Chapter 2

입/출력 end 복잡도 분석

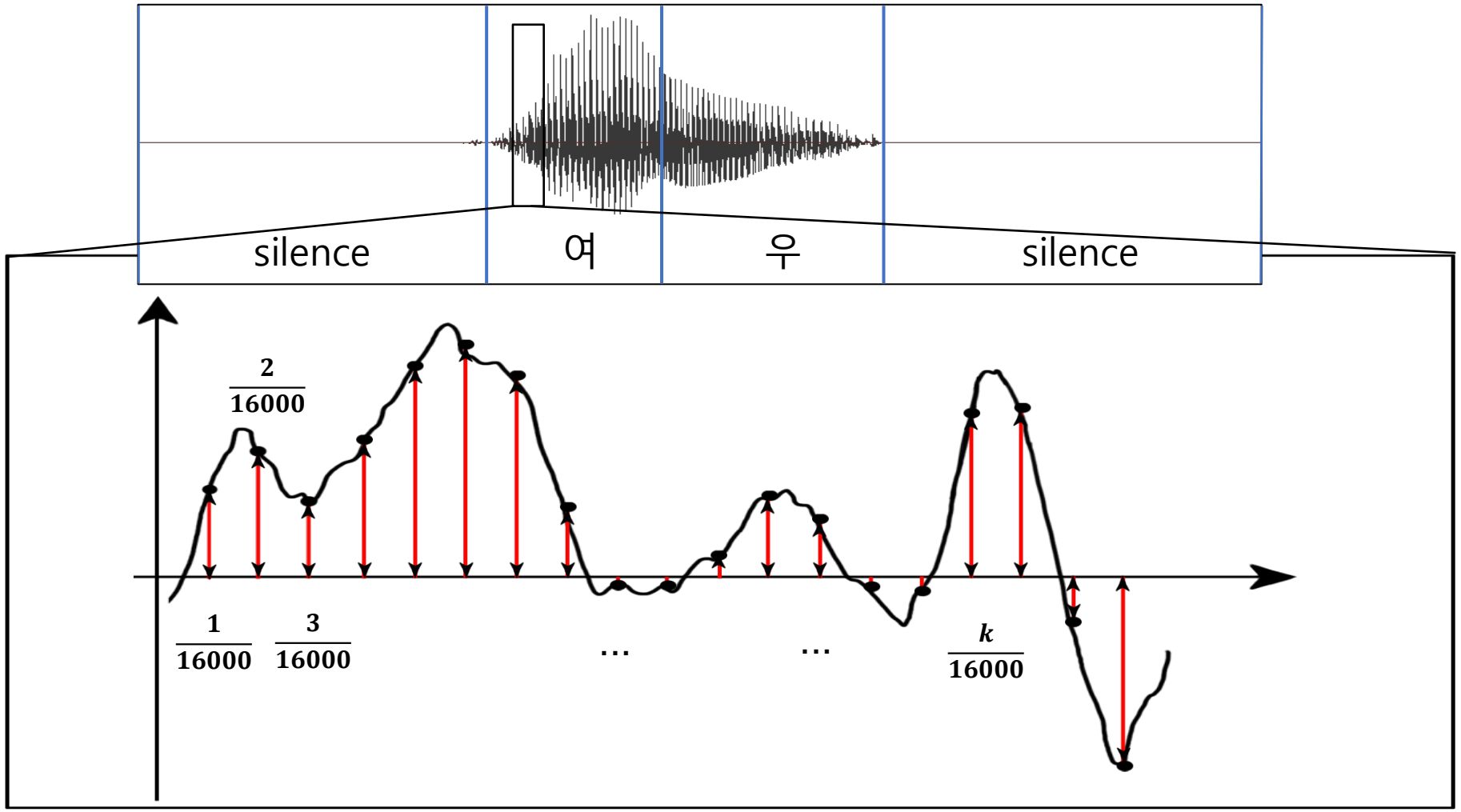
김지환

서강대학교 컴퓨터공학과

Table of contents

2.1 음성 신호 저장 방법 및 시스템 입력

- 음성 신호의 저장 방법 (sampling rate: 16K, PCM)



2.1 음성 신호 저장 방법 및 시스템 입력

- 음성 신호 저장 시 필요한 parameter (PCM, Pulse-Code Modulation 기준)
  - Sampling rate
    - 단위 시간당(초당) sampling의 횟수
    - Nyquist theorem
      - Sampling rate의 절반의 해당하는 주파수 대역까지 복원 가능함
    - Sampling rate는 음질을 결정한다.
      - 음악 저장에 많이 사용되는 sampling rate은 초당 44.1k
        - 사람의 가청 주파수 대역은 일반적으로 20Hz~20kHz로 알려져 있음
      - 전화의 sampling rate : 초당 8k
      - 현재 음성 인식에 많이 사용 되는 sampling rate는 초당 16k임
  - Sample 당 byte수
    - Sample당 2byte 사용 ($2^{16} = 65{,}536$)

