*Python*과 함께 배우는 시스템 해석

제 2 장. 이산시간 선형 시불변 시스템의 시간 영역 해석
2-3. *IIR* 필터의 시간 영역 출력

박섭형

한림대학교 전자공학과

2014년 9월

배울 내용

- 1차 IIR 필터의 임펄스 응답
- 1차 IIR 필터의 계단 응답
- N차 IIR 필터의 출력: 차분방정식의 해
- scipy.signal 모듈의 lfilter() 함수를 이용한 IIR 필터의 수치적 계산

IIR 필터의 시간 영역 응답

IIR 필터는 현재의 출력을 계산할 때 입력 신호와 함께 과거에 출력된 신호를 사용하는 특징이 있다. 이렇게 출력된 신호가 다시 입력으로 사용되는 시스템을 궤환 (feedback) 시스템 또는 되먹임 시스템이라고 부른다.
일반적인 IIR 필터의 차분 방정식은 다음과 같다.

$$y[n] = \sum_{m=0}^{M} b_m x[n-m] - \sum_{k=1}^{N} a_k y[n-k]. \tag{2.55}$$

이 식에서 $a_N \neq 0$ 일 때, 이 차분 방정식의 차수는 N이 된다. 그리고 모든 $1 \leq k \leq N$인 k에 대해서 $a_k = 0$인 경우에 이 식은 FIR 필터의 차분 방정식과 동일하게 되기 때문에 FIR 필터는 IIR 필터의 특별한 경우라고 이해할 수 있다. 그러나 FIR 필터와는 달리 IIR 필터는 입력이 모든 n에 대해서 $x[n] = 0$이 되더라도 이전 출력 값이 존재하면 출력이 0이 되지 않는다. 이 출력은 다음 시점에서 다시 입력으로 작용하기 때문에 n이 증가하여도 IIR 필터의 출력은 0이 되지 않는다. 또한 이전 시점에서의 출력을 정확히 알고 있지 못한다면 정확한 출력을 계산할 수 없다. 따라서 IIR 필터의 출력을 계산할 때는 반드시 초기 조건으로 이전 시점의 출력을 알고 있어야 한다.

1차 IIR 필터의 시간 영역 응답

1차 IIR 필터의 입출력 관계식은 다음과 같은 차분 방정식으로 표현된다.

$$y[n] = b_0 x[n] - a_1 y[n-1]. \tag{2.56}$$

이 필터에 단위 임펄스, 단위 계단 함수가 입력되는 경우의 출력을 구해 보자.

1차 IIR 필터의 임펄스 응답

$\delta[n]$이 인가되는 경우의 출력. 단, $n < 0$일 때 $y[n] = 0$이라고 가정한다.

$$y[n] = b_0 \delta[n] - a_1 y[n-1]. \tag{2.57}$$

여기에서 $n = 0, 1, \cdots$를 차례로 대입하면서 $y[n]$을 직접 다음과 같이 구해보자.

- $n = 0$일 때, $y[0] = b_0\delta[0] - a_1 y[-1] = b_0$
- $n = 1$일 때, $y[1] = b_0\delta[1] - a_1 y[0] = -a_1 y[0] = -a_1 b_0$
- $n = 2$일 때, $y[2] = b_0\delta[2] - a_1 y[1] = -a_1 y[1] = (-a_1)^2 b_0$
- $n = 3$일 때, $y[3] = b_0\delta[3] - a_1 y[2] = -a_1 y[2] = (-a_1)^3 b_0$

  $\vdots$

n	$\cdots$	-1	0	1	2	3	4	$\cdots$
$\delta[n]$	$\cdots$	0	1	0	0	0	0	$\cdots$
$y[n]$	$\cdots$	0	b_0	$-a_1 b_0$	$(-a_1)^2 b_0$	$(-a_1)^3 b_0$	$(-a_1)^4 b_0$	$\cdots$

$$h[n] = (-a_1)^n b_0 u[n]. \tag{2.58}$$

1차 IIR 필터의 단위 계단 응답

단위 계단 함수가 입력될 때의 출력

$$y[n] = b_0 u[n] - a_1 y[n-1]. \tag{2.59}$$

- $n < 0$ 일 때, 모든 n에 대해서 $u[n] = 0$, $y[n-1] = 0$ 이므로 $y[n] = 0$.
- $n = 0$ 일 때, $y[0] = b_0 u[0] - a_1 y[-1] = b_0$.
- $n = 1$ 일 때, $y[1] = b_0 u[1] - a_1 y[0] = b_0 - a_1 b_0 = b_0(1 - a_1)$.
- $n = 2$ 일 때,
  $y[2] = b_0 u[2] - a_1 y[1] = b_0 - a_1 b_0(1 - a_1) = b_0(1 - a_1 + (-a_1)^2)$.
- $n = 3$ 일 때,

  $$y[3] = b_0 u[3] - a_1 y[2] = b_0 - a_1 b_0(1 - a_1 + (-a_1)^2) = b_0(1 - a_1 + (-a_1)^2 +$$

