

1주차

1. Phonology와 Phonetics의 구분

- Phonology(음운론) : 인지적인 단계, 발화되지 않은 소리, 비연속적
- Phonetics(음성학) : 현실적인 단계, 발화된 소리, 연속적

2. English Phonetics 내에서의 구분

- 2.1. Articulatory Phonetics : from mouth
- 2.2. Acoustics Phonetics : through air
- 2.3. Auditory Phonetics : to ear

3. Articulation

3.1. Upper Vocal Tract

종류 : lip, teeth, alveolar, hard palate, soft palate, uvula, pharynx wall, ridge
주의할 점 : 영어는 alveolar에서 조음되는 음들이 발달함.

3.2. Lower Vocal Tract

종류 : lip, blade. top, front, center, back, root, epiglottis

3.3. 5 speech organs (=constrictors)

velum, larynx, lips, tongue tip, tongue body

4. Oro-nasal Process

Velum lowered : nasal tract로 가는 길이 열림. 공기가 nasal tract로 통함. nasal sound (m,n,ŋ)

Velum raised : nasal tract로 가는 길이 막힘. nasal tract로 공기가 흐르지 않음.

5. Phonation Process

Voiced와 Voiceless의 생성과정

Larynx는 Voicebox와 같다. 성대의 떨림의 여부에 따라 Voiced와 Voiceless를 구분

Voiced : Larynx closed -> vibration

Voiceless : Larynx open -> no vibration

6. Articulatory Process

6.1. Constriction location (CL) : 혀의 전후 위치

- lips : Bilabial, Labiodental
- tongue body : dental, alveolar, Palato-Alveolar, Retroflex
- tongue tip : Palatal, Velar

6.2. Constriction degree (CD) : 혀의 고저 위치

- CD의 강도에 따라 4가지로 분류함.
- stops, fricatives, approximants, vowels. (폐쇄음, 마찰음, 접근음, 모음)
- approximants : l, r, j, w
- 비음들은 nasal tract가 열리는 것이기 때문에 영어에선 stops에 해당된다.

7. 정리

- 다음의 다섯가지 분류 기준에 따라 자음과 모음을 구분하면 된다.
- specify constrictors, CD, CL, nasal tract의 개폐 여부, 성대의 진동 여부

8. Acoustics

영어음성학 Summary

2014130031 국어국문학과 민종원

- Praat의 활용. 소리의 파형과 스펙트로그램을 통해 나타난 Duration, Pitch, Intensity를 주목하기
- F1, F2는 모음을 구별하는 기준점이 된다.

2주차

1. Vowel Acoustics

- Hz : the number of occurrences of a repeating event per second
- repeating event : vibration of vocal folds
- Repeating? This might remind you of a sine wave
→ 사인 웨이브에서 소리의 높이와 밀도 볼 수 있음.

2. Complex tone in spectrum

#####이 세상에 존재하는 모든 시그널은 사인 웨이브 형태의 여러 소리의 합으로 표현될 수 있다. #####

- 그래프의 x축은 time, y축은 value.
- magnitude(amplitude) : 소리의 강도.
- frequency : 음 높이와 관련.
- 1,2,3번 그림은 simplex tone, 4번 그림은 complex tone. 4번이 우리가 일반적으로 듣는 소리의 형태.
- 1,2,3에서 4를 만드는 것을 Synthesis, 4를 1,2,3으로 쪼개는 것을 analysis라고 함.
- complex tone의 frequency는 가장 frequency가 낮은 simplex tone의 frequency와 같다.

3. Human voice source

- 성대에서 나오는 소리를 source라고 함. filter를 어떻게 바꾸냐에 따라서 달라짐.
- 스펙트럼에서 f0 에서 frequency의 배수가 커질수록 amplitude가 점차 줄어든다. (gradually decreasing)
- 남자와 여자의 배음을 생각해보자. 여자의 pitch가 더 높기 때문에 임의의 Hz 구간에서 남자의 배음이 더 많을 것!
- 스펙트로그램에서 x축은 여전히 시간, 진한 부분은 amplitude가 큰 것.