2.1 음성 신호 저장 방법 및 시스템 입력

- 음성 신호의 저장 방법(예제)
  - 가수가 10곡을 수록한 앨범을 발매했다. 이를 CD에 담았을 때, 전체 CD 중 burning된 부분은 CD 전체 면적의 몇 %인가?
    - 한 곡의 길이는 4분으로 가정
    - CD는 총 700MB를 저장한다고 가정
    - Sampling rate: 초당 44,100
    - Sample당 2byte 사용
    - Stereo로 녹음되었기 때문에 채널은 2개

    - 44,100(samples/sec) * 2(bytes) * 240(secs) * 10(곡) * 2(channels) = 423,360,000
      - 답 : 423.36/700 MB, 60.48%

2.1 음성 신호 저장 방법 및 시스템 입력

- 음성인터페이스의 가능한 입력 개수 분석(계속)
  - 정량적 분석
    - 가정 : 입력 길이 1초, sampling rate 44.1k, sample당 2B사용
    - 저장에 1 * 44100 * 2 = 88,200B 필요
    - 가능한 입력의 개수: $2^{88200*8}$
    - 입력 길이의 제한이 없으므로 가능한 입력의 개수: 무한대

- 용량을 최대한 줄이면서, 음성 인식에 유용한 정보는 최대한 유지하는 feature추출이 필요
  - 전체 음성을 20ms단위의 window로 나누고, 각 window별로 13차 MFCC feature를 추출하여 사용함

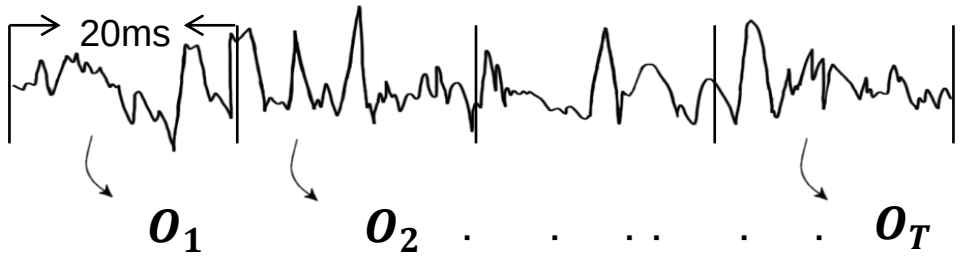
2.1 음성 신호 저장 방법 및 시스템 입력

- 왜 window별로 feature를 추출하는가?
  - Feature추출이 가능 하려면, modeling과 분석이 가능해야 함
  - 현재까지의 발화를 바탕으로 미래의 발화를 예측할 수 있는가?
    - 화자가 발화를 중지할 수 도 있고, 발화의 내용을 갑자기 바꿀 수 있음
      따라서, 예측 가능하지 않음
    - 예측 가능하지 않다면, modeling과 분석이 불가능함
  - 사람의 성대를 하나의 발성 기관으로 본다면, 발성 기관의 물리적 한계로 인하여 어떠한 짧은 시간에 대하여 미래의 발화를 예측할 수 있음
    - 음성은 quasi-stationary함
    - 다양한 실험을 바탕으로 window의 길이는 20ms로 도출됨

2.1 음성 신호 저장 방법 및 시스템 입력

- 음성 신호를 short integer type의 2차원 array로 저장함

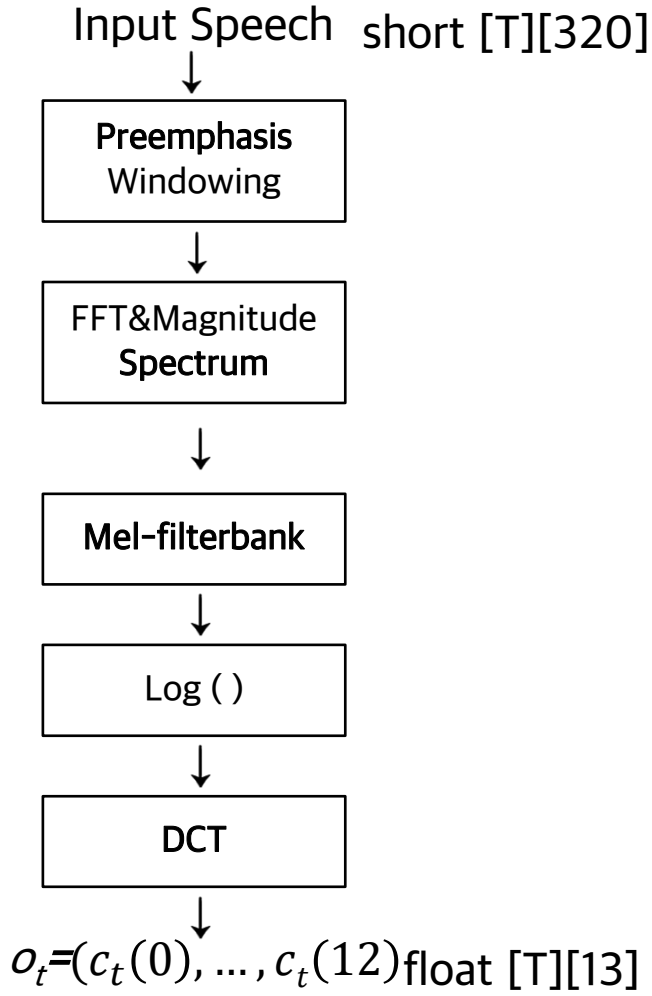
$$\boldsymbol{O} = \boldsymbol{o_1}, \ldots, \boldsymbol{o_T}$$ (*T*: window의 개수)