  $\vdots$

$$y[n] = b_0 \left(\sum_{k=0}^{n} (-a_1)^k \right) u[n]. \tag{2.60}$$

1차 IIR 필터의 단위 계단 응답

단위 계단 응답을 다음과 같이 임펄스 응답과 단위 계단 함수의 콘볼루션 합을 이용하여 구할 수도 있다.

$$
\begin{aligned}
y[n] &= \sum_{k=-\infty}^{\infty} x[k]h[n-k] \\
&= \sum_{k=-\infty}^{\infty} u[k](-a_1)^{n-k} b_0 u[n-k] \\
&= \sum_{k=0}^{\infty}(-a_1)^{n-k} b_0 u[n-k] \quad (m = n-k\text{로 치환}) \qquad (2.61) \\
&= \sum_{m=n}^{-\infty}(-a_1)^{m} b_0 u[m] \\
&= \sum_{m=-\infty}^{n} (-a_1)^{m} b_0 u[m] \\
&= \sum_{m=0}^{n}(-a_1)^{m} b_0 u[m].
\end{aligned}
$$

1차 IIR 필터 출력의 수치 계산

scipy.signal 모듈의 `lfilter()` 함수를 이용하면 된다. `lfilter()` 함수의 사용법은 다음과 같다.

```
from scipy.signal import lfilter
lfilter(b, a, x)
```

여기에서 b와 a는 각각 시스템 함수의 분자와 분모 다항식의 계수이고, x는 입력 신호이다. 이 함수를 이용해서 FIR 필터링을 수행할 수도 있는데 이 때는 a=[1]을 대입하면 된다.

IIR 필터의 초기 조건이 0이 아닌 경우에 `lfilter()` 함수의 마지막 매개 변수 `zi`에 초기 조건을 대입하고 다음과 같이 사용한다.

```
lfilter(b, a, x, zi)
```

1차 IIR 필터의 시간 영역 응답 예

Python 과 함께 배우는 시스템 해석

박섭형

1차 IIR 필터의 시간 영역 응답

N차 IIR 필터의 시간 영역 응답

IIR 필터의 출력 계산 예제

예제 *2.1* *(2.2)*

차분 방정식이 다음과 같이 주어지는 IIR 필터에서 $n < 0$일 때 $y[n] = 0$이라고 가정하자.

$$y[n] = 2x[n] + 0.9y[n-1]. \tag{2.62}$$

(1) 이 필터의 임펄스 응답을 직접 구하라.

(2) 이 필터의 임펄스 응답을 구하고 그래프를 그리는 Python 스크립트를 작성하라. 단, $0 \le n \le 60$으로 한정한다.

(3) 이 필터의 단위 계단 응답을 구하고 그 그래프를 그려라.

(4) 이 필터의 단위 계단 응답을 구하고 그래프를 그리는 Python 스크립트를 작성하라. 단, $0 \le n \le 60$으로 한정한다.

*1*차 *IIR* 필터의 시간 영역 응답 예

(1) 이 IIR 필터의 임펄스 응답은 $h[n] = 2(0.9)^n u[n]$ 이다.

(2) 임펄스 응답을 구하고 그래프를 그리는 Python 스크립트와 실행 결과는 각각 다음과 같다.

```
import matplotlib.pyplot as plt
import scipy.signal as sig
import numpy as np
def impseq(n0, n1, n2):
    assert (n0 <= n2) and (n1 <= n0) and (n1 <= n2)
    n = np.arange(n1,n2+1)
    x = (n == n0) * 1.0
    return n, x
n, x = impseq(0, 0, 60);   h = sig.lfilter([2], [1, -0.9], x)
plt.stem(n, h);   plt.xlim(0,60);  plt.xlabel("index $(n)$")
plt.tight_layout() ; plt.show()
```

1차 IIR 필터의 시간 영역 응답 예

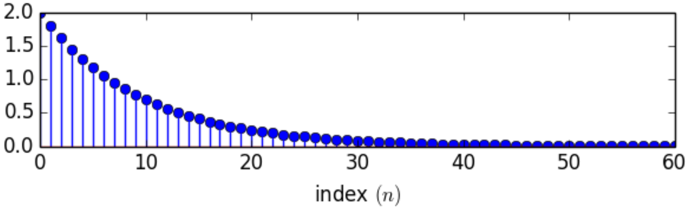

그림 *2.23:* $y[n] = 2x[n] + 0.9y[n-1]$로 표현되는 IIR 필터의 임펄스 응답 그래프.