4. Source & Filter

- filter : 입모양의 영향으로 source가 바뀜. mountains and valleys가 생긴다
- 사람의 목소리는 harmonics로 이루어져 있다 : sine wave의 배음으로 이루어져 있다
- complex tone : sum of pure tones
- Pure tone : F0, rate of vibration of the larynx
- the amplitude of pure tones gradually decreases
- Guitar plucking : a complex tone consisting of pure tone components. 사람말이랑 비슷함.
- waveform : time-value plot
- spectrum : frequency-amplitude plot
- Source Spectrum은 gradually decreasing

5. Synthesizing Source

- Hz는 100씩 증가, amplitude는 0.05씩 줄이면서 100 - 1000Hz까지!
- Combine to stereo + Convert to mono
- Convert to mono 했을 때 사운드는 100Hz와 비슷하게 들린다 (인지심리학)

6. Vowel Space

F1, F2를 기준으로 하는 그래프

F1 : Output spectrum의 1st peak. 혀의 높낮이를 결정

F2 : Output spectrum의 2nd peak. 혀의 front - back을 결정한다.

- Vowel Space를 봤을 때 한국어의 'ㅏ'와 영어의 'a'는 살짝 다름. 영어가 더 back, low
- 한국어는 구강을 좀더 좁게 이용한다.

3주차

코딩에 앞서서

- 목적 : 자동화
 - 컴퓨터 언어들의 공통점 : 단어와 문법에 영역에서
 - 컴퓨터 언어의 단어 (variable) : 정보를 담는 그릇
 - 컴퓨터 언어 문법의 핵심적인 내용
- 1) 변수에 정보를 할당하는 것 2) if문 3) for문 4) 함수에 대한 이해

Variable

- Variable은 정보를 저장한다.
- Variable에 저장될 수 있는 정보의 형태는 숫자, 문자 2개 뿐.
- 오른쪽에 있는 정보를 왼쪽에 variable에 저장하는 것.
- 함수는 원하는 대로 지정할 수 있음.

예시)

a = 1

print(a)

Jupyter notebook의 사용법

- 각 행을 클릭하고 하늘색 표시가 나올 때 단축키를 누르기
- b : below에 한 줄 추가
- a : above에 한 줄 추가
- x : 선택한 행 없애기
- 초록색 표시에서 shift + enter : run

Python의 함수

- type(*) : *의 type을 확인할 수 있다
- cf) 파이썬의 type : string, 숫자(integer, float), list, tuple, dictionary

4주차

1. Jupyter Notebook을 통해 Python 코딩 시 주의 사항

- Markdown을 체크하면 주석처럼 쓸 수 있다.
- indent 주의하기. 특히 for loop이나 if같은 조건문에서!

2. 조건문

영어음성학 Summary

2014130031 국어국문학과 민종원

(1) for loop

반복 용도

for (#A) in (#B) ####A는 변수 ###B는 범위
B는 range로도 범위를 지정할 수 있다.

예시) a=[1,2,3,4]

```
for i in a:
```

```
    print(i)
```

```
1
```

```
2
```

```
3
```

```
4
```

(2) for loop 중첩

예시)

```
for i in range(1,3):
```

```
    for j in range (3,5):
```

```
        print(i*j)
```

```
3
```

```
4
```

```
6
```

```
8
```

(3) if 함수

조건에 맞으면 실행, 조건에 맞지 않으면 실행안됨. elif와 else로 여러 조건을 중첩시킬 수 있음.

예시)

```
a_again=0
```

```
if a_again == 0:
```

```
    print("yay!")
```

```
else:
```

```
    print("No way!")
```

3. Python 문법들

(1) enumerate : 번호를 매기는 함수. for문의 변수 중 i가 번호.

cf) <https://wikidocs.net/32#enumerate>

(2) zip : 동일한 개수로 이루어진 자료형을 묶어 주는 역할을 하는 함수이다.