- 하나의 window에는 320개의 sample이 들어감
- 2byte(short integer)형식의 2차원 array로 저장
  - C 언어에서는 short[T][320]의 형태로 표현됨

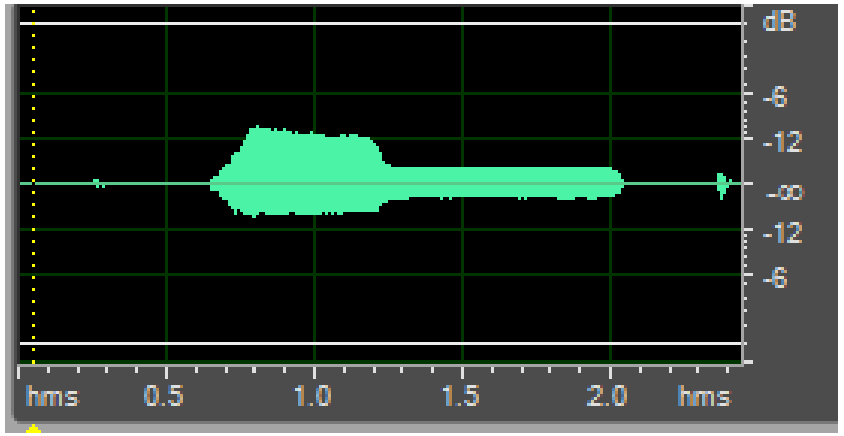
2.1 음성 신호 저장 방법 및 시스템 입력

- MFCC feature 추출 과정

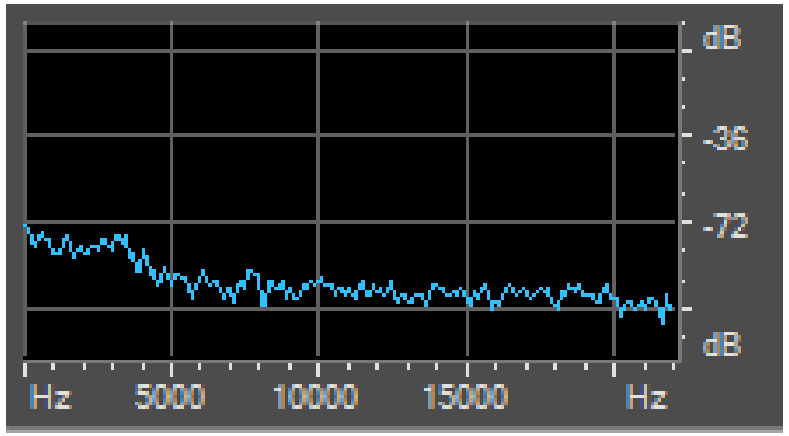


2.1 음성 신호 저장 방법 및 시스템 입력

- Fast Fourier Transform(FFT)
  - 기본 가정:
    - 모든 주기적인 신호는 정현파(ex> sin, cos)의 합으로 나타낼 수 있다
    - 모든 신호를 $\alpha * \sin(\beta * \pi)$ 의 합의 형태로 표현 가능
  - Time domain의 data를 frequency domain의 data로 변형
    - Frequency domain으로 나타내어진 결과를 spectrum이라고 함



