$y[n] = \sum_{k=0}^{M} h[k]x[n-k]. \tag{6.11}$$

FIR 필터의 임펄스 응답 $h[n]$은 차분방정식의 계수들로 구성된 신호열로서, $0 \le n \le M$ 범위에서만 0이 아닌 값을 갖고, 그 외의 구간에서는 $h[n] = 0$이 된다. 즉, 이 시스템의 임펄스 응답 $h[n]$의 지지집합 (support)는 유한 구간이 된다. 이와 같이 임펄스 응답의 지지집합이 유한 구간이 되는 필터를 FIR 필터라고 부른다.

그림 *6.1:* FIR 필터에서 임펄스 응답의 정의를 설명하는 블록도.

지연 시스템의 임펄스 응답 *(1)*

$$y[n] = x[n - n_0]. \tag{6.12}$$

이 식에서 $n_0 = 1$ 인 경우를 단위 지연(unit delay) 시스템이라고 한다.
이 시스템의 임펄스 응답 $h[n]$ 은 $x[n]$ 대신에 $\delta[n]$ 을 대입했을 때의 출력이므로, $h[n] = \delta[n - n_0]$ 가 된다.
예를 들어서, $n_0 = 5$ 인 지연 시스템의 임펄스 응답은 다음과 같다.

$$h[n] = \delta[n-5] = \begin{cases} 1, & n = 5 \\ 0, & n \neq 5 \end{cases}. \tag{6.13}$$

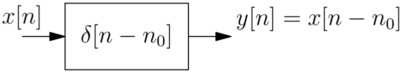
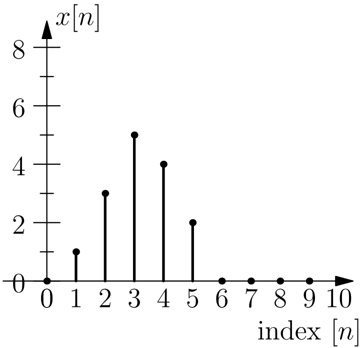
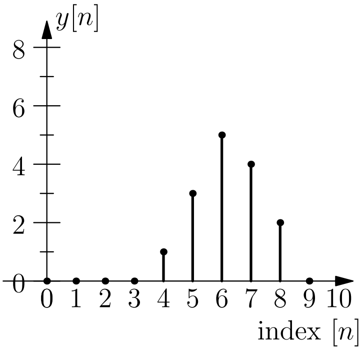


그림 *6.2:* 입력을 n_0 만큼 지연시키는 지연 시스템의 블록도.

지연 시스템의 임펄스 응답 *(2)*



(a) 입력 $x[n]$의 그래프.



(b) $y[n] = x[n-3]$의 그래프.

그림 *6.3:* 입력 $x[n]$과 지연된 출력 $y[n] = x[n-3]$의 그래프.

콘볼루션 합

임펄스 응답이 $h[n]$ 인 LTI 시스템 $\mathcal{T}$ 에 $x[n]$ 이 입력되면, 출력 $y[n]$ 은 다음과 같이 $x[n]$ 과 $h[n]$ 의 관계식으로 나타나는데, 이것을 콘볼루션 합 (convolution sum)이라고 부른다. 콘볼루션 합 연산자는 $*$ 로 표현한다.

$$y[n] = \mathcal{T}\{x[n]\} = x[n] * h[n] = \sum_{k=0}^{M} x[k]h[n-k]. \tag{6.14}$$

이 때, 출력 $y[n]$ 은 두 신호 열 $x[n]$ 과 $h[n]$ 의 콘볼루션 합의 연산을 수행한 결과라고 말한다.

콘볼루션 합 공식의 유도 *(1)*

$$\begin{aligned} x[n] &= \sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \\ &= \cdots + x[-2]\delta[n+2] + x[-1]\delta[n+1] + x[0]\delta[n] \\ &\quad + x[1]\delta[n-1] + x[2]\delta[n-2] + \cdots . \end{aligned} \tag{6.15}$$

단위 임펄스 신호 $\delta[n]$에 대한 출력은 임펄스 응답 $h[n]$이다. 즉,

$$h[n] = \mathcal{T}\{\delta[n]\}. \tag{6.16}$$

여기에 선형 시불변 시스템의 시불변성을 이용하면 모든 정수 k에 대해서 다음과 같은 입출력 관계가 성립한다.

$$h[n-k] = \mathcal{T}\{\delta[n-k]\}. \tag{6.17}$$

콘볼루션 합 공식의 유도 *(2)*

선형 시불변 시스템의 선형성을 이용하면, 다음과 같은 관계식을 얻을 수 있다.