(3) format : {} 안에 원하는 문자열을 넣어준다.

e.g. "{}: {}".format('energy', 100)

5주차

- numpy 사용법

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

1. numpy 기본 함수

```
X=np.array ([[0, 0, 0], [0, 0, 0]]) X.shape = (2,3)
```

영어음성학 Summary

2014130031 국어국문학과 민종원

```
np.empty([2,3], dtype='int') ## [2,3]은 2행 3열, 'int'는 정수로 넣으라는 것을 의미한다
np.zeros([2,3]) ## 2행 3열을 0으로 채우기
np.ones([2,3]) ## zeros 함수 대신 ones를 넣으면 1이 들어간 거 ### 뒤에 붙는 '.'을 보아하니
숫자의 디폴트가 float이다.
np.arange(0,10,2, dtype='float64') ## arange (시작, 끝, 간격)
np.arange(5) ## 숫자 하나 쓰면 0부터 시작, 간격 1
np.linspace(0,10,6) ## linspace (시작, 끝, 나누는 개수)
X = np.array([[1,2,3],[4,5,6]]) ## 벡터를 매트릭스로 만드는 식 ## 대괄호가 두 개인거 보고
2차원인 것을 알아야 함. ## 세 개면 3차원
A=np.array([[[1,2],[3,4],[5,6]],[[1,2],[3,4],[5,6]]]) ## 3차원 그래프
A.shape ## 모양에 대한 것.
A.dtype ## A의 타입을 말하는 것
A.astype(np.float64) ## 타입을 float64로 바꾸라는 함수
np.zeros_like(A) ## 다른 건 그대로 두고 A의 형식으로 해라.
Y = X1.reshape(-1,3,2) ## Y 는 모양을 바꾸는 함수. element들의 개수가 안맞으면 에러가 뜬다.
물리적으로 못 바꿈. 이 경우에선 24개가 맞아야 함. 숫자 모르겠으면 앞에 -1을 붙이면 된다
a = np.random.randint(0, 10, [2, 3]) ## randint 함수는 0부터 10까지 숫자에서 2 x 3 으로 임의로
픽해라
b = np.random.random([2, 3]) ## random 함수는 임의로 2 x 3로 만들어내라
np.savez("test", a, b) ## savez 함수는 실제 파일로 저장하는 함수
who ## 지금 사용 가능한 variables
del a, b ## 변수에 할당한 내용 지우는 거
```

2. 정규분포

```
data = np.random.normal(0,1, 100) ## 넘파이 속의 random 속의 normal이라는 함수. normal
distribution을 만드는 함수. (평균, 표준편차, 값의 개수)
plt.hist(data, bins=10) ## matplotlib 사용. bins가 10개라는 소리.
plt.show() ##### 이 값을 모두 합하면? 당연히 100개.
data.ndim ## 변수의 dimension을 알려주는 메서드
```

3. 벡터 계산

- 같은 형식이면 계산 가능하다.

```
a = np.arange(1, 10).reshape(3,3) ## 총 9개인데 3x3으로 재구성
b = np.arange(9, 0, -1).reshape(3,3) ## 총 9개인데 3x3으로 재구성
a==b ; a>b 해보기
a.sum(), np.sum(a) ## 형식의 차이. a.sum()은 그냥 숫자 45만 뱉어낸다.
a.sum(axis=0), np.sum(a, axis=0) ## 첫번째 차원의 관점에서 sum을 해라.
a.sum(axis=1), np.sum(a, axis=1) ## 두번째 차원의 관점
a+100
b = np.arange(6) 후에 a+b ## b가 각 열마다 더해짐.
```

6주차

- 이번 시간부터 파이썬 패키지를 이용해 sinusoid 형태인 음성을 분석할 것.

1. 오일러 공식

(1) 오일러 공식 : $e^{i\theta} = \cos(\theta) + i\sin(\theta)$

영어음성학 Summary

2014130031 국어국문학과 민종원

- (2) 복소수의 정의 : $a + bi$ (a, b 는 실수)
- (3) $f(\Theta) = e^{bi}$ 에서 Θ 에 $0, \pi/2, \pi, 3\pi/2, 2\pi$ 를 대입해보면 $f(\Theta)$ 값은 $1, i, -1, -i$ 가 된다.
- (4) $(\Theta, f(\Theta))$ 를 복소수 평면(complex plain)에 기록하면 $(1,0) (0,1) (-1,0) (0,-1)$
- (5) 복소수 평면에서 Θ 값이 커지면 원의 형태로 돈다.