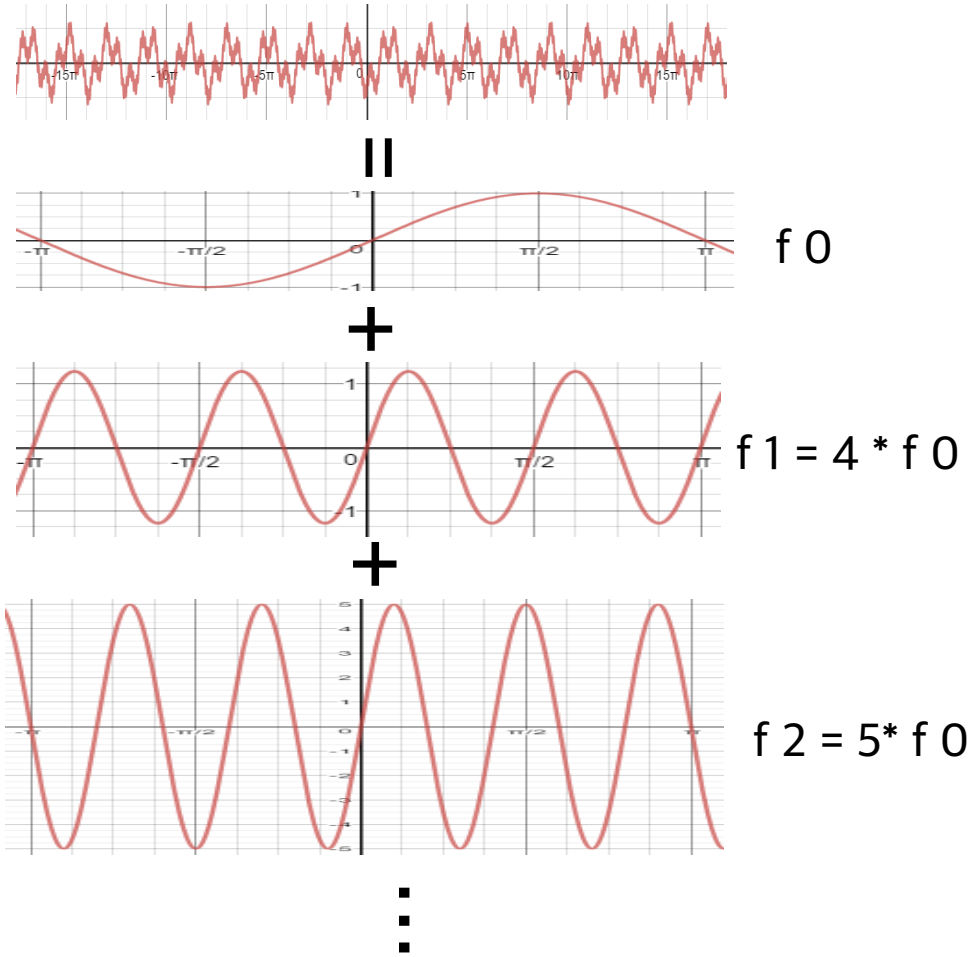
Time domain

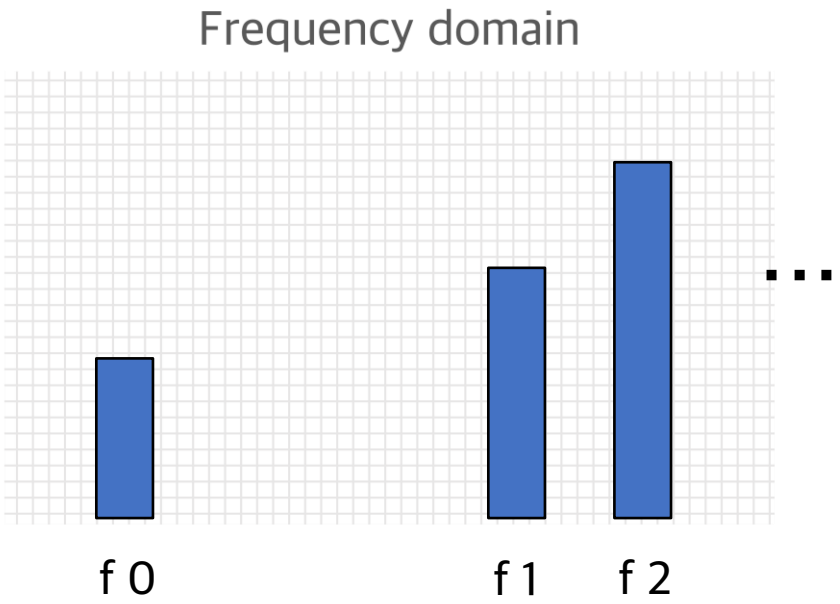


Frequency domain

2.1 음성 신호 저장 방법 및 시스템 입력

- Fast Fourier Transform(FFT)