1차 IIR 필터의 시간 영역 응답 예

(2) 이 IIR 필터의 단위 계단 응답은 다음과 같다.

- $n = 0$ 일 때, $y[0] = 2u[0] + 0.9y[-1] = 2$
- $n = 1$ 일 때, $y[1] = 2u[1] + 0.9y[0] = 2 + 0.9 \cdot 2 = 2(1 + 0.9)$
- $n = 2$ 일 때, $y[2] = 2u[2] + 0.9y[1] = 2 + 0.9 \cdot 2(1 + 0.9) = 2 + 0.9 \cdot 2 + 2 \cdot 0.9^2 = 2(1 + 0.9 + 0.9^2)$
- $n = 3$ 일 때, $y[3] = 2u[3] + 0.9y[2] = 2 + 0.9 \cdot 2(1 + 0.9 + 0.9^2) = 2(1 + 0.9 + 0.9^2 + 0.9^3)$

$\vdots$

$$y[n] = 2\left(\sum_{k=0}^{n} 0.9^k\right) u[n] = 2\frac{1 - 0.9^{n+1}}{1 - 0.9} u[n] = 20(1 - 0.9^{n+1})u[n]. \tag{2.63}$$

1차 IIR 필터의 시간 영역 응답 예

(3) 계단 응답을 구하고 그래프를 그리는 Python 스크립트와 실행 결과는 각각 다음과 같다.

```
import matplotlib.pyplot as plt
import scipy.signal as sig
import numpy as np
plt.figure(figsize=(6,2))
def stepseq(n0, n1, n2):
    assert (n0 <= n2) and (n1 <= n0) and (n1 <= n2)
    n = np.arange(n1,n2+1)
    x = (n >= n0) * 1.0
    return n, x
n, x = stepseq(0, 0, 60);    h = sig.lfilter([2], [1, -0.9], x)
plt.stem(n, h);    plt.xlim(0,60);    plt.xlabel("index $(n)$")
plt.tight_layout();    plt.show()
```

1차 IIR 필터의 시간 영역 응답 예

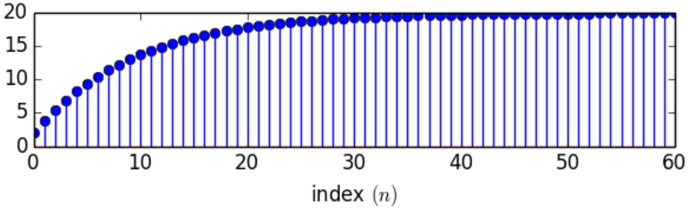

그림 *2.24:* $y[n] = 2x[n] + 0.9y[n-1]$로 표현되는 IIR 필터의 단위 계단 응답 그래프.

1차 *IIR* 필터의 일반 신호에 대한 출력

입출력 관계식이 표현되는 1차 IIR 필터에 일반적인 신호 $x[n]$이 입력될 때의 출력을 구해 보자. 단, $n < 0$일 때, $x[n] = 0$이라고 가정한다.

$$y[n] = b_0 x[n] - a_1 y[n-1]. \tag{2.64}$$

이 식에 $n = 0, 1, 2, \cdots$를 차례로 대입하면서 출력 $y[n]$을 구하면 다음과 같은 식을 얻을 수 있다.

- $n < 0$일 때, 모든 n에 대해서 $x[n] = 0$, $y[n-1] = 0$이므로 $y[n] = 0$.
- $n = 0$일 때, $y[0] = b_0 x[0] - a_1 y[-1] = b_0 x[0]$.
- $n = 1$일 때, $y[1] = b_0 x[1] - a_1 y[0] = b_0 x[1] - a_1 b_0 x[0] = b_0(x[1] - a_1 x[0])$.

*1*차 *IIR* 필터의 일반 신호에 대한 출력

- $n = 2$ 일 때,

$$\begin{aligned} y[2] &= b_0 x[2] - a_1 y[1] = b_0 x[2] - a_1 b_0 (x[1] - a_1 x[0]) \\ &= b_0 (x[2] - a_1 x[1] + (-a_1)^2 x[0]). \end{aligned}$$

- $n = 3$ 일 때,

$$\begin{aligned} y[3] &= b_0 x[3] - a_1 y[2] = b_0 x[3] - a_1 b_0 (x[2] + a_1 x[1] + (-a_1)^2 x[0]) \\ &= b_0 x[3] + b_0(-a_1) x[2] + b_0 (-a_1)^2 x[1] + b_0 (-a_1)^3 x[0] \\ &= \sum_{k=0}^{3} b_0 (-a_1)^k x[3-k]. \end{aligned}$$

따라서 입력 $x[n]$ 에 대한 일차 IIR 필터의 출력은 다음과 같다.

$$y[n] = \sum_{k=0}^{n} b_0 (-a_1)^k x[n-k] = h[n] * x[n], \qquad (2.65)$$

여기에서 $h[n] = b_0(-a_1)^n$ 은 IIR 필터의 임펄스 응답이다.

