

신호및시스템

4장. 신호의 기초 – Tone Music

한림대학교 소프트웨어융합대학

담당교수: 박섭형

2021년 1학기

```
[5]: import numpy as np
import matplotlib.pyplot as plt
```

예제 1. 톤 신호 다음 신호를 생각해 보자.

$$x(t) = A \sin(2\pi ft + \phi)$$

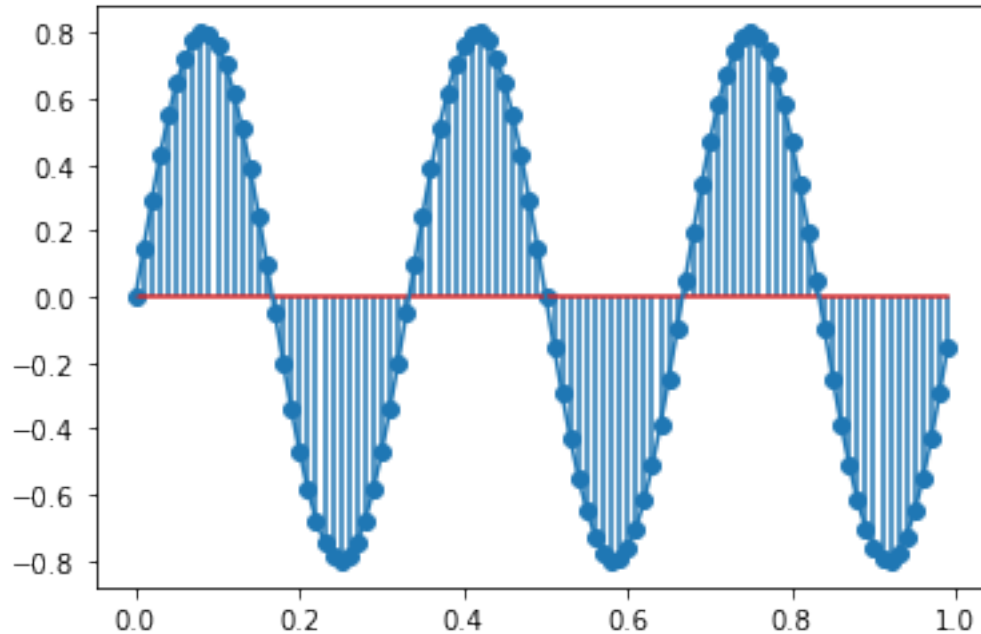
다음 함수는 주파수가 f 인 정현파를 duration (초) 동안 샘플링 주파수 f_s 로 샘플링한 이산시간 신호를 `numpy.ndarray`로 반환하는 함수이다.

```
[4]: def single_tone(A, f, fs, phi, duration):
    t = np.arange(0, duration, 1/fs)
    return t, A * np.sin(2 * np.pi * f * t + phi)
```

예제 1.

주파수가 3 Hz이고 최대 진폭이 0.8인 1초 길이의 정현파 신호를 만들고 이 신호의 그래프를 그려라. 단, 샘플링 주파수는 100 Hz로 한다.

```
[8]: A = 0.8
f0 = 3.
fs = 100.
t, x = single_tone(A, f0, fs, 0., 1)
plt.stem(t, x)
plt.plot(t, x)
plt.show()
```



```
[9]: t.shape, x.shape
```

```
[9]: ((100,), (100,))
```

예제 2.

주파수가 440 Hz이고 최대 진폭이 0.8인 정현파를 1초 길이만큼 샘플링한 신호 wav_440을 구하라. 단, 샘플링 주파수는 44100 Hz로 한다.

```
[10]: fs = 44100
      f_440 = 440
      t, wav_440 = single_tone(0.8, f_440, fs, 0, 1. )
```

```
[11]: t.shape, wav_440.shape
```

```
[11]: ((44100,), (44100,))
```

예제 3. 아래 셀의 프로그램은 이 신호를 소리로 출력하는 프로그램이다. 아래 셀의 프로그램을 실행하면서 소리를 들어보라.

```
[12]: from IPython.display import Audio
```

```
Audio(wav_440, rate=fs)
```

```
[12]: <IPython.lib.display.Audio object>
```

```
[13]: Audio(wav_440, rate=fs, autoplay=True)
```

```
[13]: <IPython.lib.display.Audio object>
```