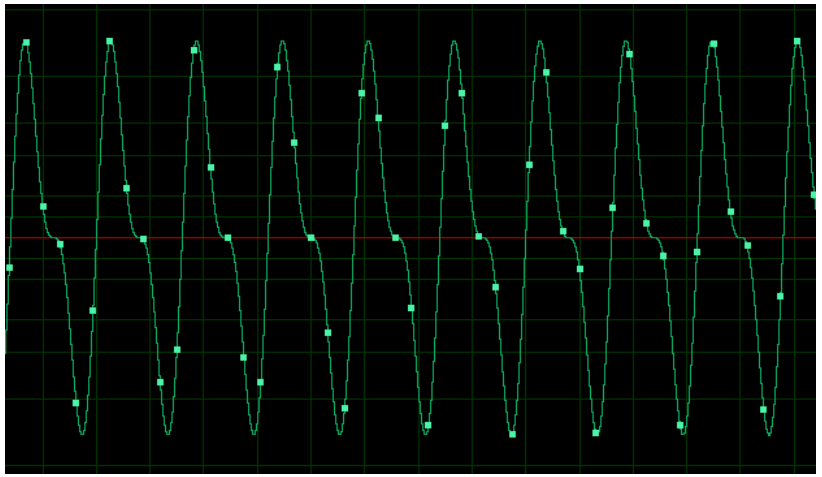


2.1 음성 신호 저장 방법 및 시스템 입력

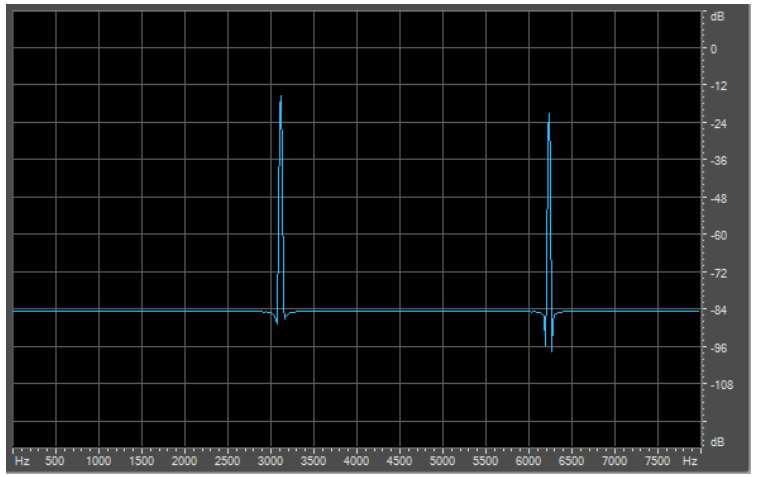
- 예제
  - 음성 신호에서 fundamental frequency를 정의하려면?
    - 일반적으로 사용하는 16kHz sampling rate의 경우, 8kHz까지 복원 가능함
    - FFT 분석을 위해 8kHz를 일정 크기로 분리 (fft size: 2^10개)
    - FFT 분석에는 복소수가 사용되는데, 실수와 허수의 벡터 성분이 대칭을 이루기 때문에 중복되는 값을 버리고 실수 값만을 사용
      - (fft size/2 = 2^9개)
    - fundamental frequency : 8000/512 = 15.625Hz
    - 음성 신호에서의 frequency는 15.625의 배수로 나타냄

2.1 음성 신호 저장 방법 및 시스템 입력

- 예제
  - 3125Hz, 6250Hz의 frequency를 가진 sine wave를 중첩 시킨 pcm



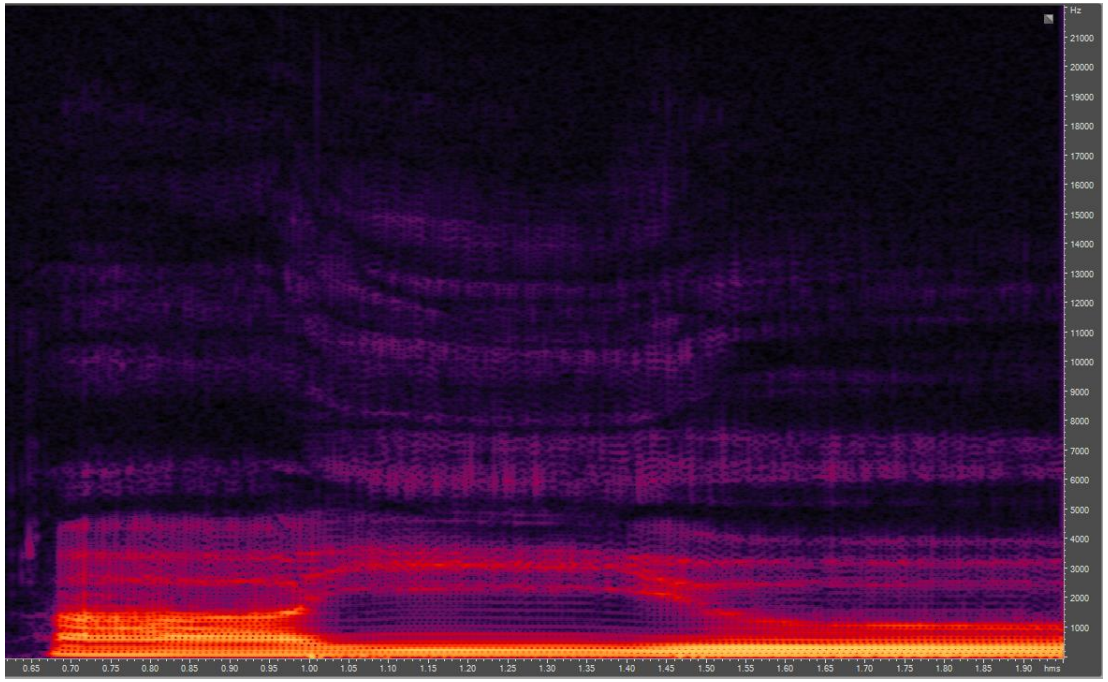
Time domain



Frequency domain

2.1 음성 신호 저장 방법 및 시스템 입력

- Time과 frequency domain을 각 축으로 표현하고, amplitude를 색으로 표현한 것을 spectrogram이라 함
- FFT를 통해 음성 신호의 특징을 표현할 수 있음



