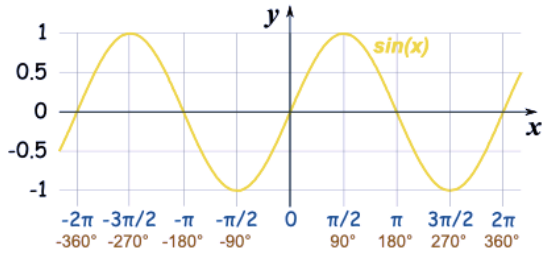
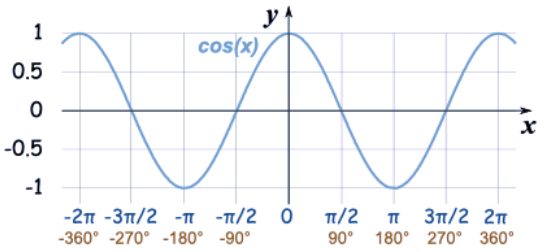
[2019-2 영어음성학]

이름(학번): 민지우(2016130575)

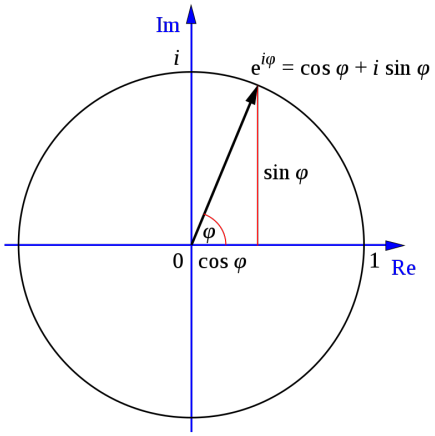
Week7 Summary

Sound





- Sine curve
  - a curve in rectangular coordinates of the equation $y=a \sin bx$ where a and b are consonants
  - it starts at 0, heads up to 1 by π/2 radians (90°) and then heads down to –1.
- Cosine curve
  - a curve whose equation in Cartesian coordinates is of the form $y = a \cos x$
  - it starts at 1 and heads down until π radians (180°) and then heads up again.
- Phasor
  - A line used to represent a complex electrical quantity as a vector
- Euler's formula



  - $f(\theta) = e^{i\theta} = a + bi = \cos(\theta) + \sin(\theta)i$
  - for any real number (θ), where i is the imaginary unit
  - θ = 0, π/2, π, 3π/2, 2π

Parameter Setting

- amp = 1 *# range [0.0, 1.0]*
- sr = 10000 *# sampling rate, Hz*
- dur = 0.5 *# in seconds*
- freq = 100.0 *# sine frequency, Hz*

1. Generate time

- t = np.arange(1, sr+1) #sampling rate만큼의 time tick/ 1초면 *1
- t = np.arange(1, sr * dur+1)/sr #duration만큼 1초면 *1, 0.5초면 *0.5: *duration, /sr
- t 0.0001 0.0002 ... 0.5000: samling rate이 1초에 10000번 들어가야 하는데, duration이 0.5니까 1초에 10000개 단위로 찍히나 1초까지 안 간다. Time tick을 0.5까지만 만든다. 1~ 5000, index는 있으나 실제 time은 아니니, sampling rate으로 나눠야 한다.

이름(학번): 민지우(2016130575)

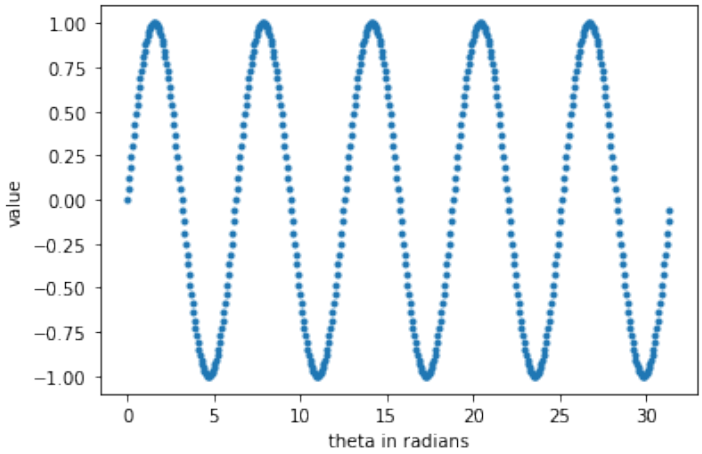
0부터 2*np.pi 하면 시간 개념 전혀 안 들어 있어서, t를 가지고 theta로 확장 시켜야 한다.