220 Hz와 240 Hz 의 높이 차이와 440Hz와 460 Hz 사이의 높이 차이를 비교하기

예제 4. 주파수가 220 Hz와 240 Hz인 정현파의 높이 차이와 주파수가 440Hz와 460 Hz인 정현파 사이의 높이 차이 비교

```
[14]: fs = 44100
      t, wav_220 = single_tone(0.8, 220, fs, 0, 1. )
      t, wav_240 = single_tone(0.8, 240, fs, 0, 1. )
      t, wav_440 = single_tone(0.8, 440, fs, 0, 1. )
      t, wav_460 = single_tone(0.8, 460, fs, 0, 1. )
```

```
[15]: wav_four = np.append(wav_220, wav_240)
      wav_four = np.append(wav_four, wav_440)
      wav_four = np.append(wav_four, wav_460)
```

```
[16]: Audio(wav_four, rate=fs, autoplay=True)
```

```
[16]: <IPython.lib.display.Audio object>
```

예제 4. 주파수가 220 Hz와 242 Hz인 정현파의 높이 차이와 주파수가 440Hz와 484 Hz인 정현파 사이의 높이 차이 비교

```
[17]: t, wav_242 = single_tone(0.8, 242, fs, 0, 1. )
      t, wav_484 = single_tone(0.8, 484, fs, 0, 1. )
      wav_four_2 = np.append(wav_220, wav_242)
      wav_four_2 = np.append(wav_four_2, wav_440)
      wav_four_2 = np.append(wav_four_2, wav_484)
```

```
[18]: Audio(wav_four_2, rate=fs, autoplay=True)
```

```
[18]: <IPython.lib.display.Audio object>
```

예제 3과 예제 4로부터 알 수 있는 사실

- $0 < f_a < f_b$, $0 < f_c < f_d$ 를 만족하는 네 주파수 f_a, f_b, f_c, f_d 라 있다고 가정하자.
 - 주파수가 높은 정현파의 소리가 주파수가 낮은 정현파의 소리보다 높다고 느낀다.
 - 사람은 두 주파수를 갖는 정현파가 만드는 소리의 높이 차이를 두 주파수 사이의 차이가 아닌 두 주파수의 비율로 구분한다.
- $\frac{f_b}{f_a} = \frac{f_d}{f_c}$ 일 때, f_a 주파수 정현파와 f_b 주파수 정현파 두 음 사이의 거리와 f_c 주파수 정현파와 f_d 주파수 정현파 두 음 사이의 거리가 같다고 느낀다.

등비 수열 (geometric sequence) 다시 생각해 보기

- 0 아닌 수들의 열, 즉 수열의 각 항이 바로 앞 항에 0이 아닌 공비(common ratio)가 곱해지는 수열
- 예: 다음 수열은 초항이 2이고 공비가 3인 등비 수열이다.

$$2, 6, 18, 54, 162, 486, \dots, 2 \times 3^{n-1}, \dots$$

- 이 수열의 n 번째 항을 a_n 이라고 하면 다음 식이 성립한다.

$$a_n = 2 \times 3^{n-1}, n \text{은 } n \geq 1 \text{인 정수.}$$

- 또는 다음 식이 성립한다.

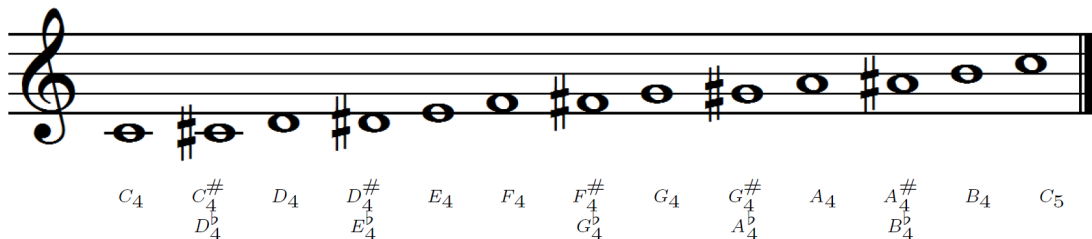
$$a_n = 3a_{n-1}, \text{ 즉 } a_{n-1} = 3^{-1}a_n, n \text{은 } n \geq 2 \text{인 정수.}$$

