신호 및 시스템

제 *8* 장. 선형 시불변 시스템의 고유신호와 주파수 응답

박섭형

한림대학교

2021학년도 1학기

배울 내용

- 선형 변환 행렬의 고유벡터와 고윳값
- 선형 시불변 시스템의 고유신호(또는 형태 불변 신호, eigensignal): 복소지수 신호
- 선형 시불변 시스템의 고윳값: 임펄스 응답의 푸리에 변환
- 주파수 응답

선형 시스템의 행렬 표현

행렬은 하나의 벡터 공간을 또 다른 벡터 공간으로 변환하는 사상이다. **x**와 **b**가 $n \times 1$ 벡터라고 하고, **A**가 다음 식을 만족하는 $n \times n$ 행렬로 표현되는 선형 변환이라고 하자.

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{8.1}$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1(n-1)} & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2(n-1)} & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{(n-1)1} & a_{(n-1)2} & \cdots & a_{(n-1)(n-1)} & a_{(n-1)n} \\ a_{n1} & a_{n2} & \cdots & a_{n(n-1)} & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}. \tag{8.2}$$

그러면 **A**는 n 차원 벡터 공간을 n 차원 벡터 공간으로 사상하는 역할을 한다.

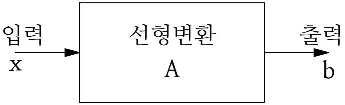


그림 *8.1:* 선형 시스템 **A**의 입력 **x**와 출력 **b**.

선형 변환 행렬의 고유벡터와 고윳값 *(1)*

정의 *8.1 (*선형 변환 A의 고유벡터와 고윳값*)*

다음 조건을 만족하는 스칼라 λ가 존재한다면, 널 벡터가 아닌 **x**는 선형 변환 **A**의 고유벡터 (eigenvector)라고 정의하고, λ는 고유벡터 **x**에 대응하는 고윳값 (eigenvalue)이라고 정의한다.

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}. \tag{8.3}$$

- 선형 변환의 고유벡터 : 선형 변환 **A**에 입력되어도 크기만 변할 뿐 방향은 변하지 않는 벡터
- 선형 변환의 고윳값 : 고유벡터가 선형 변환 **A**에 입력되었을 때 출력 벡터와 입력 벡터의 크기 비

선형 변환 행렬의 고유벡터와 고윳값 *(2)*

A가 정방 행렬일 때, 식 (8.3)은 다음과 같이 쓸 수 있다.

$$\mathbf{Ax} - \lambda\mathbf{x} = \mathbf{0}.$$

$$\mathbf{Ax} - \lambda\mathbf{I}x = \mathbf{0}, \tag{8.4}$$

여기에서 **I**는 단위 행렬이고, **0**은 모든 원소가 0인 벡터이다. 그러면 식 (8.4)는 다음과 같이 쓸 수 있다.

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0}. \tag{8.5}$$

참고 :

$$\mathbf{Ix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \mathbf{x}$$

선형 변환 행렬의 고유벡터와 고윳값 *(3)*

만약에 $(\mathbf{A} - \lambda\mathbf{I})^{-1}$ 이 존재하면, 식 (8.5)의 양변에 $(\mathbf{A} - \lambda\mathbf{I})^{-1}$ 을 곱할 수 있어서 $\mathbf{x} = \mathbf{0}$이 된다. 따라서 $\mathbf{0}$이 아닌 고유벡터가 존재하기 위해서는 $(\mathbf{A} - \lambda\mathbf{I})^{-1}$ 이 존재하지 않아야 하며 $(\mathbf{A} - \lambda\mathbf{I})^{-1}$ 이 존재하지 않기 위해서는 다음 조건을 만족해야 한다.

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0. \tag{8.6}$$

이 식을 행렬 $\mathbf{A}$의 특성 방정식이라고 부르며, 이 특성 방정식의 해가 행렬 $\mathbf{A}$의 고윳값이 된다.

선형 변환 행렬의 고유벡터와 고윳값 *(4)*

예제 *8.1*

선형 변환 $\mathbf{A} = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$의 고유벡터와 고윳값을 구하라.

$$\begin{aligned} \det(\mathbf{A} - \lambda \mathbf{I}) &= \det\left(\begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \right) \\ &= \det \begin{bmatrix} 3-\lambda & 1 \\ 1 & 3-\lambda \end{bmatrix} \\ &= (3-\lambda)(3-\lambda) - 1 \cdot 1 \\ &= \lambda^2 - 6\lambda + 9 - 1 \\ &= \lambda^2 - 6\lambda + 8 = (\lambda-2)(\lambda-4). \end{aligned} \tag{8.7}$$

특성 방정식 $(\lambda-2)(\lambda-4) = 0$의 해인 $\lambda = 2$ 또는 $\lambda = 4$이 고윳값이 된다.

선형 변환 행렬의 고유벡터와 고윳값 *(5)*

이 행렬의 고유벡터를 $\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^T$ 라고 하자.
첫 번째 고윳값 $\lambda = 2$에 대응하는 고유벡터는 다음과 같이 구할 수 있다.

$$\begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 2 \begin{bmatrix} x \\ y \end{bmatrix}. \tag{8.8}$$

이 행렬식을 풀어 쓰면 다음 연립 방정식을 얻는다.

$$3x + y = 2x, \tag{8.9}$$

$$x + 3y = 2y. \tag{8.10}$$

이 방정식을 풀면, $x = -y$가 되며 연립 방정식을 만족하는 벡터는 무수히 많은 것을 알 수 있다. 이 가운데 $x \neq 0$인 임의의 x를 선택하면 된다. 예를 들어서, $x = 1$을 선택하면, $\lambda = 2$에 대응하는 고유벡터 $\begin{bmatrix} 1 & -1 \end{bmatrix}^T$를 구할 수 있다.