‘아이유’에 대한 spectrogram

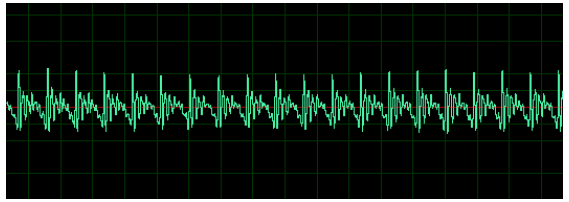
2.1 음성 신호 저장 방법 및 시스템 입력

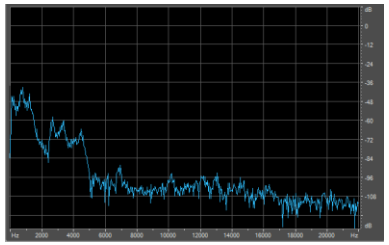
- 음소별로 spectrum의 모양이 다름

Time domain

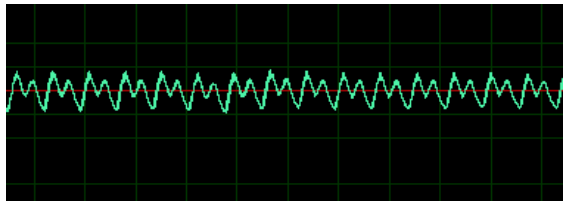
Frequency domain

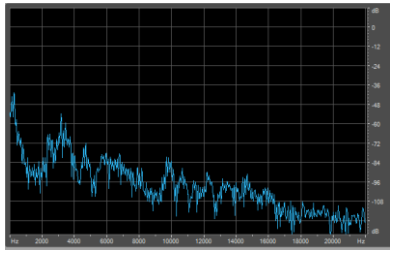
"아"



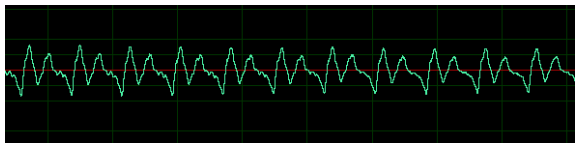


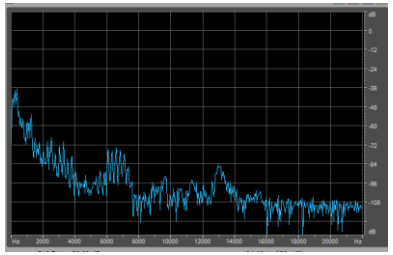
"이"





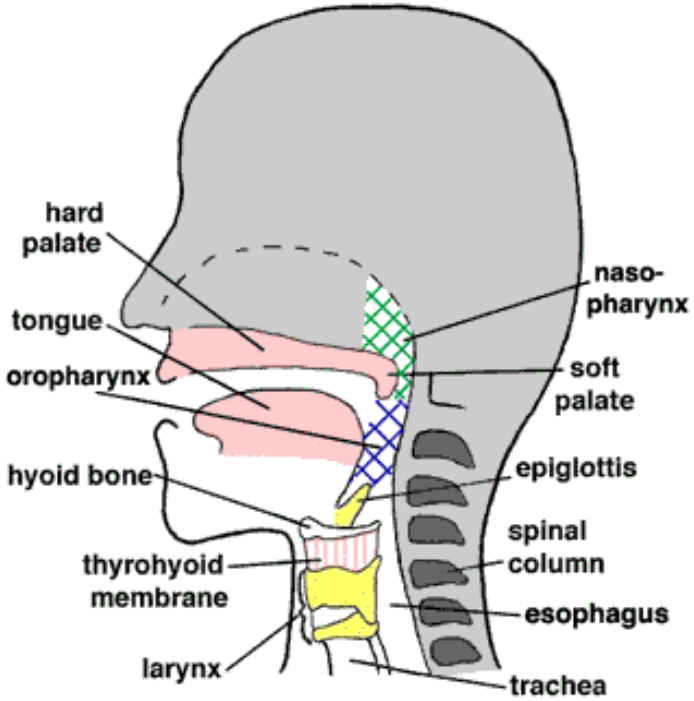
"유"