$$
\begin{aligned}
y[n] = \mathcal{T}\{x[n]\} &= \mathcal{T}\left\{\sum_{k=-\infty}^{\infty} x[k]\delta[n-k]\right\} \\
&= \cdots + \mathcal{T}\{x[-2]\delta[n+2]\} + \mathcal{T}\{x[-1]\delta[n+1]\} + \mathcal{T}\{x[0]\delta[n]\} \\
&\quad + \mathcal{T}\{x[1]\delta[n-1]\} + \mathcal{T}\{x[2]\delta[n-2]\} + \cdots \\
&= \cdots + x[-2]\mathcal{T}\{\delta[n+2]\} + x[-1]\mathcal{T}\{\delta[n+1]\} + x[0]\mathcal{T}\{\delta[n]\} \\
&\quad + x[1]\mathcal{T}\{\delta[n-1]\} + x[2]\mathcal{T}\{\delta[n-2]\} + \cdots \\
&= \cdots + x[-2]h[n+2] + x[-1]h[n+1] + x[0]h[n] \\
&\quad + x[1]h[n-1] + x[2]h[n-2] + \cdots \\
&= \sum_{k=-\infty}^{\infty} x[k]h[n-k].
\end{aligned} \tag{6.18}
$$

콘볼루션 합 공식의 유도 *(3)*

모든 이산시간 선형 시불변 시스템의 출력은 다음과 같이 입력과 임펄스 응답의 콘볼루션 합으로 표현할 수 있다.

이산시간 선형 시불변 시스템의 출력 : 콘볼루션 합

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]. \tag{6.19}$$

콘볼루션 합의 성질: 교환 법칙

두 신호를 콘볼루션할 때 신호의 순서를 바꾸어도 결과는 같다.

$$\begin{aligned} y[n] = x[n] * h[n] &= \sum_{l=-\infty}^{\infty} x[l]h[n-l] \\ &= \sum_{k=-\infty}^{\infty} x[n-k]h[k] \qquad (k = n - l\text{로 변수 치환}) \\ &= \sum_{k=-\infty}^{\infty} h[k]x[n-k] = h[n] * x[n]. \end{aligned} \tag{6.20}$$

즉, 콘볼루션은 다음과 같이 교환성을 가지고 있다.

$$x[n] * h[n] = h[n] * x[n]. \tag{6.21}$$

콘볼루션 합의 성질: 결합 법칙 *(1)*

$$
\begin{aligned}
x_1[n] * (x_2[n] * x_3[n]) &= x_1[n] * \left(\sum_{k=-\infty}^{\infty} x_2[k] x_3[n-k] \right) \\
&= \sum_{m=-\infty}^{\infty} x_1[m] \left(\sum_{k=-\infty}^{\infty} x_2[k] x_3[(n-m)-k] \right). \qquad (6.22)
\end{aligned}
$$

콘볼루션 합의 성질: 결합 법칙 *(2)*

여기에서 $q = k + m$으로 치환하면, $k = q - m$이므로,

$$
\begin{aligned}
x_1[n] * (x_2[n] * x_3[n]) &= \sum_{m=-\infty}^{\infty} x_1[m] \sum_{q=-\infty}^{\infty} x_2[q-m]x_3[n-q] \\
&= \sum_{q=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} x_1[m]x_2[q-m]x_3[n-q] \\
&= \sum_{q=-\infty}^{\infty} \left(\sum_{m=-\infty}^{\infty} x_1[m]x_2[q-m] \right) x_3[n-q] \\
&= \sum_{q=-\infty}^{\infty} (x_1[q] * x_2[q])\, x_3[n-q] \\
&= (x_1[n] * x_2[n]) * x_3[n].
\end{aligned} \tag{6.23}
$$

콘볼루션 합의 성질: 결합 법칙 *(3)*

즉, 콘볼루션 합은 다음과 같이 결합성을 가지고 있다.

$$(x_1[n] * x_2[n]) * x_3[n] = x_1[n] * (x_2[n] * x_3[n]). \tag{6.24}$$

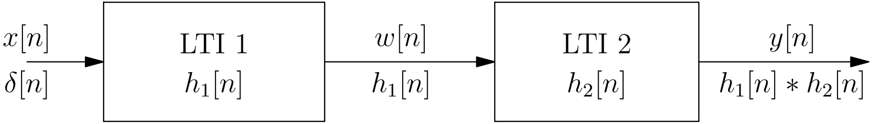


그림 *6.4:* 두 개의 이산시간 선형 시불변 시스템을 직렬 연결한 그림.