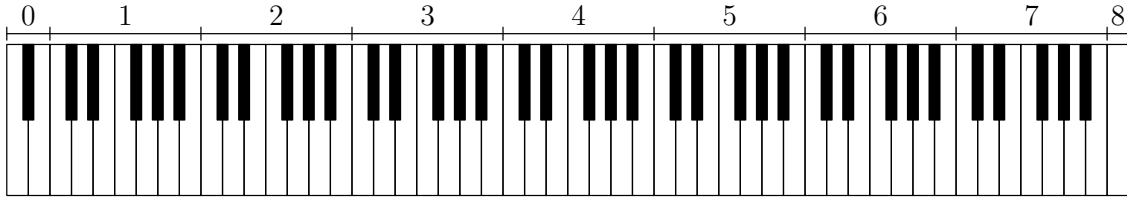
평균율(equal temperament)에서 각 음의 주파수 물리학에서는 주파수가 2 배 차이 나는 소리의 차이를 한 옥타브라고 한다.

음악에서는 어떤 음과 같은 음을 가진 다음 음 사이의 거리를 한 옥타브라고 말한다.

예를 들면, 도와 그 다음으로 높은 도 사이를 한 옥타브라고 한다.

서양 음악에서 사용하는 평균율은 한 옥타브 사이를 12 개의 음이 이웃한 음 사이의 거리가 같도록 음을 배치한다. 그 음의 이름은 다음과 같다.





번호가 n 번인 피아노 건반의 주파수를 f_n 이라고 하자.

예를 들면, A4에 해당하는 피아노 건반의 번호는 49번이고, A4 음의 주파수는 440 Hz이다.

$$f_{49} = 440.000$$

인간이 두 음 사이의 거리(높이의 차이)를 인식할 때 두 음의 주파수의 차이가 아니라 두 음의 주파수의 비율로 구분한다.

예를 들어서, C4와 D4 사이의 음 차이와 D4와 E4 사이의 음 차이가 같다고 인식하려면 다음 관계식이 성립해야 한다.

$$\frac{f_{42}}{f_{40}} = \frac{f_{44}}{f_{42}}$$

C4부터 B4까지의 12 음에서 이웃 음 사이의 거리가 같다면, 각 주파수 사이에는 다음 식이 성립해야 한다.

$$\frac{f_{41}}{f_{40}} = \frac{f_{42}}{f_{41}} = \frac{f_{43}}{f_{42}} = \dots = \frac{f_{50}}{f_{49}} = \frac{f_{51}}{f_{50}} = r.$$

그러면 다음 수열은 공비가 r 인 등비수열이다.

$$f_{40}, f_{42}, \dots, f_{51}$$

그리고 같은 이유로 다음 수열도 공비가 r 인 등비수열이다.

$$f_1, f_2, \dots, f_{88}$$

그러므로, 다음 식이 성립한다.

$$f_n = f_1 r^{n-1}, \quad 2 \leq n \leq 88$$

,

그리고 f_{40} 과 f_{52} 는 각각 C4와 C5의 주파수이므로 한 옥타브 차이가 난다. 즉, 다음 식이 성립한다.

$$\frac{f_{52}}{f_{40}} = 2$$

그리고, $f_{52} = f_{40}r^{12}$ 가 성립한다.

따라서 $r^{12}f_{40} = 2f_{40}$ 이므로, $r^{12} = 2$ 이다. 즉, $r = 2^{\frac{1}{12}}$ 이다.