N차 *IIR* 필터의 출력

N차 IIR 필터의 입출력 사이의 관계식은 다음과 같은 차분 방정식으로 표현된다.

$$y[n] = \sum_{m=0}^{M} b_m x[n-m] - \sum_{k=1}^{N} a_k y[n-k]. \tag{2.66}$$

이 IIR 필터의 출력 $y[n]$을 구하는 것은 이 차분 방정식의 해를 구하는 것과 같다. 일반적인 차분 방정식의 해는 다음과 같다.

$$y[n] = y_h[n] + y_p[n], \tag{2.67}$$

여기에서 $y_h[n]$은 동차 해(homogeneous solution) 또는 일반해(general solution)라고 하며, $y_p[n]$은 특별 해(particular solution)이라고 한다.

차분 방정식의 동차 해

$y_h[n]$은 모든 n에 대해서 $x[n] = 0$일 때의 해, 즉 다음 방정식의 해이다.

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k]. \tag{2.68}$$

이 방정식을 동차 방정식(homogeneous equation)이라고 한다. $a_0 = 1$이라고 두면 식 (2.68)은 다음과 같이 쓸 수 있다.

$$\sum_{k=0}^{N} a_k y[n-k] = 0. \tag{2.69}$$

이 방정식의 해는 다음과 같다는 것이 잘 알려져 있다.

$$y[n] = c\lambda^n, \tag{2.70}$$

여기에서 c와 λ는 복소수이다. 이 해를 식 (2.69) 에 대입하면 다음 식을 얻는다.

$$\sum_{k=0}^{N} a_k c\lambda^{n-k} = 0. \tag{2.71}$$

차분 방정식의 동차 해

$$c\left(a_0\lambda^n + a_1\lambda^{n-1} + \cdots + a_N\lambda^{n-N}\right) = 0. \tag{2.72}$$

$$c\lambda^{n-N}\left(a_0\lambda^N + a_1\lambda^{N-1} + \cdots + a_N\right) = 0. \tag{2.73}$$

양 변을 $c\lambda^{n-N}$으로 나누면 다음 식을 얻는다.

$$a_0\lambda^N + a_1\lambda^{N-1} + \cdots + a_N = 0. \tag{2.74}$$

이 식을 차분 방정식의 특성 방정식(characteristic equation)이라고 부른다. 이 방정식은 λ에 관한 N차 방정식이므로 이 식을 만족하는 해는 N개가 존재한다. 이 방정식이 N개의 서로 다른 해를 갖는다고 가정하고, 이 해를 각각 $\lambda_1, \lambda_2, \cdots, \lambda_N$이라고 하면, 동차 해는 다음과 같이 주어진다.

$$y_h[n] = c_1\lambda_1^n + c_2\lambda_2^n + \cdots + c_N\lambda_N^n. \tag{2.75}$$

특성 방정식이 다중 해를 갖는 경우에 동차 해는 다음과 같이 구한다. 예를 들어서, $\lambda_i = \lambda_{i+1} = \cdots = \lambda_{i+m}$ 인 경우에, 즉, $(m+1)$ 개의 다중 해가 있는 경우에 $c_i\lambda_i^n + c_{i+1}\lambda_{i+1}^n + \cdots + c_{i+m}\lambda_{i+m}^n$을 $c_i\lambda_i^n + c_{i+1}n\lambda_{i+1}^n + \cdots + c_{i+m}n^m\lambda_{i+m}^n$으로 대치한다.

차분 방정식의 동차 해

예제 2.2 (2.3)

다음 차분 방정식의 동차 해를 구하라.

$$y[n] = 2y[n-1] + 2y[n-2], \quad y[0] = 1, y[1] = 2. \tag{2.76}$$

먼저, 이 방정식의 특성 방정식은 다음과 같다.

$$\lambda^2 - 2\lambda - 2 = 0. \tag{2.77}$$

이 식을 만족하는 λ는 $\lambda_1 = 1 + \sqrt{3}, \lambda_2 = 1 - \sqrt{3}$이다. 따라서 이 차분 방정식의 동차 해는 다음과 같다.

$$y_h[n] = c_1(1 + \sqrt{3})^n + c_2(1 - \sqrt{3})^n. \tag{2.78}$$

여기에서 초기 조건을 이용하면 다음 두 식을 얻을 수 있다.