2. Phasor

- (1) Praat의 스펙트럼은 시간과 value 그래프. 하지만 사인 곡선의 x축은 각도 (Θ) 이다.
- (2) 소리에는 각도 개념뿐만 아니라 시간의 개념이 필요하다. 그래서 시간 설정을 먼저 할 것.
- (3) $\text{amp} = 1$; $\text{sr} = 10000$; $\text{dur} = 0.5$; $\text{freq} = 100.0$
- (4) $t = \text{np.arange}(1, \text{sr} * \text{dur} + 1) / \text{sr}$ ##### Generate Time
1/10000초부터 0.5초까지의 시간을 의미한다.
sampling rate는 시간의 해상도.
duration은 지속 시간
+1 해준 것은 np.arange 함수가 끝의 숫자를 포함하지 않기 때문이다.
벡터의 요소들을 sampling rate로 나눠 시간의 해상도를 결정해준다.
- (5) $\text{theta} = t * 2 * \text{np.pi} * \text{freq}$ ##### Generate Phasor
np.pi만 있으면 시간의 개념이 포함되지 않은 것.
t를 넣어줌으로써 시간의 개념이 포함되었다.
- (6) $s = \text{np.sin}(\text{theta})$ ##### Generate signal by cosine-phasor

3. 2차 평면 plotting

- (1) $\text{fig} = \text{plt.figure}()$ ##### figure 은 그래프 전체
 $\text{ax} = \text{fig.add_subplot}(111)$ ##### 1X1로 figure 나눠 첫번째를 선택
 $\text{ax.plot}(t[0:1000], s[0:1000], '.')$ ##### 여기서 '.'은 .으로 plotting 하라는 것
 $\text{ax.set_xlabel}(\text{"time (s)"})$ 디폴트는 점이 아니라 선이다.
 $\text{ax.set_ylabel}(\text{"real"})$
- (2) $\text{theta} = \text{np.arange}(0, 2 * \text{np.pi}, 0.1)$ ##### 그래프에 점을 뿔뿔하게 해줌.
- (3) 사인함수기에 x축은 균일한 간격(equidistant)을 보이지만 y축은 아니다. (non-linear)
x축도 균일, y축도 균일하려면 linear function이어야 함.
- (4) **예상 시험문제 : ~~ 조건 하에 그래프의 점의 개수는 몇개인가?**
- (5) $\text{ax.plot}(t, s, '.')$ 처럼 범위를 설정해주지 않는다면?
엄청나게 촘촘하게 나온다. 사인 함수같지 않은 형태의 그래프가 나옴.

4. 3차 평면 plotting

- (1) $c = \text{np.exp}(\text{theta} * 1j)$ ##### $a+bi$ 의 형태로 표현한 것.
(exponential)
- (2) $\text{fig} = \text{plt.figure}()$
 $\text{ax} = \text{fig.add_subplot}(111, \text{projection}=\text{"3d"})$
 $\text{ax.plot}(t[0:1000], c.\text{real}[0:1000], c.\text{imag}[0:1000], '.')$
3차원 벡터를 표현한 것. real은 실수 부분만, imag는 허수 부분만 가져옴.
 $\text{ax.set_xlabel}(\text{"time (s)"})$
 $\text{ax.set_ylabel}(\text{"real"})$
 $\text{ax.set_zlabel}(\text{"imag"})$
- (3) 그래프 돌려가면서 실수 부분만 볼 수 있고, 반대로 허수 부분만 볼 수 있음.
- (4) 실수 부분은 cosine 함수의 형태를, 허수 부분은 sine 함수의 형태를 띈다.

영어음성학 Summary

2014130031 국어국문학과 민종원

- (5) `ipd.Audio(s, rate=sr)` ##### 사인 웨이브를 재생하는 함수
- (6) `ipd.Audio`의 `s`는 `c.imag`로 대체할 수 있다.