따라서,

$$f_n = 2^{\frac{1}{12}} f_{n-1}, \quad 2 \leq n \leq 88.$$

$f_{49} = 440.000$ Hz라고 두면, 다음 식이 성립한다.

$$f_n = 440 \times \left(2^{\frac{1}{12}}\right)^{n-49} = 440 \times 2^{(n-49)/12}, \quad 1 \leq n \leq 88$$

| 건반 번호 | 이름 | 주파수 |
|-------|----------|------------------------|
| 52 | C5 | $440 \times 2^{3/12}$ |
| 51 | B4 | $440 \times 2^{2/12}$ |
| 50 | A#4, Bb4 | $440 \times 2^{1/12}$ |
| 49 | A4 | 440.000 |
| 48 | G#4, Ab4 | $440 \times 2^{-1/12}$ |
| 47 | G4 | $440 \times 2^{-2/12}$ |
| 46 | F#4, Gb4 | $440 \times 2^{-3/12}$ |
| 45 | F4 | $440 \times 2^{-4/12}$ |
| 44 | E4 | $440 \times 2^{-5/12}$ |
| 43 | D#4, Eb4 | $440 \times 2^{-6/12}$ |
| 42 | D4 | $440 \times 2^{-7/12}$ |
| 41 | C#4, Db4 | $440 \times 2^{-8/12}$ |
| 40 | C4 | $440 \times 2^{-9/12}$ |

예제 5. 정현파로 악보 연주하기.

```
[19]: A4_no = 49

C_scale_wave = np.array([])
for n in [40, 42, 44, 45, 47, 49, 51, 52]:
    f = 440.0 * (2 ** ((n-A4_no)/12))
    _, wav = single_tone(0.8, f, fs, 0., 0.5)
    C_scale_wave = np.append(C_scale_wave, wav)

[20]: C_scale_wave.shape

[20]: (176400,)

[21]: Audio(C_scale_wave, rate=fs, autoplay=True)
```

[21]: <IPython.lib.display.Audio object>

```
[22]: school_bell_sop = [47, 47, 49, 49, 47, 47, 44, 44]
```

```
[23]: school_bell_sop_wave = np.array([])
      for n in school_bell_sop:
          f = 440.0 * (2 ** ((n-69)/12))
          _, tone = single_tone(0.8, f, fs, 0, 0.5)
          school_bell_sop_wave = np.append(school_bell_sop_wave, tone)
```

```
[24]: school_bell_sop_wave.shape
```

[24]: (176400,)

```
[25]: Audio(school_bell_sop_wave, rate=fs, autoplay=True)
```

[25]: <IPython.lib.display.Audio object>

음이름과 피아노 건반 번호 연결하기

```
[26]: note_no = {'C4': 40, 'D4': 42, 'E4': 44, 'F4': 45,
                'G4': 47, 'A4': 49, 'B4': 51, 'C5': 52}
```

```
[27]: note_no['C4']
```

[27]: 40

```
[28]: note_no['G4'], note_no['G4'], note_no['A4'], note_no['A4'], \
      note_no['G4'], note_no['G4'], note_no['E4'], note_no['E4']
```

[28]: (47, 47, 49, 49, 47, 47, 44, 44)

```
[30]: school_bell_sop = [47, 47, 49, 49, 47, 47, 44, 44]
      school_bell_sop_note = ['G4', 'G4', 'A4', 'A4', 'G4', 'G4', 'E4', 'E4']
      [note_no[note] for note in school_bell_sop_note]
```

[30]: [47, 47, 49, 49, 47, 47, 44, 44]

```
[31]: school_bell_sop_note = ['G4', 'G4', 'A4', 'A4', 'G4', 'G4', 'E4', 'E4']
      school_bell_sop = [note_no[note] for note in school_bell_sop_note]

      school_bell_sop_wave = np.array([])
      for n in school_bell_sop:
```

```
f = 440.0 * (2 ** ((n-69)/12))  
t, tone = single_tone(0.8, f, fs, 0, 0.5)  
school_bell_sop_wave = np.append(school_bell_sop_wave, tone)
```

```
[32]: Audio(school_bell_sop_wave, rate=fs, autoplay=True)
```

```
[32]: <IPython.lib.display.Audio object>
```

생각해 보기

- 심표를 연주하려면 어떻게 해야할까?
- 각 음의 끝 부분을 살짝 끊으려면 어떻게 해야할까?
- 음 사이의 잡음을 없애려면 어떻게 해야할까?

```
[ ]:
```