선형 변환 행렬의 고유벡터와 고윳값 *(6)*

고윳값 $\lambda = 4$에 대응하는 고유벡터는 다음과 같이 구할 수 있다.

$$\begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 4 \begin{bmatrix} x \\ y \end{bmatrix}. \tag{8.11}$$

이 행렬식을 풀어 쓰면 다음 연립 방정식을 얻는다.

$$3x + y = 4x, \tag{8.12}$$

$$x + 3y = 4y. \tag{8.13}$$

이 방정식으로부터 $x = y$가 되는 것을 알 수 있는데, $x \neq 0$인 임의의 x를 선택하면 된다. 예를 들어서, $x = 1$을 선택하면,

$\begin{bmatrix} 1 & 1 \end{bmatrix}^{\mathsf{T}}$가 $\lambda = 4$에 대응하는 고유벡터가 된다.

선형 변환 행렬의 고유벡터와 고윳값 *(7)*

예제 *8.2*

예제 8.1의 변환행렬로 $\mathbf{x}_1 = \begin{bmatrix}1 & 1\end{bmatrix}^\intercal$, $\mathbf{x}_2 = \begin{bmatrix}1 & -1\end{bmatrix}^\intercal$, $\mathbf{x}_3 = \begin{bmatrix}2 & 2\end{bmatrix}^\intercal$, $\mathbf{x}_4 = \begin{bmatrix}2 & 0\end{bmatrix}^\intercal$, 그리고 $\mathbf{x}_5 = \begin{bmatrix}0 & 2\end{bmatrix}^\intercal$ 를 각각 변환한 결과 $\mathbf{y}_k$ 를 구하고, 각 벡터가 이 선형 변환의 고유벡터인지 아닌지를 판단하라.

$$\mathbf{y}_1 = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \cdot 1 + 1 \cdot 1 \\ 1 \cdot 1 + 3 \cdot 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix} = 4 \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 4\mathbf{x}_1,$$

$$\mathbf{y}_2 = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 3 \cdot 1 + 1 \cdot (-1) \\ 1 \cdot 1 + 3 \cdot (-1) \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ -1 \end{bmatrix} = 2\mathbf{x}_2,$$

$$\mathbf{y}_3 = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \cdot 2 + 1 \cdot 2 \\ 1 \cdot 2 + 3 \cdot 2 \end{bmatrix} = \begin{bmatrix} 8 \\ 8 \end{bmatrix} = 4 \begin{bmatrix} 2 \\ 2 \end{bmatrix} = 4\mathbf{x}_3,$$

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ 는 고유벡터이다.

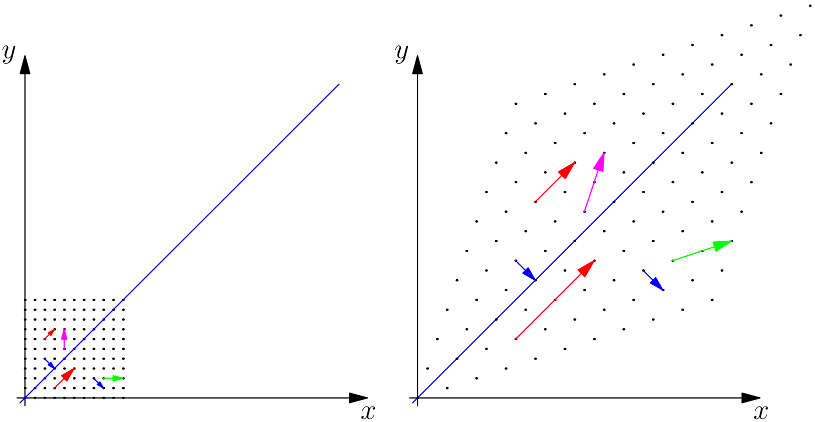
선형 변환 행렬의 고유벡터와 고윳값 *(8)*

$$\mathbf{y}_4 = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \cdot 2 + 1 \cdot 0 \\ 1 \cdot 2 + 3 \cdot 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \neq \lambda \mathbf{x}_4,$$

$$\mathbf{y}_5 = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \cdot 0 + 1 \cdot 2 \\ 1 \cdot 0 + 3 \cdot 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 6 \end{bmatrix} \neq \lambda \mathbf{x}_5.$$

$\mathbf{x}_4, \mathbf{x}_5$ 는 고유벡터가 아니다.

고유벡터의 물리적인 의미



(a) 변환 전의 벡터들. (b) 변환 후의 벡터들.

그림 8.2: 예제 8.1의 변환 행렬에 의해서 변환하는 벡터들의 예. 두 그림에서 변환 전과 후에 방향 변화가 없는 벡터들이 고유벡터들이다. 출처: 위키피디아, `http://en.wikipedia.org/wiki/Eigen_vector`.

