Periodic signals (주기 시그널) : 반복되는 시그널 (Sine, Square, Triangle, Sawtooth)

Cycle : 반복되는 complete한 주기

Period : 한 Cycle 지속시간(T) == f(t) == f(t+T)

Diagram

Description automatically generated with medium confidence

Frequency : 초당 사이클 횟수 == Period의 역수 == 1/T

Phase(위상) : 반복되는 파형의 한 주기에서 첫 시작점의 각도 혹은 어느 한 순간의 위치를 말한다. ( 2파이 \* f \* t), 즉 wave의 한 cycle == 360도 == 2파이 라디안

구하는 예시

Diagram

Description automatically generated

360 도 == 2파이 \* radiars

2파이 \* (0.5의 역수 == 2) \* 1/8 == 파이/4

뮤지컬 공연의 periodic signal은 sinusoidal 하지 못한다(sine, cosine wave)

이러한 periodic signal은 wave form이라 부른다

사람들은 복잡한 waveform일 수록 좋게 지각한다.

Spectral decomposition

-> 어떤 신호든 sinusoids 신호의 다른 frequencies로 표현할 수 있다.

->어떤 신호든 sinusoids의 합으로 나타낼 수 있다.

Diagram

Description automatically generated

DFT(Discrete Fourier Transform)

Diagram

Description automatically generated

spectrum == 이걸 다 더하면 signal로 다시 복구 가능하다

Fundamental frequency : 가장 낮은 frequency 성분

Dominanat frequency : 가장 큰 힘

보통은 둘이 같음 but dominant < fundamental

Harmonics(하모닉) : fundmental의 정수배 ex) 440, 880, 1320, ~~

thinkdsp 코드

sine. cosine signal 만드는 방법

cos\_sig = thinkdsp.CosSignal(freq = 440, amp=1.0, offset=0)

sin\_sig = thinkdsp.SinSignal(freq = 880, amp=0.5, offset=0)

amp -> 진폭

offset -> phase(angle) -> 0이면 위상차 없이

만약 pi/2 였다면 1늦게 시작

wave = mix.make\_wave(duration=0.5, start=0, framerate = 11025)

duration -> 만드는 범위, ex) duration = period\*3 이면 3 cycle 반복

ts1초 == 11025

read : violin\_wave = thinkdsp.read\_wave('input.wav')

write : wave.write(filename='output.wav')

play : thinkdsp.play\_wave(filename='output.wav',player='aplay')

spectrum = wave.make\_spectrum()

low\_pass == cutoff freq기준으로 얘보다 낮은거 0으로

high\_pass == cutoff freq기준으로 얘보다 높은거 0으로

band\_pass == 중간 영역

spectrum.low\_pass(cutoff=600,factor=0.01)

0.01 - > 다 없애기는 힘드니 99% 없애라

framerate == 표본집단->sampling 한다