7주차

- 그동안 `theta`에 `amplitude`를 반영하지 않았음. `amplitude`가 1이었기에 반영하지 않아도 값이 달라지지 않았다.
- `s`의 원래 값에 `amp`값을 곱해 `amp*np.sin(theta)` 표현하여 `theta`에 `amplitude`를 반영한다.
- 3차원 그래프에서 실수와 허수값만을 확인한다면, 원의 형태를 띈다.
- 이때 이 원의 반지름의 값은 `amplitude` 값.

1. Nyquist Frequency

- (1) Sampling Rate의 $\frac{1}{2}$ 값이다.
- (2) 원리
 - Sampling Rate가 10Hz이고 Frequency가 100Hz인 소리를 체크해본다면?
 - 사인함수 특유의 모양으로 인해 최대 5번의 반복만을 체크할 수 있다.
 - Sampling Rate로 체크할 수 있는 지점의 수는 최대 $\frac{1}{2}$ 이다. 이 값이 Nyquist Frequency
- (3) CD음질은 왜 44100Hz인가?
 - Sampling Rate가 44100Hz 라면 Nyquist Frequency는 약 20000Hz
 - 일반적으로 사람의 가청 주파수는 약 20000Hz 정도라고 한다. 즉 사람이 들을 수 있는 최대의 음질 해상도인 것.

2. Generate Samples

- (1) Sampling rate를 `F0`로 한다면 Sampling rate를 정해야 함. Nyquist Frequency는 자동으로 정해짐
- (2) 코드

```
F0 = 100; Fend = int(sr/2); s = np.zeros(len(t)); ## Fend는 Nyquist Frequency의 정수형
for freq in range(F0, Fend+1, F0): ## for 루프로 s 값을 구함
    theta = t * 2*np.pi * freq
    s = s + amp * np.sin(theta)
fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(t[0:1000], s[0:1000]);
ax.set_xlabel('time (msec)') ## 아래 그림은 스펙트럼이 아님. wave form임
```
- (3) 참고사항
 - `s = np.sin(theta)`에서 `sin`을 `cos`으로 바꿔도 출력되는 소리는 달라지지 않는다
 - 사인함수와 코사인함수는 x축에서 $\pi / 2$ 만큼만 차이이기 때문이다.
 - 우리 귀는 각도의 변화를 인지하지 못한다.
 - 일상생활의 옥타브에서 한 음정이 올라가는 것은 주파수가 2배 증가한 것을 의미한다.

3. Generate Pulse Train

- 임의 생성한 함수 `hz2w`와 `resonance`는 논리 구조만 참고.

영어음성학 Summary

2014130031 국어국문학과 민종원

(1) 코드

```
RG = 0 # RG is the frequency of the Glottal Resonator
BWG = 100 # BWG is the bandwidth of the Glottal Resonator
a, b=resonance(sr, RG, BWG)
s = lfilter(b, a, s, axis=0) ## RG를 기준으로 gradually decreasing의 모양을 만들
거임.
ipd.Audio(s, rate=sr) 0을 축으로 산을 만드는데 x축의 음의 영역은 표시되지
```

않음.

- (2) BWG와 RG 값을 다르게 설정하여 코드를 추가 실행함.
- (3) Praat으로 최종 생성된 소리의 스펙트로그램을 확인하면, 산맥이 부드럽게 올라가는 형태임.
- (4) $s = \text{lfilter}(\text{np.array}([1, -1]), \text{np.array}([1]), s)$ 의 의미
 - 사람은 입술이 있기 때문에 소리의 공명이 강화돼 나감. 입술을 만들어주는 과정이라고 생각하면 된다. 실제로 위의 함수로 인해 소리가 더 강하게 나옴.