$$y_h[0] = c_1 + c_2 = 1, \tag{2.79}$$

$$y_h[1] = c_1(1 + \sqrt{3}) + c_2(1 - \sqrt{3}) = 2. \tag{2.80}$$

차분 방정식의 동차 해

이 두 식으로부터 c_1 과 c_2 를 구하면 $c_1 = \frac{3+\sqrt{3}}{6}$ 과 $c_2 = \frac{3-\sqrt{3}}{6}$ 를 얻는다. 따라서 차분 방정식의 동차 해는 최종적으로 다음과 같이 쓸 수 있다.

$$\begin{aligned} y_h[n] &= \frac{3+\sqrt{3}}{6}(1+\sqrt{3})^n + \frac{3-\sqrt{3}}{6}(1-\sqrt{3})^n \\ &= \frac{\sqrt{3}}{6}(1+\sqrt{3})^{n+1} - \frac{\sqrt{3}}{6}(1-\sqrt{3})^{n+1}. \end{aligned} \tag{2.81}$$

차분 방정식의 동차 해

예제 *2.3 (2.4)*

다음 차분 방정식의 동차 해를 구하라.

$$y[n] = 4y[n-1] - 4y[n-2], \quad y[0] = 3, y[1] = 8. \tag{2.82}$$

$$\lambda^2 - 4\lambda + 4 = 0. \tag{2.83}$$

$\lambda_1 = \lambda_2 = 2$(중근).

$$y_h[n] = c_1 2^n + c_2 n 2^n. \tag{2.84}$$

여기에서 초기 조건을 이용하면 다음 두 식을 얻을 수 있다.

$$y_h[0] = c_1 + c_2 \cdot 0 \cdot 2^0 = 3, \tag{2.85}$$

$$y_h[1] = c_1 2^1 + c_2 \cdot 1 \cdot 2^1 = 8. \tag{2.86}$$

$c_1 = 3$과 $c_2 = 1$.동차 해는 최종적으로 다음과 같이 쓸 수 있다.

$$y_h[n] = 3 \cdot 2^n + n 2^n. \tag{2.87}$$

차분 방정식의 특별 해

차분 방정식의 특별 해는 입력 신호 $x[n]$ 에 따라서 달라지며, $x[n]$ 의 몇 가지 형태에 대해서는 특별 해를 구하는 방법이 알려져 있으나 일반적으로 특별 해를 구하는 것은 쉽지 않다.

예제 *2.4* *(2.5)*

다음 1 차 차분 방정식의 해를 구하라.

$$y[n] = 0.5y[n-1] + 3, \quad y[0] = 2. \tag{2.88}$$

단, 초기 정지 상태를 가정하라.

이 필터는 $x[n] = 3u[n]$ 인 1 차 IIR 필터로 생각할 수 있다.
먼저, 이 방정식 $y[n] - 0.5y[n-1] = 0$ 의 특성 방정식은 다음과 같다.

$$\lambda - 0.5 = 0. \tag{2.89}$$

이 식을 만족하는 λ는 $\lambda = \dfrac{1}{2}$ 이므로, 이 차분 방정식의 동차 해는 다음과 같다.

차분 방정식의 동차 해

$$y_h[n] = c\left(\frac{1}{2}\right)^n. \tag{2.90}$$

$x[n] = 3$일 경우의 특수 해를 $y_p[n] = \beta$라고 두자. 그러면 $y[n]$은 다음과 같다.

$$y[n] = y_h[n] + y_p[n] = c\left(\frac{1}{2}\right)^n + \beta \tag{2.91}$$

이 식을 식 (2.88)에 대입하면, 다음 식을 얻는다.

$$c\left(\frac{1}{2}\right)^n + \beta = \frac{1}{2}\left[c\left(\frac{1}{2}\right)^{n-1} + \beta\right] + 3 \tag{2.92}$$

$$= c\left(\frac{1}{2}\right)^n + \frac{1}{2}\beta + 3 \tag{2.93}$$

이 식으로부터 $\beta = \frac{1}{2}\beta + 3$이므로, $\beta = 6$이다. $\beta = 6$을 식 (2.91)에 대입하고, $y[0] = 2$를 이용하면 $y[0] = c + \beta = c + 6 = 2$이므로, $c = -4$이다.
따라서 $y[n]$은 다음과 같다.

$$y[n] = -4\left(\frac{1}{2}\right)^n u[n] + 6u[n]. \tag{2.94}$$

차분 방정식의 동차 해

이 문제를 반복적 계산 방법으로 다시 풀어보자.