2.1 음성 신호 저장 방법 및 시스템 입력

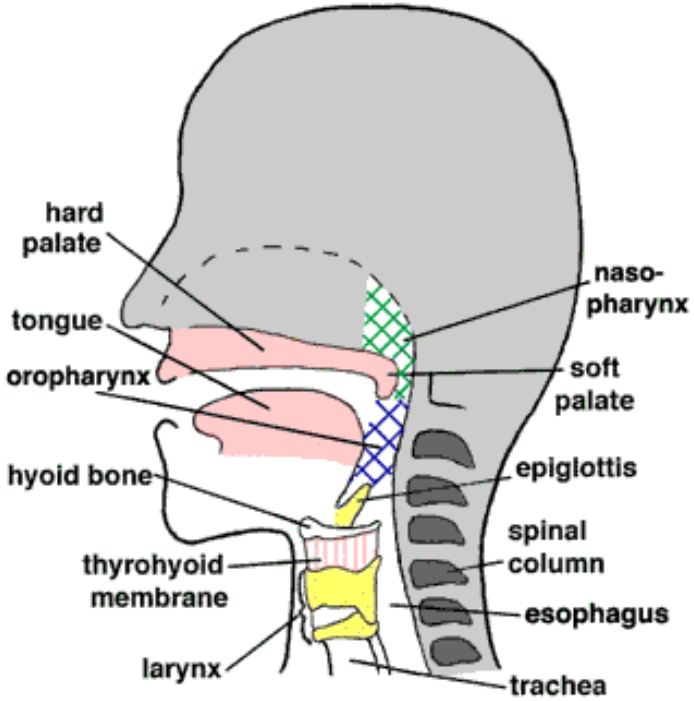
- Mel-filterbank & Log
  - x(n) = v(n)*e(n) 을 음성이라고 가정
    - v(n) = 구강구조
    - e(n) = 성대에서 울리는 소리
  - v(n)은 음성에 대한 정보
    - 무슨 말을 했는지에 대한 정보
  - e(n)은 화자에 대한 정보
    - 어떤 사람이 말을 했는지에 대한 정보



http://www.lionsvoiceclinic.umn.edu/page2.htm#anatomy101

2.1 음성 신호 저장 방법 및 시스템 입력

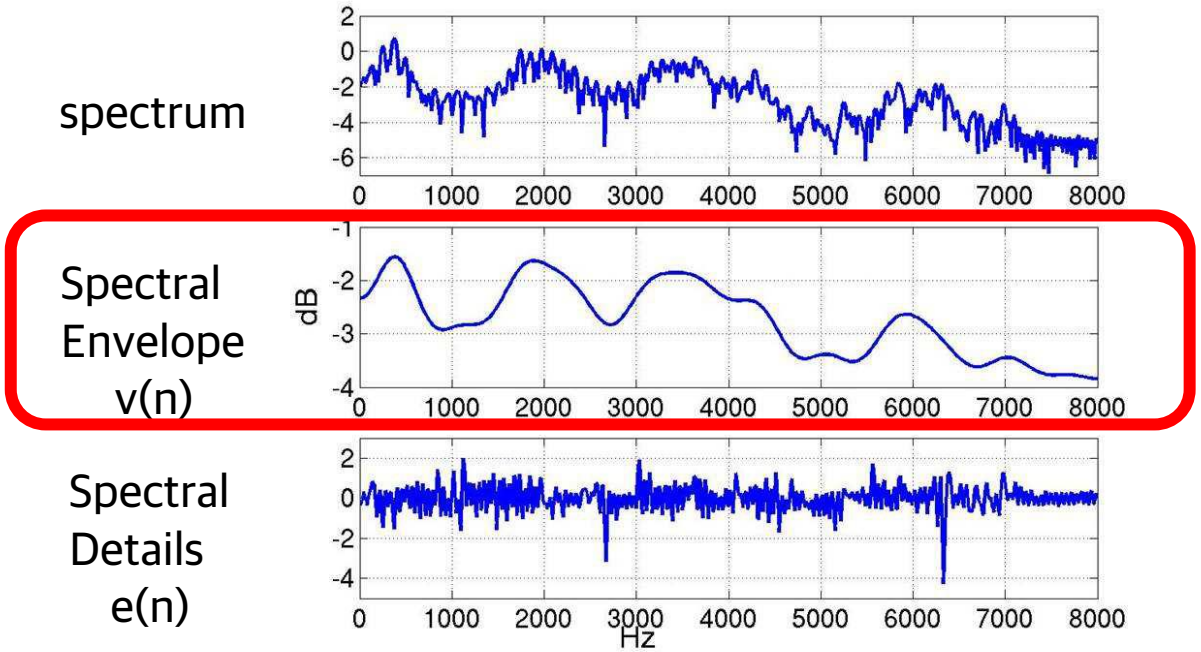
- Mel-filterbank & Log
  - x(n) = v(n)*e(n) 이 마이크를 통해 입력됨
    - Time domain
  - FFT를 통해 frequency domain으로 변환
    - convolution이 곱셈으로 변환
  - X(n)에서 화자 정보인 e(n)을 제거하기 위해 log를 씌워 덧셈으로 변환
    - Log(X(n)) = Log(V(n)) + Log(E(n))