8주차

- 인공지능 : “데이터 -> 기계 -> 데이터”의 구조.
- 데이터에 어떤 벡터가 들어가냐에 따라 인공지능의 종류 결정

1. 선형대수 기초

- (1) Linear combination : $c * v + d * w$ 의 구조. v, w 는 차원이 같은 벡터. c 와 d 는 scalar
- (2) Vector space : Linear combination의 ‘가능한 모든’ 면을 말한다. 평면에서 일부분만이 vector space라고 부를 수 없음.
- (3) R^n 은 스페이스를 표기하는 방식. n components의 벡터로 구성되어 있다는 뜻이다.
- (4) Column space : column vector에 의한 공간.
ex) 2차원 Column space : 2차원의 두 벡터와 원점은 벡터들이 일직선에 있는게 아니면 삼각형을 이룸. 벡터의 조합이 무한이기 때문에 2차원 평면을 모두 채울 수 있음
- (5) Whole space : vector 자체가 가지고 있는 차원
cf) Whole space == Column space? 답은 No
Whole space의 차원 \geq Column space의 차원
- (6) Independent : 2차원 벡터의 예시에서는 두 벡터가 일직선 상에 있는 것이 아닌 것. 3×3 matrix에서는 한 벡터가 나머지 벡터의 합으로 이루어질 때도 dependent라고 봐야 함.

2. $m \times n$ matrix의 공간

- (1) Transpose : Matrix의 Column과 row를 바꾸는 것을 말함.
- (2) Column의 관점에서 : whole space, column space
- (3) Row의 관점에서 : whole space, row space
- (4) null space의 기하적 정의 : whole space에서 column(row) space가 차지한 나머지 space
- (5) null space의 대수적 정의 : 행렬 A 의 components 모두 0이 되게 만드는 모든 행렬
- (6) Column 관점에서 matrix에 dependent한 벡터가 있다면 whole space와 column space의 차원의 차이가 생겨버림. 그 차원의 차를 null space라고 생각하면 될 듯.
- (7) null space의 방향에 따라 left null space도 있음

3. Linear Transformation

- (1) $Ax = b$ 에서 (A 는 임의의 주어진 행렬, x 는 입력 벡터, b 는 출력벡터)
- (2) 위의 식에서 x 의 차원과 b 의 차원은 반드시 같다고 할 수는 없다

영어음성학 Summary

2014130031 국어국문학과 민종원

- (3) 입력 벡터에 A를 곱하면서 차원이 달라지는 것을 Linear transformation이라고 한다.
- (4) A는 transformation matrix라고 부름
- (5) Original grid와 transformation grid를 활용해 대수적 개념을 기하적 개념으로 해결. 입력 데이터의 grid가 그래프 상에서 이동하는 것으로 생각할 수 있다.
- (6) 역행렬 적용이 가능. $A^{-1} * b = x$ 가 가능하다는 소리
- (7) 단, PPT의 예시처럼 dependent한 벡터는 역행렬 적용이 안된다. 즉, Determinant가 0인 경우에는 Grid를 활용해 해를 찾을 수 없음.
- (8) Eigenvector : 원점과 일직선 상에 있도록 하는 역할

9주차

- 1. Null Space를 왜 배우는가?
 - (1) Null Space의 수학적 해석 : 행렬 A의 components 모두 0이 되게 만드는 모든 행렬
 - (2) Null Space의 기하적 해석 : whole space에서 column(row) space가 차지한 나머지 space
 - (3) Null Space의 응용적 해석 : 출력에 영향을 미치지 않는 공간. 이것이 우리가 Null Space를 배우는 이유
 - (4) 머신러닝 : 강아지 사진을 넣으면 강아지라는 출력이 나오는, 개별 사건에서 보편 사건을 뽑아내는 역할.
 - (5) $A = 2 \times 3$ $x = 3 \times 1$ $b = 2 \times 1$ 의 matrix, grid를 그렸을 때 A의 점이 grid의 방향으로 움직임. A의 벡터가 grid와 평행하게 움직인다면, output인 b에 영향을 미치지 않는다.
- 2. Eigenvector
 - (1) <http://setosa.io/ev/eigenvectors-and-eigenvalues/> 에서 그래프 참조
 - (2) 두번째 그래프에서 v를 조절하다가 원점과 입출력이 일직선상에 생김. Grid 위에 v를 놔둘시 그렇다. 맨처음 Vector는 matrix의 종류라고 정의했음. 좀더 정확한 정의는 방향이라고 생각해라.
 - (3) Eigen value : Eigen vector space에서 원점과 v의 거리와 v와 A_v 의 거리의 비율이 일정함.
 - (4) Eigenvector의 활용 : 매트릭스는 벡터들의 결합. 벡터들의 고유한(unique) 성질을 뽑아내는 것.
- 3. 상관관계
 - (1) 상관관계의 범위 : $-1 \leq r \leq 1$ // (단 r은 0이 될 수 없다.)
 - (2) 이차원 평면의 두 점에서의 상관관계 : $r = \cos\theta$
 - (3) 두 점의 θ 가 90도, 즉 $\cos\theta$ 가 0이 된다면 두 점의 상관관계가 없다는 말과 같다.
- 4. Inner Product
 - (1) 평면이 아닌, 고차원에서 θ 값을 구하기 위한 방법
 - (2) Inner Product의 대수적 개념 : $[1,2,3]$ 과 $[4,5,6]$ 의 벡터에서 $1*4 + 2*5 + 3*6 = 32$ 의 결과가 나오는 것. 각각의 component를 곱해 더하는 것이라고 생각
 - (3) Inner Product의 기하적 개념 : 두 점 a, b 에서 a까지 거리 $\times \cos \theta \times$ b까지 거리