i) $n = 1$: $y[1] = \frac{1}{2} \cdot y[0] + 3$

ii) $n = 2$:

$$y[2] = \frac{1}{2} \cdot y[1] + 3 = \frac{1}{2}\left(\frac{1}{2} \cdot y[0] + 3\right) + 3 = \left(\frac{1}{2}\right)^2 y[0] + \frac{1}{2} \cdot 3 + 3$$

iii) $n = 3$:

$$\begin{aligned} y[3] &= \frac{1}{2} \cdot y[2] + 3 \\ &= \frac{1}{2}\left[\left(\frac{1}{2}\right)^2 y[0] + \frac{1}{2} \cdot 3 + 3\right] + 3 \\ &= \left(\frac{1}{2}\right)^3 y[0] + 3\left[\left(\frac{1}{2}\right)^2 + \frac{1}{2} + 1\right]. \end{aligned}$$

차분 방정식의 동차 해

따라서 $n \geq 0$ 일 때, $y[n]$ 은 다음과 같이 구할 수 있다.

$$\begin{aligned} y[n] &= \left(\frac{1}{2}\right)^n y[0] + 3\left[\left(\frac{1}{2}\right)^{n-1} + \left(\frac{1}{2}\right)^{n-2} + \cdots + \frac{1}{2} + 1\right] \\ &= \left(\frac{1}{2}\right)^n y[0] + 3\sum_{k=0}^{n-1}\left(\frac{1}{2}\right)^k \\ &= \left(\frac{1}{2}\right)^n y[0] + 3\frac{1-\left(\frac{1}{2}\right)^n}{1-\frac{1}{2}} \\ &= \left(\frac{1}{2}\right)^n y[0] + 6 - 6\left(\frac{1}{2}\right)^n \\ &= (y[0]-6)\left(\frac{1}{2}\right)^n + 6 \\ &= -4\left(\frac{1}{2}\right)^n + 6. \end{aligned} \tag{2.95}$$

즉, $y[n]$ 은 다음과 같다.

$$y[n] = -4\left(\frac{1}{2}\right)^n u[n] + 6u[n]. \tag{2.96}$$

IIR 필터의 출력 구하기 예제

예제 *2.5* *(2.6)*

차분 방정식이 다음과 같이 주어지는 IIR 필터에서 $n < 0$일 때 $y[n] = 0$이라고 가정하자.

$$y[n] = 0.5x[n] + 0.8y[n-1]. \tag{2.97}$$

(1) 반복 계산 방법으로 이 필터의 임펄스 응답을 구하라.

(2) 반복 계산 방법으로 이 필터의 단위 계단 응답을 구하라.

(3) 콘볼루션 합을 이용하여 이 필터의 단위 계단 응답을 구하라.

IIR 필터의 출력 구하기 예제

(1) 반복 계산 방법을 이용하여 임펄스 응답을 다음과 같이 구할 수 있다.

- $n = 0$ 일 때, $y[0] = 0.5\delta[0] + 0.8y[-1] = 0.5$
- $n = 1$ 일 때, $y[1] = 0.5\delta[1] + 0.8y[0] = 0.8y[0] = 0.8 \cdot 0.5$
- $n = 2$ 일 때, $y[2] = 0.5\delta[2] + 0.8y[1] = 0.8y[1] = (0.8)^2 0.5$
- $n = 3$ 일 때, $y[3] = 0.5\delta[3] + 0.8y[2] = 0.8y[2] = (0.8)^3 0.5$

$\vdots$

따라서 이 IIR 필터의 임펄스 응답 $h[n]$은 다음과 같이 쓸 수 있다.

$$h[n] = 0.5(0.8)^n u[n]. \tag{2.98}$$

IIR 필터의 출력 구하기 예제

(2) 반복 계산 방법을 이용하여 단위 계단 응답을 다음과 같이 구할 수 있다.

- $n = 0$일 때, $y[0] = 0.5u[0] + 0.8y[-1] = 0.5$
- $n = 1$일 때, $y[1] = 0.5u[1] + 0.8y[0] = 0.5 + 0.8 \cdot 0.5 = 0.5(1 + 0.8)$
- $n = 2$일 때,

$$\begin{aligned} y[2] &= 0.5u[2] + 0.8y[1] \\ &= 0.5 + 0.8 \cdot 0.5(1 + 0.8) = 0.5 + 0.8 \cdot 0.5 + 0.5 \cdot (0.8)^2 \\ &= 0.5\left[1 + 0.8 + (0.8)^2\right] \end{aligned}$$

- $n = 3$일 때,

$$\begin{aligned} y[3] &= 0.5u[3] + 0.8y[2] \\ &= 0.5 + 0.8 \cdot 0.5\left[1 + 0.8 + (0.8)^2\right] \\ &= 0.5\left[1 + 0.8 + (0.8)^2 + (0.8)^3\right] \end{aligned}$$

$\vdots$

따라서, 이 IIR 필터의 단위 계단 응답의 일반식은 다음과 같이 구할 수 있다.

$$\begin{aligned} y[n] &= 0.5\left(\sum_{k=0}^{n}(0.8)^k\right)u[n] \\ &= 0.5\frac{1-(0.8)^{n+1}}{1-0.8}u[n] = 2.5\left[1-(0.8)^{n+1}\right]u[n]. \end{aligned}$$

IIR 필터의 출력 구하기 예제

(3) 콘볼루션 합을 이용하여 단위 계단 응답을 다음과 같이 구할 수 있다.