$$\begin{aligned} y[n] &= (x[n] * h_1[n]) * h_2[n] \\ &= x[n] * (h_1[n] * h_2[n]) \\ &= x[n] * (h_2[n] * h_1[n]) \\ &= (x[n] * h_2[n]) * h_1[n]. \end{aligned} \tag{6.25}$$

콘볼루션 합의 성질: 결합 법칙 *(4)*

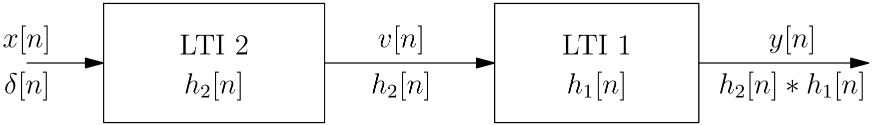


그림 ***6.5:*** 두 개의 이산시간 선형 시불변 시스템의 순서를 바꾸어 직렬 연결한 그림.

콘볼루션 합의 성질: 분배 법칙 *(1)*

$$
\begin{aligned}
x_1[n] * (x_2[n] + x_3[n]) &= \sum_{k=-\infty}^{\infty} x_1[k]\left(x_2[n-k] + x_3[n-k]\right) \\
&= \sum_{k=-\infty}^{\infty} x_1[k]x_2[n-k] + \sum_{k=-\infty}^{\infty} x_1[k]x_3[n-k] \\
&= x_1[n] * x_2[n] + x_1[n] * x_3[n].
\end{aligned} \tag{6.26}
$$

콘볼루션 합의 성질: 분배 법칙 *(2)*

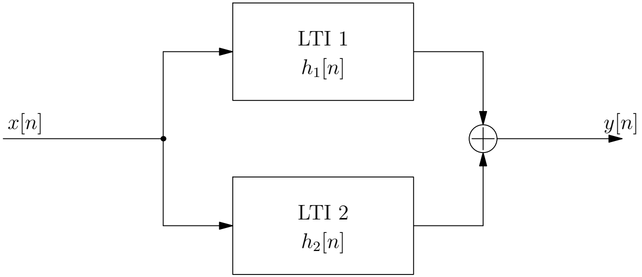


그림 *6.6:* 두 개의 이산시간 선형 시불변 시스템을 병렬로 연결한 시스템.

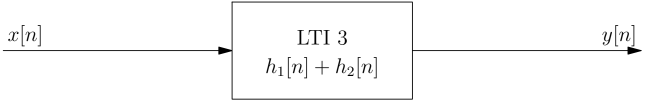


그림 *6.7:* 두 개의 이산시간 선형 시불변 시스템을 병렬로 연결한 것과 등가 시스템.

콘볼루션 합의 성질 : 항등원

모든 신호 $x[n]$에 대해서 다음 식이 성립한다.

$$x[n] * \delta[n] = x[n]. \tag{6.27}$$

이 식은 다음과 같이 증명할 수 있다.

$$x[n] * \delta[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n-k]. \tag{6.28}$$

$$\sum_{k=-\infty}^{\infty} x[k]\delta[n-k] = x[n]. \tag{6.29}$$

이 식이 의미하는 것은 콘볼루션 합 연산의 항등원은 $\delta[n]$라는 것이다.

$$x[n] * \delta[n-n_0] = x[n-n_0], \quad n_0\text{은 정수} \tag{6.30}$$

$$x[n-n_0] * \delta[n-n_1] = x[n-n_0-n_1], \quad n_0, n_1\text{은 정수} \tag{6.31}$$

입력 신호를 n_0 만큼 지연시키는 시스템의 임펄스 응답은 $h[n] = \delta[n-n_0]$인 것을 알 수 있다.

임펄스 응답과 단위 계단 응답

이산시간 LTI 시스템의 단위 계단 응답을 $s[n]$라고 하면, $s[n]$은 다음과 같다.

$$s[n] = h[n] * u[n] = \sum_{k=-\infty}^{\infty} h[k]u[n-k]. \tag{6.32}$$

그런데

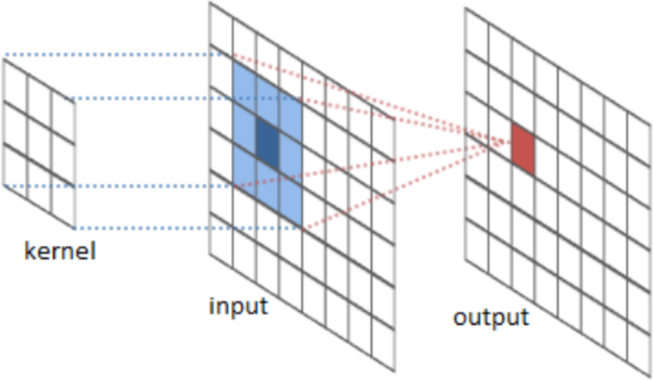
$$u[n-k] = \begin{cases} 1, & n-k \geq 0 \quad (k \leq n) \\ 0, & n-k < 0 \quad (k > n) \end{cases} \tag{6.33}$$