선형 변환의 고유벡터 확인 예

예제 8.3

행렬 $\mathbf{A} = \begin{bmatrix} 3 & 4 \\ 4 & 3 \end{bmatrix}$ 로 표시되는 선형 변환에서 입력 $\mathbf{x}_1 = [2 \;\; 2]^\mathsf{T}$, $\mathbf{x}_2 = [2 \;\; -2]^\mathsf{T}$, $\mathbf{x}_3 = [2 \;\; 1]^\mathsf{T}$ 에 대한 출력 $\mathbf{y}_1$, $\mathbf{y}_2$, $\mathbf{y}_3$ 를 각각 구해서 각 입력 벡터가 고유벡터인지 아닌지를 판정하라.

$$\mathbf{y}_1 = \begin{bmatrix} 3 & 4 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \cdot 2 + 4 \cdot 2 \\ 4 \cdot 2 + 3 \cdot 2 \end{bmatrix} = \begin{bmatrix} 14 \\ 14 \end{bmatrix} = 7 \begin{bmatrix} 2 \\ 2 \end{bmatrix} = 7\mathbf{x}_1,$$

$$\mathbf{y}_2 = \begin{bmatrix} 3 & 4 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ -2 \end{bmatrix} = \begin{bmatrix} 3 \cdot 2 - 4 \cdot 2 \\ 4 \cdot 2 - 3 \cdot 2 \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \end{bmatrix} = -1 \begin{bmatrix} 2 \\ -2 \end{bmatrix} = -1\mathbf{x}_2,$$

$$\mathbf{y}_3 = \begin{bmatrix} 3 & 4 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \cdot 2 + 4 \cdot 1 \\ 4 \cdot 2 + 3 \cdot 1 \end{bmatrix} = \begin{bmatrix} 10 \\ 11 \end{bmatrix} \neq \lambda\mathbf{x}_3.$$

$\mathbf{x}_1$ 과 $\mathbf{x}_2$ 는 고유벡터이고, $\mathbf{x}_3$ 는 고유벡터가 아니다.

행렬의 고유벡터와 고윳값을 *Python*으로 구하기 *(1)*

`numpy` 모듈은 선형대수와 관련된 함수들을 지원하는 `linalg` 모듈을 포함하고 있다. 이 모듈 안에 `eig()` 함수는 행렬의 고윳값과 고유벡터를 계산하는 함수이다.

다음은 2×2행렬 **A**의 고윳값과 고유벡터를 구하는 Python 스크립트의 예이다.

```
import numpy as np
from numpy import linalg as LA
A = np.array([[3,4],[4,3]])
w, v = LA.eig(A)
print('eigen values are ', w)
print('The first eigen vector is ', v[:,0], \
  '\n and the corresponding eigen value is', w[0])
print('The second eigen vector is ', v[:,1], \
  '\n and the corresponding eigen value is', w[1])
```

행렬의 고유벡와 고윳값을 *Python*으로 구하기 *(2)*

이 스크립트를 실행하면 다음 결과를 얻는다.

```
eigen values are  [ 7. -1.]
The first eigen vector is  [ 0.70710678  0.70710678]
and the corresponding eigen value is 7.0
The second eigen vector is  [-0.70710678  0.70710678]
and the corresponding eigen value is -1.0
```

- `w`는 고윳값들로 이루어진 행렬이고, `v`는 고유벡터들로 이루어진 정방행렬
- `w`의 7과 -1이 각각 고윳값
- 7의 고유벡터는 `v`의 첫 번째 열벡터인 $\begin{bmatrix}0.70710678 & 0.70710678\end{bmatrix}^\intercal = \begin{bmatrix}\sqrt{2}/2 & \sqrt{2}/2\end{bmatrix}^\intercal$
- -1의 고유벡터는 두 번째 열벡터인 $\begin{bmatrix}-0.70710678 & 0.70710678\end{bmatrix}^\intercal = \begin{bmatrix}-\sqrt{2}/2 & \sqrt{2}/2\end{bmatrix}^\intercal$
- `eig()` 함수가 반환하는 고유벡터의 크기는 1

행렬의 고유벡와 고윳값을 *Python*으로 구하기 *(3)*

```python
import sympy as sym
A = sym.Matrix([[3,4],[4,3]])
A.eigenvects()
```

실행 결과

```
[(-1, 1, [Matrix([
[-1],
[ 1]])]), (7, 1, [Matrix([
[1],
[1]])])]
```

행렬의 고유벡터와 고윳값을 *Python*으로 확인하기

```
>>> np.dot(A,v[:,0])
array([ 4.94974747,  4.94974747])
>>> np.dot(A,v[:,1])
array([ 0.70710678, -0.70710678])
>>> np.dot(A, v[:,0]) - w[0]*v[:,0]
array([ 0.,  0.])
>>> np.dot(A, v[:,1]) - w[0]*v[:,1]
array([ 5.65685425, -5.65685425])
>>> np.dot(A, v[:,1]) - w[1]*v[:,1]
array([ 0.,  0.])
>>> np.dot(A,v)
array([[ 4.94974747,  0.70710678],
       [ 4.94974747, -0.70710678]])
>>> np.dot(A,v)-w*v
array([[ 0.,  0.],
       [ 0.,  0.]])
```

Python 계산 과정의 절삭 오류 *(1)*

```
>>> B = np.array([[1,3,7], [-1,3,-2], [0,5,3]])
>>> w, v = LA.eig(B)
>>> w
array([-0.23564993+0.j        ,  3.61782496+4.07632468j,
        3.61782496-4.07632468j])
>>> v
array([[ 0.96018425+0.j, 0.77761641+0.j, 0.77761641+0.j],
       [ 0.15177853+0.j        , -0.20190383+0.34529972j,
        -0.20190383-0.34529972j],
       [-0.23454103+0.j        ,  0.37733931+0.3048454j ,
         0.37733931-0.3048454j ]])
```

Python 계산 과정의 절삭 오류 (2)

```
>>> np.dot(B,v)-w*v
array([[ 6.93889390e-16 +0.00000000e+00j,
        -2.22044605e-15 +2.22044605e-15j,
        -2.22044605e-15 -2.22044605e-15j],
       [ -3.95516953e-16 +0.00000000e+00j,
         1.77635684e-15 +7.77156117e-16j,
         1.77635684e-15 -7.77156117e-16j],
       [ -2.77555756e-17 +0.00000000e+00j,
        -2.22044605e-16 +8.88178420e-16j,
        -2.22044605e-16 -8.88178420e-16j]])
```