(i) $n < 0$일 때, $y[n] = 0$

(ii) $n \geq 0$일 때

$$y[n] = u[n] * h[n] = \sum_{k=0}^{n} h[k] = \sum_{k=0}^{n} 0.5(0.8)^k$$

$$= 0.5\frac{(1-0.8)^{n+1}}{1-0.8} = 2.5\left\{1-(0.8)^{n+1}\right\}$$

따라서 $y[n] = 2.5\left[1-(0.8)^{n+1}\right]u[n]$이다.

IIR 필터의 출력 구하기 예제

예제 2.6 (2.7)

다음 2 차 차분 방정식의 해를 구하라.

$$y[n] = 2.5y[n-1] - y[n-2] + 3, \quad y[0] = 2, y[1] = 4. \tag{2.99}$$

IIR 필터의 출력 구하기 예제

이 필터는 $x[n] = 3$인 2차 IIR 필터로 생각할 수 있다.
먼저, 방정식 $y[n] - 2.5y[n-1] + y[n-2] = 0$의 특성 방정식은 다음과 같다.

$$\lambda^2 - 2.5\lambda + 1 = 0. \tag{2.100}$$

$$2\lambda^2 - 5\lambda + 2 = 0. \tag{2.101}$$

$$(2\lambda - 1)(\lambda - 2) = 0. \tag{2.102}$$

이 식을 만족하는 λ는 $\lambda_1 = \frac{1}{2}, \lambda_2 = 2$이므로, 이 차분 방정식의 동차 해는 다음과 같다.

$$y_h[n] = c_1 \left(\frac{1}{2}\right)^n + c_2 2^n. \tag{2.103}$$

$x[n] = 3$일 경우의 특수 해를 $y_p[n] = \beta$라고 두자. 그러면 $y[n]$은 다음과 같다.

$$y[n] = y_h[n] + y_p[n] = c_1 \left(\frac{1}{2}\right)^n + c_2 2^n + \beta \tag{2.104}$$

$y[n]$을 식 (2.99)의 $y[n]$에 대입하면 다음 식을 얻는다.

$$\begin{aligned} c_1\left(\frac{1}{2}\right)^n + c_2 2^n + \beta &= 2.5\left[c_1\left(\frac{1}{2}\right)^{n-1} + c_2 2^{n-1} + \beta\right] \\ &\quad - \left[c_1\left(\frac{1}{2}\right)^{n-2} + c_2 2^{n-2} + \beta\right] + 3 \\ &= \left[5c_1\left(\frac{1}{2}\right)^n + \frac{5}{4}c_2 2^n + \frac{5}{2}\beta\right] \\ &\quad - \left[4c_1\left(\frac{1}{2}\right)^n + \frac{1}{4}c_2 2^n + \beta\right] + 3 \\ &= c_1\left(\frac{1}{2}\right)^n + c_2 2^n + \frac{3}{2}\beta + 3 \end{aligned} \tag{2.105}$$

IIR 필터의 출력 구하기 예제

이 식으로부터 $\beta = \frac{3}{2}\beta + 3$, 즉 $\beta = -6$이므로, $y[n]$은 다음과 같이 쓸 수 있다.

$$y[n] = c_1\left(\frac{1}{2}\right)^n + c_2 2^n - 6. \tag{2.106}$$

초기 조건 $y[0] = 2, y[1] = 4$를 이용하여 다음 관계식을 얻을 수 있다.

$$y[0] = c_1 + c_2 - 6 = 2, \tag{2.107}$$

$$y[1] = \frac{1}{2}c_1 + 2c_2 - 6 = 4. \tag{2.108}$$

이 두 식으로부터, $c_1 = 4$, $c_2 = 4$를 구할 수 있다.
따라서, $y[n]$은 다음과 같다.

$$y[n] = 4\left(\frac{1}{2}\right)^n u[n] + 4 \cdot 2^n u[n] - 6u[n]. \tag{2.109}$$

scipy.signal 패키지의 *lfilter()* 함수를 이용한 *IIR* 필터의 출력 구하기

Python과 scipy.signal 패키지의 `lfilter()` 함수를 이용하면 간단하게 IIR 필터의 `lfilter()` 함수의 사용법은 다음과 같다.

```
from scipy.signal import lfilter
lfilter(b, a, x)
```

여기에서 `b`와 `a`는 각각 시스템 함수의 분자와 분모 다항식의 계수이고, `x`는 입력 신호이다. 이 함수를 이용해서 FIR 필터링을 수행할 수도 있는데 이 때는 `a=[1]`을 대입하면 된다.