이므로

$$s[n] = \sum_{k=-\infty}^{n} h[k]. \tag{6.34}$$

$$h[n] = \sum_{k=-\infty}^{n} h[k] - \sum_{k=-\infty}^{n-1} h[k] = s[n] - s[n-1]. \tag{6.35}$$

2 차원 콘볼루션의 예



디지털 필터

- FIR 필터: 임펄스 응답의 지지집합이 유한한 (finite) 필터를 말한다. 입력 신호열에서 한정된 계수의 샘플에 계수를 곱하여 출력 신호열을 얻는 필터이다.
- IIR 필터: 임펄스 응답의 지지집합이 무한한 (infinite) 필터를 말한다. 출력 신호를 계산할 때 입력 신호와 과거의 출력 신호를 함께 사용하면 이런 효과를 얻을 수 있다.

FIR 필터의 선형성

입력 $x_1[n]$과 $x_2[n]$에 대한 출력을 각각 $y_1[n]$과 $y_2[n]$라고 하자.

$$y_1[n] = \sum_{k=0}^{M} b_k x_1[n-k], \qquad y_2[n] = \sum_{k=0}^{M} b_k x_2[n-k].$$

α와 β를 임의의 스칼라라고 하고 $x[n] = \alpha x_1[n] + \beta x_2[n]$이라고 정의하자. 그러면 $x[n]$을 FIR 필터에 입력했을 때의 출력 $y[n]$은 다음과 같이 구할 수 있다.

$$\begin{aligned} y[n] &= \sum_{k=0}^{M} b_k x[n-k] \\ &= \sum_{k=0}^{M} b_k(\alpha x_1[n-k] + \beta x_2[n-k]) \\ &= \alpha \sum_{k=0}^{M} b_k x_1[n-k] + \beta \sum_{k=0}^{M} b_k x_2[n-k] \\ &= \alpha y_1[n] + \beta y_2[n]. \end{aligned} \tag{6.36}$$

FIR 필터의 시불변성

$v[n] = x[n-n_0]$ 라고 정의하고, $w[n]$ 을 $v[n]$ 에 대한 출력이라고 하자.

$$\begin{aligned} w[n] &= \sum_{k=0}^{M} b_k v[n-k] \\ &= \sum_{k=0}^{M} b_x x[(n-k)-n_0] \\ &= \sum_{k=0}^{M} b_k x[(n-n_0)-k]. \end{aligned} \tag{6.37}$$

그리고 $y[n] = \sum_{k=0}^{M} b_k x[n-k]$ 로부터 $y[n-n_0]$ 을 계산하면 다음 식을 얻는다.

$$y[(n-n_0)] = \sum_{k=0}^{M} b_k x[(n-n_0)-k]. \tag{6.38}$$

즉, $w[n] = y[n-n_0]$ 이므로 FIR 필터는 시불변 시스템이다.

IIR 필터의 선형 시불변성

입출력 관계식이 다음과 같이 주어지는 IIR 필터는 선형, 시불변 시스템이다.

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] - \sum_{l=1}^{N} a_l y[n-l]. \tag{6.39}$$

여기에서 $a_0 = 1$이라고 하면 식 (6.39)는 다음과 같이 쓸 수 있다.

$$\sum_{l=0}^{N} a_l y[n-l] = \sum_{k=0}^{M} b_k x[n-k]. \tag{6.40}$$

IIR 필터의 선형성

입력 $x_1[n]$과 $x_2[n]$에 대한 출력을 각각 $y_1[n]$과 $y_2[n]$라고 하자.

$$\sum_{l=0}^{N} a_l y_1[n-l] = \sum_{k=0}^{M} b_k x_1[n-k], \qquad \sum_{l=0}^{N} a_l y_2[n-l] = \sum_{k=0}^{M} b_k x_2[n-k].$$

$x[n] = \alpha x_1[n] + \beta x_2[n]$에 대한 출력 $y[n]$은 다음과 같이 구할 수 있다.