고유벡터와 고윳값 중 복소수가 포함되는 예 *(1)*

```
>>> x1 = v[:,0]; x2 = v[:,1]; x3 = v[:,2]
>>> x1
array([ 0.96018425+0.j,  0.15177853+0.j, -0.23454103+0.j])
>>> x2
array([ 0.77761641+0.j        , -0.20190383+0.34529972j,
        0.37733931+0.3048454j ])
>>> x3
array([ 0.77761641+0.j        , -0.20190383-0.34529972j,
        0.37733931-0.3048454j ])
>>> w
array([-0.23564993+0.j        ,  3.61782496+4.07632468j,
        3.61782496-4.07632468j])
```

각 고유벡터의 고윳값은 각각 $-0.2356, 3.6178 + 4.0763j, 3.6178 - 4.0763j$이다.

고유벡터와 고윳값 중 복소수가 포함되는 예 *(2)*

```
>>> np.dot(B,x1)-w[0]*x1
array([ 6.93889390e-16+0.j, -3.95516953e-16+0.j,
       -2.77555756e-17+0.j])
```

첫 번째 고유벡터 `x1`(이를 $\mathbf{x}_1$ 이라 하자)의 고윳값은 `w[0]=-0.2356`인데 이를 λ_1 이라고 하자.

$$\begin{aligned}\mathbf{Bx}_1 - \lambda_1\mathbf{x}_1 &= \begin{bmatrix} 1 & 3 & 7 \\ -1 & 3 & -2 \\ 0 & 5 & 3 \end{bmatrix}\begin{bmatrix} 0.96018425 \\ 0.15177853 \\ -0.23454103 \end{bmatrix} + 0.23564993\begin{bmatrix} 0.96018425 \\ 0.15177853 \\ -0.23454103 \end{bmatrix} \\ &= \begin{bmatrix} 6.9389 \cdot 10^{-16} \\ -3.9552 \cdot 10^{-16} \\ -2.7756 \cdot 10^{-17} \end{bmatrix} \simeq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \tag{8.14}$$

고유벡터와 고윳값 중 복소수가 포함되는 예 *(3)*

같은 방법으로 두 번째와 세 번째 고유벡터와 고윳값이 특성방정식을 만족하는지 확인해 보자.

```
>>> np.dot(B,x2)-w[1]*x2
array([ -2.22044605e-15 +2.22044605e-15j,
         1.77635684e-15 +7.77156117e-16j,
        -2.22044605e-16 +8.88178420e-16j])
>>> np.dot(B,x3)-w[2]*x3
array([ -2.22044605e-15 -2.22044605e-15j,
         1.77635684e-15 -7.77156117e-16j,
        -2.22044605e-16 -8.88178420e-16j])
>>>
```

세 결과 모두 $\begin{bmatrix}0 & 0 & 0\end{bmatrix}^\intercal$에 근접한 결과가 나오기는 하지만 정확하게 $\mathbf{0} = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^\intercal$가 나오지 않는 이유는 디지털 컴퓨터의 유한 단어길이 효과 때문에 발생하는 절삭 오류 때문이다.

이산 시간 선형 시불변 시스템의 고유신호와 주파수 응답 개요

- 선형 시불변 시스템은 한 신호 공간을 다른 신호 공간으로 사상한다
- 선형변환의 고유벡터에 해당하는 것을 선형 시불변 시스템에서는 고유신호라고 부른다
- 선형변환의 고윳값에 해당하는 것을 선형 시불변 시스템에서는 주파수 응답 (frequency response) 이라고 부른다

이산시간 선형 시불변 시스템의 고유신호와 고윳값 정의

정의 8.2 (이산시간 선형 시불변 시스템의 고유신호와 고윳값)

이산시간 선형 시불변 시스템 $\mathcal{T}$에 신호 $x[n]$이 입력된다고 가정하자. 만약에 다음 조건을 만족하는 복소수 λ가 존재한다면, 이산시간 신호 $x[n]$은 이산시간 선형 시스템 $\mathcal{T}$의 고유신호(eigensignal) 또는 형태 불변 신호라고 하고, λ는 고윳값이라고 한다.

$$\mathcal{T}\{x[n]\} = \lambda x[n]. \tag{8.15}$$

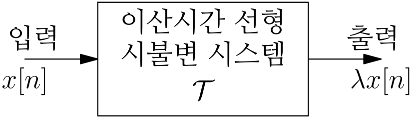


그림 8.3: 이산시간 선형 시불변 시스템 $\mathcal{T}$의 고유 신호 $x[n]$.

FIR 필터의 고유신호와 주파수 응답

임펄스 응답이 $h[n]$ 인 FIR 필터의 입출력 관계식은 다음과 같이 콘볼루션 합으로 표현된다.

$$y[n] = \sum_{k=0}^{M} h[k]x[n-k]. \tag{8.16}$$

입력 신호 $x[n]$ 이 다음과 같은 복소지수 신호일 때를 생각해 보자.

$$x[n] = Ae^{j\phi}e^{j\hat{\omega}n}, \qquad -\infty < n < \infty, \tag{8.17}$$

여기에서 $\hat{\omega}$ 은 정규화된 라디안 주파수이다.

