Examples of Programming Skills

Project 1: Electrocardiogram Signal Generator – Python

Project Description:

I established a tool to generate synthesized electrocardiogram signals to assist other researchers in developing analytic tools. The electrocardiogram (ECG) signals are records of the electrical signal from human's heart for examination of heart conditions.

Method:

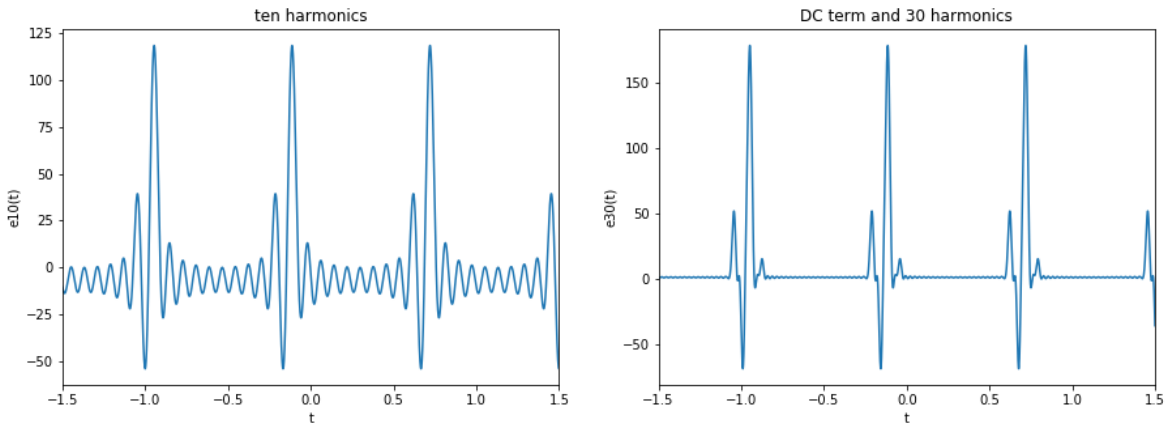
Through a for loop, ECG signals were synthesized by summation of a set of sinusoids as the following equation. The signals of a normal human at rest are roughly periodic signals with 1.2 Hz fundamental frequency (corresponding to a heart rate of 72 beats per minute). Harmonic frequencies of the sinusoids were calculated by the equation $f_k = k \cdot f_0$, where f_0 is the fundamental frequency and given as 1.2 Hz. The parameter φ_k is the phase shift of each sinusoid, while p_k is the magnitude. In addition, n is the number of harmonics considered.

$$e(t) = \sum_{k=1}^{n} p_k \cdot \cos(2\pi \cdot f_k \cdot t + \varphi_k)$$

Demo:

To demonstrate the capacity of this tool, two ECG signals were synthesized and visualized. One considered only first ten harmonics, and the other considered 30 harmonics as well as a direct current (DC), which refers to electric current that never changes the direction of the flow.

Result:



Code:

```python
# -*- coding: utf-8 -*-
"""
2018/10/31 HW2
Fourier Series

"""

import numpy as np
import matplotlib.pyplot as plt

t = np.linspace(-1.5,1.5,20000)
e_a = np.zeros(t.size)
e_b = np.zeros(t.size)
pk = np.array([8, 12.434, 10.775, 9.121, 8.77, 10.053, 12.066, 13.924, 15.069,
               15.143, 14.021, 11.911, 9.361, 7.112, 5.732, 5.121, 4.701,
               4.115, 3.3, 2.283, 1.129, 0.555, 1.489, 2.25, 2.54, 2.349,
               1.815, 1.15, 0.549, 0.25, 0.494])
Ok = np.array([0, 0.912, 1.769, 2.513, 3.165, 3.902, 4.8, 5.799, 0.553, 1.593,
               2.610, 3.585, 4.489, 5.295, 6.029, 0.531, 1.459, 2.519, 3.670,
               4.886, 6.261, 2.713, 4.555, 5.745, 0.545, 1.579, 2.582, 3.573,
               4.660, 0.302, 1.979])

# (a)ten harmonics
for k in np.arange(1,11) :  # k = 1,2,...,10
    e_a = e_a + pk[k]*np.cos(2*np.pi*k*1.2*t + Ok[k])

plt.figure(figsize=(15,5))
plt.subplot(1, 2, 1)
plt.plot(t, e_a)
plt.xlim(-1.5,1.5)
plt.xlabel("t")
plt.ylabel("e10(t)")
plt.title("ten harmonics")

# (b) DC term and 30 harmonics
for k in range(31) :  # k = 0,1,2,...,30
    e_b = e_b + pk[k]*np.cos(2*np.pi*k*1.2*t + Ok[k])

plt.subplot(1, 2, 2)
plt.plot(t, e_b)
plt.xlim(-1.5,1.5)
plt.xlabel("t")
plt.ylabel("e30(t)")
plt.title("DC term and 30 harmonics")

# save the figure
plt.savefig(fname = "HW2.png", format = "png")
```