2. Generate phase

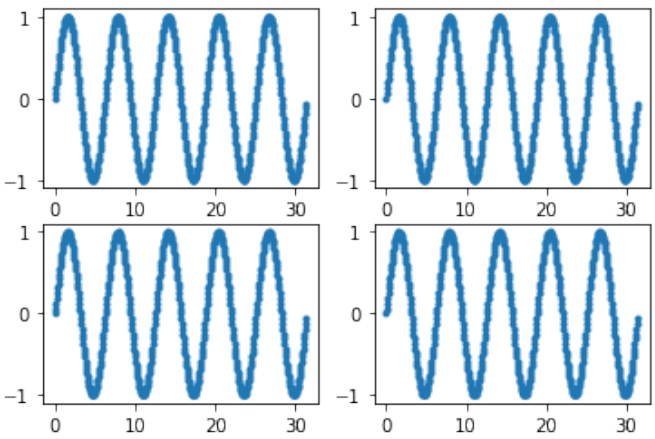
- theta = t * 2*np.pi * freq #np.pi np 안에 있는 pi를 불러온다. 2*np.pi 사인 한 바퀴 몇 바퀴? *freq
- 1초에 2pi ~ 한 바퀴 도니 ~ * frequency = 총 만들 순환 수

3. Generate signal by sine-phasor

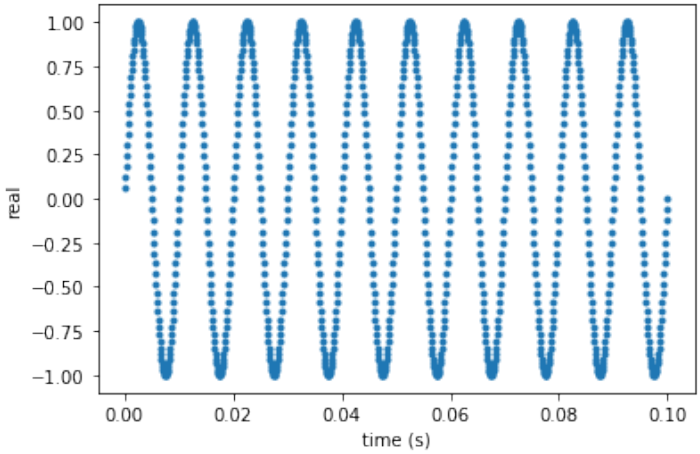
- **1. time 만들기 2. phase 연동(각도)**
- theta = np.arange(0, 2*np.pi*주기, 2*np.pi*주기/간격) #radian
- s = np.sin(theta)
  - time 벡터 크기와 theta 벡터 크기는 같다
- fig = plt.figure() # figure은 화면 전체
- ax = fig.add_subplot(111)
  # x축에서 equidistance 한데, y축에서 equidistance하지 않다. linear function이면 x, y equidistance 하나, curve이기 때문에, x equidistance 성격이 y에 적용되지 않는다.
- ax.plot(theta, s, '.')
- ax.set_xlabel('theta in radians')
- ax.set_ylabel('value')



- fig = plt.figure()
- ax = fig.add_subplot(221)
- ax.plot(theta, s, '.')
- ax = fig.add_subplot(222)
- ax.plot(theta, s, '.')
- ax = fig.add_subplot(223)
- ax.plot(theta, s, '.')
- ax = fig.add_subplot(224)
- ax.plot(theta, s, '.')



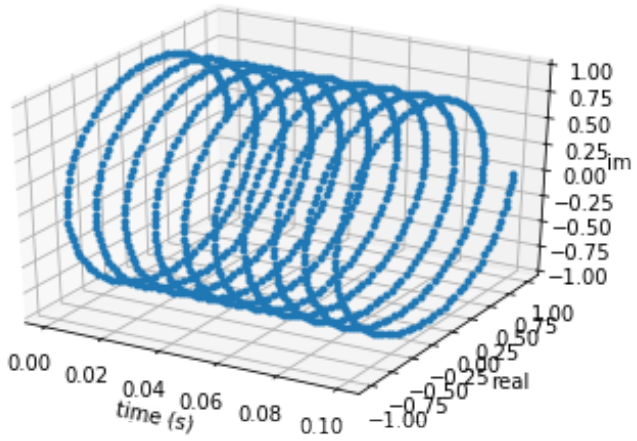
- fig = plt.figure()
- ax = fig.add_subplot(111)
- ax.plot(t[0:1000], s[0:1000], '.') # x축 y축 단위 같아야 한다. 각각 1000개씩
- ax.set_xlabel('time (s)')
- ax.set_ylabel('real')



- ipd.Audio(s, rate=sr)

4. Generate signal by complex-phasor

- c = np.exp(theta*1j) #np.exp가 exponential, e; 1j = i
- fig = plt.figure()
- ax = fig.add_subplot(111, projection = '3d')
- ax.plot(t[0:1000], c.real[0:1000], c.imag[0:1000], '.')
  # complex 대표하는 c라는 변수로 받음
  # 복소수는 a+bi니까 c값에 a, b 두개의 값이 나온다; a값이 real 이고 b가 imaginary
- ax.set_xlabel('time (s)')
- ax.set_ylabel('real')
- ax.set_zlabel('imag')



- ipd.Audio(c.imag, rate=sr)