참고 :

$$x[n] = Ae^{j\phi}e^{j\hat{\omega}n} = Ae^{j(\hat{\omega}n+\phi)} = A\cos(\hat{\omega}n+\phi) + jA\sin(\hat{\omega}n+\phi)$$

FIR 필터의 고유신호와 고윳값 *(1)*

식 (8.17)을 식 (8.16)에 대입하여 출력 $y[n]$을 다음과 같이 계산할 수 있다.

$$\begin{aligned} y[n] &= \sum_{k=0}^{M} h[k] A e^{j\phi} e^{j\hat{\omega}(n-k)} = \sum_{k=0}^{M} h[k] A e^{j\phi} e^{j\hat{\omega}n} e^{-j\hat{\omega}k} \\ &= \left(\sum_{k=0}^{M} h[k] e^{-j\hat{\omega}k} \right) A e^{j\phi} e^{j\hat{\omega}n} = \left(\sum_{k=0}^{M} h[k] e^{-j\hat{\omega}k} \right) x[n]. \end{aligned} \tag{8.18}$$

여기에서

$$\lambda = \sum_{k=0}^{M} h[k] e^{-j\hat{\omega}k} \tag{8.19}$$

라고 두면, 식 (8.18)은 다음과 같이 쓸 수 있다.

$$y[n] = \lambda x[n]. \tag{8.20}$$

이 식으로부터 식 (8.17)은 FIR 필터의 고유신호이고, 이에 대응하는 고윳값은 식 (8.19)와 같이 주어지는 것을 알 수 있다.

FIR 필터의 고유신호와 고윳값 (2)

- 고윳값은 FIR 필터의 임펄스 응답 $h[n]$의 이산시간 푸리에 변환으로서, 입력 복소지수 신호 $x[n] = Ae^{j\phi}e^{j\hat{\omega}n}$의 주파수인 $\hat{\omega}$의 함수
- FIR 필터의 고윳값은 FIR 필터의 임펄스 응답과 고유신호인 복소지수 신호의 주파수에 따라서 달라진다
- 이런 이유 때문에 FIR 필터의 고윳값을 주파수 응답이라고 부른다

$$H(e^{j\hat{\omega}}) = \sum_{k=0}^{M} h[k]e^{-j\hat{\omega}k}. \tag{8.21}$$

$$y[n] = H(e^{j\hat{\omega}})x[n]. \tag{8.22}$$

또한, 주파수 응답 $H(e^{j\hat{\omega}})$는 복소값이므로, 크기와 위상으로 구분하여 표현할 수 있다.

$$H(e^{j\hat{\omega}}) = \left|H(e^{j\hat{\omega}})\right|e^{j\angle H(e^{j\hat{\omega}})}, \tag{8.23}$$

여기에서 $\left|H(e^{j\hat{\omega}})\right|$를 FIR 필터의 크기 응답 (magnitude response) 또는 FIR 필터의 이득 (gain)이라 하고, $\angle H(e^{j\hat{\omega}})$를 FIR 필터의 위상 응답이라고 한다.

FIR 필터의 고유신호와 고윳값 *(3)*

FIR 필터에 입력되는 신호가 $x[n] = Ae^{j\phi}e^{j\hat{\omega}n}$ 이라면, 출력 $y[n]$ 은 다음과 같이 표현된다.

$$\begin{aligned} y[n] &= \left|H(e^{j\hat{\omega}})\right| e^{j\angle H(e^{j\hat{\omega}})} \cdot Ae^{j\phi}e^{j\hat{\omega}n} \\ &= \left(\left|H(e^{j\hat{\omega}})\right| A\right) \cdot e^{j(\angle H(e^{j\hat{\omega}})+\phi)}e^{j\hat{\omega}n}. \end{aligned} \tag{8.24}$$

출력 신호를 크기와 위상으로 구분해서 생각해 보면 다음과 같은 사실을 알 수 있다.

- 출력 신호의 크기는 $\left|H(e^{j\hat{\omega}})\right|$, 즉 FIR 필터의 주파수 응답의 크기 $\left|H(e^{j\hat{\omega}})\right|$ 와 입력신호의 크기 A 의 곱이다.
- 출력 신호의 위상은 FIR 필터의 주파수 응답의 위상 $\angle H(e^{j\hat{\omega}})$ 와 입력 복소 지수 신호의 위상 ϕ 의 합이다.

FIR 필터의 고유신호와 고윳값 (4)

예제 8.4

차분 방정식 계수가 $b_k = \{\underset{\uparrow}{1}, 2, 1\}$ 인 FIR 필터의 입력 $x[n]$과 출력 $y[n]$ 사이의 관계식을 적고, 이 필터의 주파수 응답을 구하라.

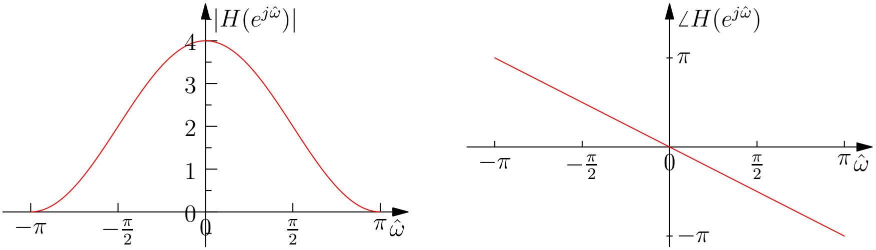
$$y[n] = x[n] + 2x[n-1] + x[n-2]. \tag{8.25}$$

$$\begin{aligned} H(e^{j\hat{\omega}}) &= \sum_{k=0}^{M} h[k]e^{-j\hat{\omega}k} = 1 + 2e^{-j\hat{\omega}} + e^{-j2\hat{\omega}} \\ &= e^{-j\hat{\omega}}\left(e^{j\hat{\omega}} + 2 + e^{-j\hat{\omega}}\right) = e^{-j\hat{\omega}}\left(2 + 2\cos\hat{\omega}\right). \end{aligned} \tag{8.26}$$

$$\begin{aligned} \left|H(e^{j\hat{\omega}})\right| &= \left|e^{-j\hat{\omega}}\left(2 + 2\cos\hat{\omega}\right)\right| = \left|e^{-j\hat{\omega}}\right| \cdot |2 + 2\cos\hat{\omega}| = \left(2 + 2\cos\hat{\omega}\right), \\ \angle H(e^{j\hat{\omega}}) &= -\hat{\omega}. \end{aligned} \tag{8.27}$$

FIR 필터의 고유신호와 고윳값 *(5)*

그림 8.4는 예제 5.4에서 구한 FIR 필터의 크기 응답과 위상 응답의 그래프를 $-\pi < \hat{\omega} \leq \pi$ 구간에서 나타낸 것이다.



