

BMED311 Week04-2 LabP5 Summary

목적: sinusoids를 이용하여 음악을 합성(Music Synthesis using Sinusoids)한 후 연주함

$$x(t) = \sum_k A_k \cos(\omega_k t + \varphi_k)$$

실습 단계:

1. 다장조의 음계(도레미파솔라시도)를 생성하여 연주
2. 실제 음악(바흐의 Well-Tempered Clavier – Fugue #2, bach_fugue_short.mat)의 악보정보를 읽어서 각 음표를 sinusoid로 변환하여 합성하여 연주

1단계: 다장조 음계 생성 및 연주

- ① 피아노 건반의 이해: 2.5
 - Key numbers(40-52), key notation (C4–C5)
 - Tuning A: A4 or A-440 : 440 Hz
 - the ratio between the frequencies of the adjacent key numbers: $2^{\frac{1}{12}}$
- ② 악보의 이해: 2.5
 - Note: 음표, eighth note, quarter note, half note
 - Tone: 온음(간격) 또는 음(pitch), beat: 박자, 4분음표, pulse: 16분음표
- ③ 음악 음표와 주파수의 관계 이해 및 하나의 음표를 sinusoid로 변환: 3.1
 - 피아노 키 번호(key number)를 주파수로 변환하는 공식 작성
 - key2note() 함수를 구현: 주어진 key number와 지속 시간(duration)에 맞는 sinusoid 생성하여 return
 - input parameters: phasor, key number, duration
 - output: cosine data xx
- ④ 다장조 음계 신호 생성 및 연주: 3.2
 - 사인파 신호를 지정된 주파수, 위상, 진폭으로 생성: phase = 0, amplitude = 1
 - soundsc() 함수를 이용하여 사운드를 출력
 - soundsc - 데이터 스케일링 및 소리로 재생
 - 오디오 신호 y의 값을 -1.0 ~ 1.0의 범위에 맞게 스케일링한 다음 이 데이터를 8192Hz의 디폴트 샘플 레이트로 스피커로 전송합니다.
 - soundsc(y) : default sampling frequency 적용
 - soundsc(y, Fs) : sampling frequency 지정
- ⑤ 생성된 다장조 음계 신호를 spectrogram으로 보기: 3.3 (a)-(b)
 - Spectrogram을 zoom in 하여 A-440 구간을 확대해서 frequency, duration 등 확인
 - Spectrogram figure에서 colormap을 변경하여 확인

2단계: fugue_short.mat을 읽어서 sinusoids로 합성(synthesis)하여 연주

- ⑥ 각 음표(note)에 해당하는 cosine signal의 sampling frequency and period 결정: 1초에 몇 개의 discrete data를 생성할지 결정: 4.(a)
- ⑦ fugue_short.mat을 load후 파일 포맷 이해: 4.(b) and 4.3
 - theVoices(i).noteNumbers : 각 음표의 건반 번호를 기록한 array
 - theVoices(i).startPulses : 각 음표의 시작 시간을 pulse(16분음표)의 개수로 기록한 array
 - theVoices(i).durations : 각 음표의 연주 시간을 pulse의 개수로 기록한 array

- bpm(beats per minute): 분당 연주할 4분음표 개수
- ppq(pulses per quarter note) = 4: 4분음표 당 4개로 정의 즉 1 pulse = 16분음표
- ⑧ theVoices(i)를 위한 sinusoids의 길이를 결정하여 blank xx(i) array 생성(initialization): 4.(b)
 - seconds_per_pulse를 bpm으로부터 계산
 - 각 음표의 시작 index 결정: $\text{seconds_per_pulse} * \text{theVoice}(i).\text{startPulses} * f_s$
 - 각 음표의 연주 시간동안 data 개수 결정: $\text{seconds_per_pulse} * \text{theVoice}(i).\text{durations} * f_s$
- ⑨ 음계 합성: 기존의 play_scale.m을 수정하여 theVoice(i)에 기록된 대로 waveform을 합성: 4.(c)
- ⑩ 합성 신호 연주: 4.(c)
- ⑪ 합성신호 검증: 몇 개의 sinusoid를 3 periods 떼어내어 plotting하여 frequency/period 확인
- ⑫ Spectrogram 분석: zoom in 이용