http://www.lionsvoiceclinic.umn.edu/page2.htm#anatomy101

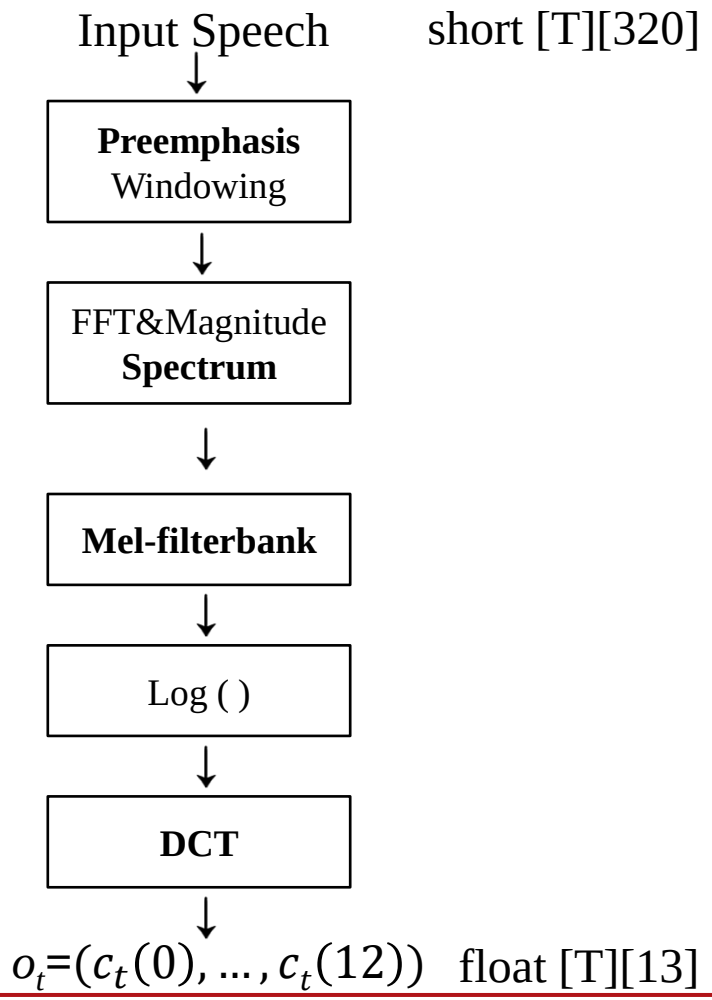
2.1 음성 신호 저장 방법 및 시스템 입력

- Discrete Cosine Transform(DCT)
  - 적은 차원의 데이터로 envelop을 구할 때 사용됨
  - Discrete Cosine Transform(DCT) 과정을 통해 음성 인식에 있어 필요한 정보만을 담은 13차 vector를 생성



2.1 음성 신호 저장 방법 및 시스템 입력

- MFCC feature 추출 과정



2.2 음성 출력 단위 결정

- 각 단어들은 컴퓨터 내부에서 어휘에 대한 index로 표현
  - 어휘(Vocabulary): 인식 가능한 단어들의 집합
  - 예: 어휘의 크기가 10만이고, 단어들은 어절단위로 구분 되었으며, 어휘 내 단어들을 가나다순으로 정렬했을 때, 단어들이 아래의 순서의 단어로 어휘 내 위치해 있다고 가정한다.
    - '가자': 12,844번째 단어
    - '내일': 24,882번째 단어
    - '세시에': 35,493번째 단어
    - '오후': 69,864번째 단어
    - '학교': 95,867번째 단어
  - '내일/오후/세시에/학교/가자'는 아래와 같이 index의 열로 표현된다.
    - 24,882/69,864/35,493/95,867/12,844

2.2 음성 출력 단위 결정

- 한국어 문장 예제: (형태소 단위로 시퀀스를 구성)
  - 올 여름 평년보다 덥고 강수량 지역 차 크다
    - <s>/올/여름/평년/보다/덥/고/강수량/지역/차/크/다/</s>
    - w0/w1/w2/w3/w4/w5/w6/w7/w8/w9/w10/w11/</s>

- 영어 문장 예제: (단어 단위로 시퀀스를 구성)
  - It is hotter than usual this summer, and the regional difference of precipitation is big
    - <s>/It/is/hotter/than/usual/this/summer,/and/the/regional/difference/of/precipitation/is/big/</s>
    - w0/w1/w2/w3/w4/w5/w6/w7/w8/w9/w10/w11/w12/w13/w14/w15/w16