(a) 크기 응답 (b) 위상 응답

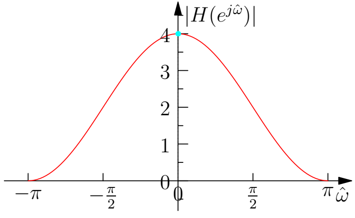
그림 ***8.4:*** 주파수 응답 $H(e^{j\hat{\omega}}) = (2 + 2\cos\hat{\omega})e^{-j\hat{\omega}}$ 의 크기(a)와 위상(b).

FIR 필터의 고유신호와 고윳값 (6)

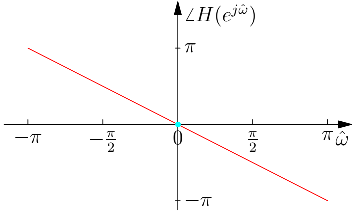
- 이 필터에 $\hat{\omega} = 0$ 인 복소 지수 신호가 입력되는 경우의 출력. 예 : $x[n] = 2e^{j0n}$ 일 때, $y[n]$
  입력 신호의 주파수가 $\hat{\omega} = 0$ 이므로, $y[n]$ 은 다음과 같다.

$$y[n] = H(e^{j\hat{\omega}})\big|_{\hat{\omega}=0} 2e^{j0n} = (2 + 2\cos 0)e^{-j0} 2e^{j0n}$$

$$= (2+2)2e^{j0n} = 8e^{j0n} = 8. \tag{8.28}$$



(a) 크기 응답

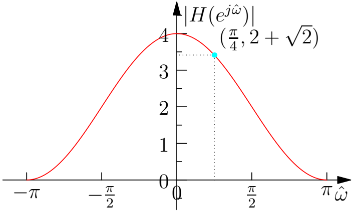


(b) 위상 응답

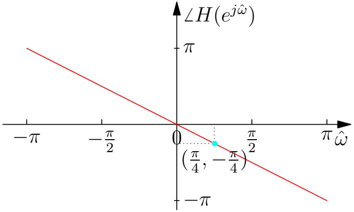
FIR 필터의 고유신호와 고윳값 *(7)*

- 이 필터에 $\hat{\omega} = \frac{\pi}{4}$ 인 복소 지수 신호가 입력되는 경우의 출력. 예 : $x[n] = 2e^{j\frac{\pi}{4}n}$ 일 때, $y[n]$
  입력 신호의 주파수가 $\hat{\omega} = \frac{\pi}{4}$ 이므로, $y[n]$ 은 다음과 같다.

$$
\begin{aligned}
y[n] &= H(e^{j\hat{\omega}})\Big|_{\hat{\omega}=\frac{\pi}{4}} 2e^{j\frac{\pi}{4}n} \\
&= \left(2 + 2\cos\frac{\pi}{4}\right) e^{-j\frac{\pi}{4}} 2e^{j\frac{\pi}{4}n} = \left(4 + 2\sqrt{2}\right) e^{j\frac{\pi}{4}(n-1)}.
\end{aligned} \tag{8.29}
$$



(a) 크기 응답

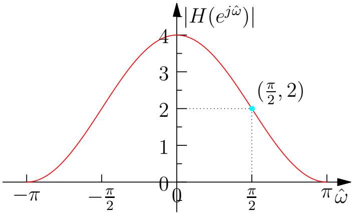


(b) 위상 응답

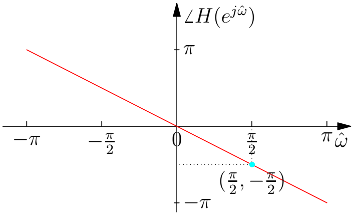
FIR 필터의 고유신호와 고윳값 (8)

- 이 필터에 $\hat{\omega} = \frac{\pi}{2}$ 인 복소 지수 신호가 입력되는 경우에 출력을 살펴 보자. 예를 들어, $x[n] = 2e^{j\frac{\pi}{2}n}$ 일 때, $y[n]$ 을 구해 보자.
  입력 신호의 주파수가 $\hat{\omega} = \frac{\pi}{2}$ 이므로, $y[n]$ 은 다음과 같다.

$$\begin{aligned} y[n] &= H(e^{j\hat{\omega}})\Big|_{\hat{\omega}=\frac{\pi}{2}} 2e^{j\frac{\pi}{2}n} = (2 + 2\cos\frac{\pi}{2})e^{-j\frac{\pi}{2}} 2e^{j\frac{\pi}{2}n} \\ &= 2 \cdot 2e^{j(\frac{\pi}{2}n - \frac{\pi}{2})} = 4e^{j\frac{\pi}{2}(n-1)}. \end{aligned} \tag{8.30}$$