영어음성학 Summary

2014130031 국어국문학과 민종원

- (4) 고차원에서 한 점까지의 거리 : 벡터 a가 (q,w,e,r)로 이뤄졌다 가정하면, 원점에서 s까지 거리는

$$s = \sqrt{q^2 + w^2 + e^2 + r^2}$$

5. 스펙트로그램

- 각 Hz 별로 inner product를 구하면 해당 hz에 얼마나 분포했는지 알 수 있음
- 이를 쌓아나가면 스펙트로그램이 된다.
- 그래프 a, b, c에서 a * c는 a * b보다 작음. 불일치가 존재하기 때문

10주차

- 스펙트로그램 만들기

1. Phasor의 두 가지 유형

- 1) Simple Phasor : 사인이나 코사인으로 만든 Phasor.
- 2) Complex Phasor : $e^{j\omega t}$ 로 표현됨. 복소수 Phasor.
- 3) inner product의 결과로 복소수 값이 나온다면? 실수 그래프에 표시할 수 없다.
- 4) 위의 문제를 해결하기 위해 Complex plane에 plotting 후, 점과 원점 사이의 거리를 구함.
- 5) Complex Plain에서 점과 원점 사이의 거리는 $e^{j\omega t}$ 로 표현된 inner product 결과의 절대값

2. 코드 설명

for n in range(0,nFFT): ## 샘플 수만큼 for 루프 돈다

 omega = 2*np.pi*n/nFFT # angular velocity

 z = np.exp(omega*1j)**(np.arange(0,nSamp)) ## complex phasor를 만드는 줄이다.

(e)^{jωt} * [0...100]. ω는 오메가. 100은 예시

첫번째 루프에서 omega = 2파이 x 1 / 100

z = e^{j2파이}[0...1]j = e^{j0...2파이}j. 백개의 샘플 가지고 complex plain 한바퀴 도는 거.

두번째 루프에서는 e^{j0...4파이}j 되는 거. 두바퀴 도는 거. 이제 샘플 개수 n까지 가면서 n바퀴 도는 거임.

 amp.append(np.abs(np.dot(s,z))) ## abs는 절대값.

cf) 시험문제 amp에는 허수가 있는가? ㄴ ㄴ. 그럴 수가 없다.

3. Amplitude Plotting

fig = plt.figure()

ax = fig.add_subplot(111)

freq = np.arange(1,nFFT+1)*sr/nFFT;

ax.plot(freq, amp)

ax.set_xlabel('frequency (Hz)')

ax.set_ylabel('amplitude')

그래프의 결과는 좌우 대칭인데 Nyquist frequency와도 관련있다. 좌우 대칭이라 절반 기준으로 한쪽은 의미없는 반복이라고 해석할 수 있음.

주목해야할 포인트는 각 스펙트럼이 얼마나 많이 섞여있는지 파악하는 것. 음악 어플의 이퀄라이즈 기능을 생각하면 된다.

영어음성학 Summary

2014130031 국어국문학과 민종원