$$\begin{aligned}\sum_{l=0}^{N} a_l y[n-l] = \sum_{k=0}^{M} b_k x[n-k] &= \sum_{k=0}^{M} b_k(\alpha x_1[n-k] + \beta x_2[n-k]) \\ &= \alpha \sum_{k=0}^{M} b_k x_1[n-k] + \beta \sum_{k=0}^{M} b_k x_2[n-k] \\ &= \alpha \sum_{l=0}^{N} a_l y_1[n-l] + \beta \sum_{l=0}^{N} a_l y_2[n-l] \\ &= \sum_{l=0}^{N} a_l \left\{\alpha y_1[n-l] + \beta y_2[n-l]\right\}.\end{aligned} \tag{6.49}$$

IIR 필터의 시불변성

$v[n] = x[n - n_0]$ 라고 정의하고, $w[n]$을 $v[n]$에 대한 출력이라고 하자.

$$\begin{aligned} \sum_{l=0}^{N} a_l w[n-l] &= \sum_{k=0}^{M} b_k v[n-k] \\ &= \sum_{k=0}^{M} b_x x[(n-k)-n_0] \\ &= \sum_{k=0}^{M} b_k x[(n-n_0)-k]. \end{aligned} \tag{6.50}$$

그리고 $\sum_{l=0}^{N} a_l y[n-l] = \sum_{k=0}^{M} b_k x[n-k]$에 n 대신에 $n - n_0$을 대입하면 다음 식을 얻는다.

$$\sum_{l=0}^{N} a_l y[(n-n_0)-l] = \sum_{k=0}^{M} b_k x[(n-n_0)-k]. \tag{6.51}$$

FIR 필터의 콘볼루션 합

$h[n]$의 지지집합이 $0 \le n \le M$인 경우의 콘볼루션 합

$$y[n] = \sum_{k=0}^{M} h[k]x[n-k]. \tag{6.52}$$

입력 $x[n]$과 임펄스 응답 $h[n]$이 다음과 같이 주어지는 예를 통하여 콘볼루션을 계산하는 방법을 이해해 보자.

$$x[n] = \{\underset{\uparrow}{1}, 3, 5, 4, 2\},$$

$$h[n] = \{\underset{\uparrow}{1}, 3, -2, 2\},$$

여기에서 화살표 ($\uparrow$)는 $n = 0$일 때의 위치를 나타낸다.
이 경우에 출력 $y[n]$은 다음과 같이 쓸 수 있다.

$$\begin{aligned} y[n] &= \sum_{k=0}^{3} h[k]x[n-k] \\ &= h[0]x[n-0] + h[1]x[n-1] + h[2]x[n-2] + h[3]x[n-3]. \end{aligned} \tag{6.53}$$

FIR 필터의 콘볼루션 합 구하는 방법 *I-1*

이 방법은 다음 식의 오른 쪽 네 항을 계산하여 더하는 방법이다.

$$y[n] = h[0]x[n-0] + h[1]x[n-1] + h[2]x[n-2] + h[3]x[n-3].$$

n	0	1	2	3	4	5	6	7	8
$x[n]$	1	3	5	4	2				
$h[n]$	1	3	-2	2					
$h[0]x[n-0]$	1	3	5	4	2				
$h[1]x[n-1]$		3	9	15	12	6			
$h[2]x[n-2]$			-2	-6	-10	-8	-4		
$h[3]x[n-3]$				2	6	10	8	4	
$y[n]$	1	6	12	15	10	8	4	4	0

이 표에서 $y[n]$을 구하기 위해서는 $y[n]$ 바로 위에 네 줄을 더하면 된다.

FIR 필터의 콘볼루션 합 구하는 방법 *I-2*

$$y[n] = \sum_{k=0}^{4} x[k]h[n-k] = x[0]h[n-0] + x[1]h[n-1] + x[2]h[n-2] + x[3]h[n-3] + x[4]h[n-4].$$

n	0	1	2	3	4	5	6	7	8
$x[n]$	1	3	5	4	2				
$h[n]$	1	3	-2	2					
$x[0]h[n-0]$	1	3	-2	2					
$x[1]h[n-1]$		3	9	-6	6				
$x[2]h[n-2]$			5	15	-10	10			
$x[3]h[n-3]$				4	12	-8	8		
$x[4]h[n-4]$					2	6	-4	4	
$y[n]$	1	6	12	15	10	8	4	4	0

이 표에서 $y[n]$을 구하기 위해서는 $y[n]$ 바로 위에 다섯 줄을 더하면 된다.

FIR 필터의 콘볼루션 합: *numpy.convolve()*

```
import numpy as np
x = np.array([1, 3, 5, 4, 2])
h = np.array([1, 3, -2, 2])
y1 = np.convolve(h, x)
y2 = np.convolve(x, h)
print(y1)
print(y2)
```

실행 결과

[1 6 12 15 10 8 4 4]

[1 6 12 15 10 8 4 4]