(a) 크기 응답

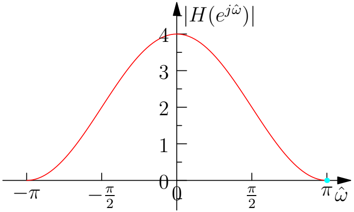


(b) 위상 응답

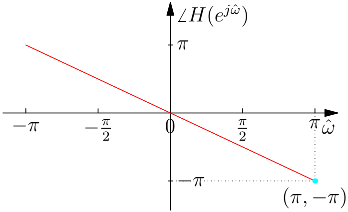
FIR 필터의 고유신호와 고윳값 (9)

- 이 필터에 $\hat{\omega} = \pi$ 인 신호가 입력되는 경우에 출력을 살펴 보자. 예를 들어, $x[n] = 2e^{j\pi n}$ 일 때, $y[n]$ 을 구해 보자. 입력 신호의 주파수가 $\hat{\omega} = \pi$ 이므로, $y[n]$ 은 다음과 같다.

$$\begin{aligned} y[n] &= H(e^{j\hat{\omega}})\Big|_{\hat{\omega}=\pi} 2e^{j\pi n} = (2 + 2\cos\pi)e^{-j\pi} 2e^{j\pi n} \\ &= (2-2)2e^{j\pi(n-1)} = 0. \end{aligned} \tag{8.31}$$



(a) 크기 응답



(b) 위상 응답

FIR 필터의 고유신호와 고윳값 *(10)*

이상의 결과로부터 다음과 같은 사실을 알 수 있다.

- 이 FIR 필터에 복소지수 신호가 입력되면 주파수가 동일한 복소지수가 출력되지만 크기는 입력 복소지수의 주파수의 크기에 따라서 달라진다
- 주파수가 증가할수록 출력되는 복소지수 신호의 크기가 줄어든다
- 이 필터는 주파수의 크기가 증가할수록 입력대 출력의 비율이 줄어들어서 주파수가 최댓값인 $\hat{\omega} = \pi$가 되면 출력의 크기가 0이 된다

이런 필터를 저역 통과 필터 (low–pass filter)라고 한다

FIR 필터의 고유신호와 고윳값 (9)

예제 8.5

차분방정식 계수 $\{b_k\} = \{\underset{\uparrow}{1}, 1, 1\}$ 인 FIR 필터가 있다.

- (a) 주파수 크기 응답과 위상 응답을 구하라.
- (b) 이 FIR 필터에 복소 지수 신호 $x[n] = 2e^{j\frac{\pi}{4}}e^{j\frac{\pi}{3}n}$ 이 입력될 때 출력 $y[n]$ 을 구하라.

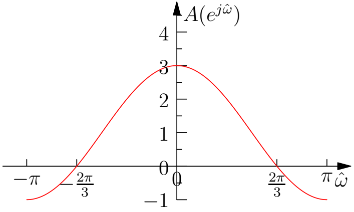
(a) 이 필터의 주파수 응답은 다음과 같다.

$$H(e^{j\hat{\omega}}) = \sum_{k=0}^{M} h[k]e^{-j\hat{\omega}k} = 1 + e^{-j\hat{\omega}} + e^{-j2\hat{\omega}}$$

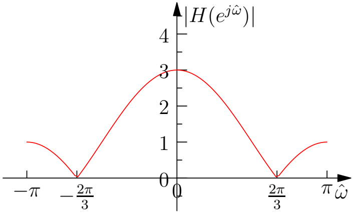
$$= e^{-j\hat{\omega}}\left(e^{j\hat{\omega}} + 1 + e^{-j\hat{\omega}}\right) = e^{-j\hat{\omega}}\left(1 + 2\cos\hat{\omega}\right). \tag{8.32}$$

여기에서 $-1 \le 1 + 2\cos\hat{\omega} \le 3$이기 때문에 크기 응답과 위상 응답을 구할 때 다음과 같이 구간을 구분해서 생각해야 한다.

FIR 필터의 고유신호와 고윳값 *(10)*



(a) 진폭 응답



(b) 크기 응답

FIR 필터의 고유신호와 고윳값 (11)

- $-1 \leq 1 + 2\cos\hat{\omega} < 0$일 때, 즉 $-\pi < \hat{\omega} < -\frac{2\pi}{3}$ 또는 $\frac{2\pi}{3} < \hat{\omega} \leq \pi$일 때, 주파수 응답을 다음과 같이 쓸 수 있다.

$$\begin{aligned} H(e^{j\hat{\omega}}) &= -\left(1 + 2\cos\hat{\omega}\right)(-1)e^{-j\hat{\omega}} \\ &= -\left(1 + 2\cos\hat{\omega}\right)e^{\pm j\pi}e^{-j\hat{\omega}} \\ &= -\left(1 + 2\cos\hat{\omega}\right)e^{j(\pm\pi - \hat{\omega})}. \end{aligned} \tag{8.33}$$

따라서 주파수 크기 응답과 위상 응답은 각각 다음과 같다.

$$\left|H(e^{j\hat{\omega}})\right| = -1 - 2\cos\hat{\omega}, \tag{8.34}$$

$$\angle H(e^{j\hat{\omega}}) = \pm\pi - \hat{\omega}. \tag{8.35}$$

FIR 필터의 고유신호와 고윳값 (12)

- (ii) $0 \le 1 + 2\cos\hat{\omega} \le 3$일 때, 즉 $-\frac{2\pi}{3} \le \hat{\omega} \le \frac{2\pi}{3}$일 때, 주파수 응답을 다음과 같이 쓸 수 있다.