IIR 필터의 출력 구하기 예제

IIR 필터의 초기 조건이 0이 아닌 경우에 `lfilter()` 함수의 마지막 매개 변수 `zi`에 초기 조건을 대입하고 다음과 같이 사용한다.

```
lfilter(b, a, x, zi)
```

IIR 필터의 출력 구하기 예제

예제 2.6에 주어진 문제를 Python을 이용해서 풀어 보자. 입출력 관계식은 다음과 같다.

$$y[n] = 0.5x[n] + 0.8y[n-1]. \tag{2.110}$$

임펄스 응답과 단위계단 응답을 구하고 그래프를 그리는 Python 스크립트는 다음과 같다.

```
import numpy as np
from scipy.signal import lfilter
import matplotlib.pyplot as plt
a=[1,-0.8];      b=[0.5]
imp=np.array([1,0,0,0,0,0,0,0,0,0,0])
ustep = np.ones(11)
h=lfilter(b,a,imp);  sr = lfilter(b,a,ustep)
plt.subplot(211);    plt.stem(np.arange(len(imp)), h)
plt.subplot(212);    plt.stem(np.arange(len(ustep)), sr)
plt.tight_layout();  plt.show()
```

IIR 필터의 출력 구하기 예제

이 스크립트를 실행하면 화면에서 다음 그래프를 볼 수 있다.

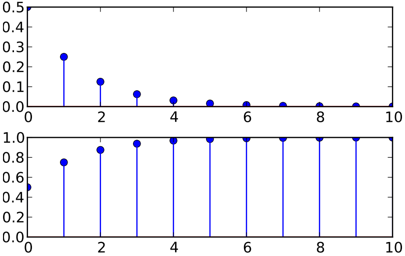

그림 *2.25:* 예제 2.6의 임펄스 응답(상)과 단위계단 응답(하) 그래프.

IIR 필터의 출력 구하기 예제

예제 2.7 *(2.8: lfilter() 함수를 이용한 IIR 필터의 임펄스 응답 계산)*

입출력 관계식이 다음과 같이 주어지는 IIR 필터를 가정하자

$$y[n] = 0.8y[n-1] - 0.5y[n-2] + x[n].$$

이 IIR 필터의 임펄스 응답을 구하고 그래프를 그리는 Python 스크립트를 작성하라. 단, 초기 조건이

(a) $y[-2] = 0, y[-1] = 0$

(b) $y[-2] = 1, y[-1] = 2$

인 두 가지 경우의 응답을 구하여 비교하라.

IIR 필터의 출력 구하기 예제

Python 스크립트는 다음과 같다.

```
import numpy as np
from scipy.signal import lfilter
import matplotlib.pyplot as plt
a = [1, -0.8, 0.5];    b=[1]
n=np.arange(30)
imp = np.zeros(30);    imp[0]=1 # impulse
h1 = lfilter(b, a, imp)
h2 = lfilter(b, a, imp, zi=[1,2])
plt.subplot(211)
plt.stem(np.arange(h1.size), h1)
plt.axis([0, 30, -2, 4])
plt.subplot(212)
plt.stem(np.arange(h2[0].size), h2[0])
plt.show()
```

이 예제에서 lfilter 사용시 초기 조건이 있는 경우와 없는 경우에 return 값의

IIR 필터의 출력 구하기 예제

다음 그림은 Python을 이용하여 구한 임펄스 응답의 그래프이다.

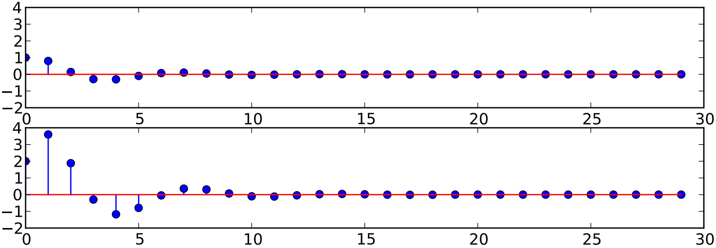

그림 *2.26:* 예제에 정의된 시스템의 임펄스 응답. (위: 초기 정지 조건의 경우, 아래: 초기 정지 조건이 아닌 경우)

이 그림으로부터 동일한 IIR 필터라도 초기 조건이 다르면 임펄스 응답이 다르다는 사실을 확인할 수 있다.