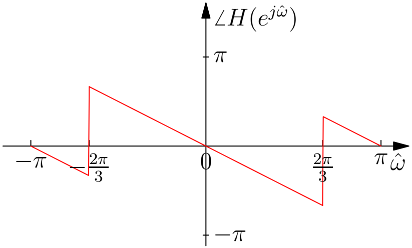
$$H(e^{j\hat{\omega}}) = (1 + 2\cos\hat{\omega})\, e^{-j\hat{\omega}}. \tag{8.36}$$

따라서 주파수 크기 응답과 위상 응답은 각각 다음과 같다.

$$\left|H(e^{j\hat{\omega}})\right| = 1 + 2\cos\hat{\omega}, \tag{8.37}$$

$$\angle H(e^{j\hat{\omega}}) = -\hat{\omega}. \tag{8.38}$$

FIR 필터의 고유신호와 고윳값 *(13)*



위상 응답

FIR 필터의 고유신호와 고윳값 (14)

(b) $H(e^{j\frac{\pi}{3}}) = e^{-j\frac{\pi}{3}}(1 + 2\cos(\frac{\pi}{3})) = 2e^{-j\frac{\pi}{3}}$ 이므로,

$$y[n] = H(e^{j\hat{\omega}})\Big|_{\hat{\omega}=\frac{\pi}{3}} x[n] = 2e^{-j\frac{\pi}{3}} x[n] = 2e^{-j\frac{\pi}{3}} 2e^{j\frac{\pi}{4}} e^{j\frac{\pi}{3}n}$$

$$= (2 \cdot 2)e^{j\frac{\pi}{4}} e^{j(\frac{\pi}{3}n - \frac{\pi}{3})} = 4e^{j\frac{\pi}{4}} e^{j\frac{\pi}{3}(n-1)}. \tag{8.39}$$

Scipy.signal 모듈의 *freqz* 함수를 이용한 디지털 필터의 주파수 응답 계산

IIR 필터의 주파수 응답은 다음과 같이 분수식으로 표현된다.

$$H(e^{j\hat{\omega}}) = \frac{B(e^{j\hat{\omega}})}{A(e^{j\hat{\omega}})} = \frac{b_0 + b_1 e^{-j\hat{\omega}} + \cdots + b_N e^{-jN\hat{\omega}}}{a_0 + a_1 e^{-j\hat{\omega}} + \cdots + a_M e^{-jM\hat{\omega}}}. \tag{8.40}$$

```
scipy.signal.freqz(b, a=1, worN=None, whole=0, plot=None)
```

Parmeters

- `b`: 분자 다항식의 계수
- `a`: 분모 다항식의 계수
- `worN`: `None`이면 단위원 위를 512 개의 점에서 주파수 응답을 계산하고, 정수를 지정하면 그 수만큼의 점에서 계산한다.
- `whole`: `True`면 $[0, 2\pi]$ 범위에서 지정하지 않으면 $[0, \pi]$ 범위에서 주파수 응답을 계산한다.
- `plot`: callable

Returns

- `w`: 주파수 응답이 계산되는 주파수의 값들
- `h`: 주파수 응답

$b_k = \{1, 2, 1\}$인 *FIR* 필터의 주파수 응답을 구하는 *Python* 스크립트의 예

```python
from scipy import signal
import numpy as np
import matplotlib.pyplot as plt
b = np.array([1, 2, 1])
w, h = signal.freqz(b)
plt.subplot(121)
plt.plot(w, np.abs(h))
plt.title('Magnitude response')
plt.xlabel('Normalized freq.')
plt.xlim(0, np.pi)
plt.xticks([0, np.pi/2, np.pi],
labels=["$0$", "$\pi/2$", "$\pi$"])
t = "$|H(e^{j\hat{\omega}})|$"
plt.ylabel(t)
plt.subplot(122)
```

```python
plt.plot(w, np.angle(h))
plt.title('Phase response')
plt.xlabel('Normalized freq.')
plt.xlim(0, np.pi)
plt.xticks([0, np.pi/2, np.pi],
labels=["$0$", "$\pi/2$", "$\pi$"])
plt.yticks([-np.pi, -np.pi/2, 0,
            np.pi/2, np.pi],
labels=["$-\pi$", "$\pi/2$",
   "$0$", "$\pi/2$", "$\pi$"])
t = r"$\angle H(e^{j\hat{\omega}})$"
plt.ylabel(t)
plt.xlim(0, np.pi)
plt.ylim(-np.pi, np.pi)
plt.tight_layout()
plt.show()
```

$b_k = \{1, 2, 1\}$ 인 *FIR* 필터의 주파수 응답을 구하는 *Python* 스크립트의 예

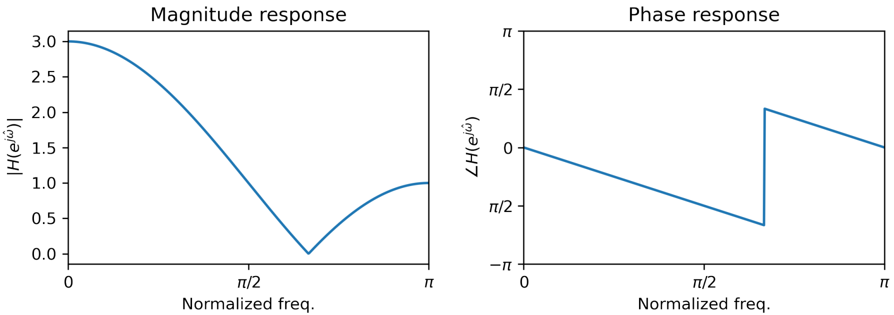


그림 *8.5:* $b_k = \{1, 2, 1\}$ 인 FIR 필터의 주파수 응답.

이 필터의 모든 계수가 실수이기 때문에 주파수 응답이 켤레 대칭 성질을 만족하므로 $[0, \pi]$ 범위에서의 그래프만 보아도 $[-\pi, \pi]$ 범위의 주파수 응답을 유추할 수 있기 때문이다.