****

**LABORATORY MANUAL**

**CE3007: Digital Signal Processing**

**Hardware Lab 1 (Location: N4-01a-03)**

**SESSION 2017/2018**

**SEMESTER 2**

**COMPUTER ENGINEERING COURSE**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**NANYANG TECHNOLOGICAL UNIVERSITY**

**LAB – 1**

**DIGITAL SIGNAL PROCESSING USING PYTHON**

**1. OBJECTIVE**

The objective of this laboratory is to learn digital signal processing using python. In this laboratory, we will

1. generate discrete time signal, DTMF (Dual Tone Multi-freqeuncy signaling), adding signals together, finding the period of the discrete time sequence, generating complex exponential signals.
2. visualizing the generate discrete time signal.
3. saving the discrete time signals as wavefiles so that we can listen to them.

Specifically, we will work with python 3.6 under the Anancona environment [1]. You can choose to use the school’s laboratory environment or your own laptop’s environment during the laboratory.

You can read about python and how to use it for development in [2-4]. You can also read about why python vs Matlab here [7-9] and Matlab’s response [10].

Snippets of python code which can help in this laboratory is provided in Lab1Example.py

**2. Expectations of students for the Laboratory sessions.**

***The following rules and regulations apply for all the laboratories pertaining to CE3007.***

***You are expected to have completed all the exercises of the laboratory before attending the laboratory***. The purpose the laboratory is to conduct a laboratory quiz.

If you have questions on how to develop the code, check with the TA (Teaching Assistant) before the laboratory class. If you have not complete the laboratory exercises before coming to lab, you are unlikely to finish the quiz questions.

How we conduct Laboratory quizzes:

1. At the beginning of the laboratory lesson, you will be provided with a questionnaire (aka as the laboratory quiz) and you should answer all the questions.
2. It is an open book laboratory quiz, i.e, you can use your notes, and the internet to help answer the questions.You are however **not allow** to receive help from any other person. The questions will be related to the given tasks of the lab as well as the lectures you have attended.
3. Your answers must be submitted at the end of the laboratory session and will be marked. It will contribute towards your final grade for the subject. Rememebr to write your name and student ID, date, and laboratory number in the submission.
4. If you are absent for the laboratory quiz with valid medical reason, the quiz mark for the absent quiz will be the average of your other quiz marks. Without a valid reason, you will receive 0 marks. Note - there will be no re-schedule of the laboratory quiz even for valid medical reasons. For this course, we will only consider absent with valid medical excuse (and others) for 2 quizes as participation in the quiz is an integral part of the course. If you are absent for more than 2 quizzes (with valid reasons), the subsequent quiz marks will still be 0 – this is in view that the laboratory component is an integral part of the course. Plase submit valid MC and exemptions to be away by emailing them to the TA for the laboratory and cc the laboratory technician in charge.

**3. Developing Pyton programs to generate discrete time signals.**

Develop your own python routines to do the following tasks listed in 2.1-2.5. In other words, you are not allow to use the routines of others, public libraries found in github e.g [6], and etc.

You are however allow to use the following python libraries for generic functions, such as generating sine, cosine, exponential values, plotting figures for visualization, saving sequences as wavefiles, and playing seqeuces into sound card.

**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
**import** scipy.io.wavfile **as** wavfile  
**import** winsound

To help students start on this laboratory, see the provided codes in Lab1Example.py. It shows how you can use python to generate, save a discrete time sequence into a wavefile, plot it, as well as playing it using the sound card. However, do not write your code in this manner – we expect you to write routines which can be reused into Python modules.

3.1 Write a python program to generate a sampled signal, where the continuous time signal is described by , which is sampled at Hz, where F=1000Hz, from t= 0 to 0.5seconds.

* 1. Clearly sketch the first 5 samples of y(t) indicating its y-axis and x-axis values.
  2. How many samples are needed to store J (integer) cycles of y(t)? Provide the equation.
  3. How many samples are needed to store M (float) seconds of y(t)? Provide the equation.
  4. Plot the corresponding sampled sequence for 2 cycles of the signal.
  5. If the parameters for this question are now F=1, and , comment on the new y[n] vs the original parameters of F=1000, Fs=16000.
  6. Write a python program to generate a DTMF sequence, save it in a wavefile (such that we can examine it), play it as an audio (within python). E.g into a module such as

[t,y] = myDSPfn.GenSampledDTMF(‘0123#’,Fs,durTone)

where t stores the sample index, and y stores the sample values.

* 1. Given y1(t) = Acos(2\*pi\*10\*t); y2(t)= Bcos(2\*pi\*15\*t); and y3(t)=y1(t)+y2(t)

sampled at Fs = 60.

1. Find the period of the individual discrete sequence y1[n],y2[n],and y3[n].
2. How are the periods of y3[n] related to y1[1] and y2[n]?
   1. Write a python program to generate and plot the following signals

where A is a real number, digital frequency (in radian/sample), phase shift (radian), for a given range n=0..N, and n,N,K . Visualize the complex discrete sequence y[n] it in 3ways,

* + - 1. 2-D plot of real and imag values in the same figure using different colors.
      2. Polar plots of the sequence.
      3. 3-D plot showing trajectory with respect to sample index.
  1. Let

1. Generate 3-D plots of for N=16, n=0..N-1, plot
2. Comment on their relationship with respect to k and N. Hint – periodicity and number of samples for each sequence’s period.

**4. References**

[1] <https://www.anaconda.com/download/>

[2] “A Crash Course in Python for Scientist” <http://nbviewer.jupyter.org/gist/rpmuller/5920182>

[3] “Scientific Computing with Python”, <http://nbviewer.jupyter.org/url/atwallab.cshl.edu/teaching/QBbootcamp3.ipynb>

[4] “Python for Signal Processing”, José Unpingco, <http://www.springer.com/gp/book/9783319013411>

[5] “Think DSP – digital signal processing in python”, Allen Downey, <http://greenteapress.com/wp/think-dsp/>

[6] Pho – python for DSP, https://github.com/belangeo/pyo

[7] “Adventures in Signal Processing with Python” , Jason Sachs, June 2013, https://www.embeddedrelated.com/showarticle/197.php

[8] “I used Matlab. Now I use Python”, Steve Tjoa, Sep 2010. https://stevetjoa.com/305/

[9] <http://phillipmfeldman.org/Python/Advantages_of_Python_Over_Matlab.html>

[10] “Matlab vs Python: Top reasons to choose Matlab” <https://www.mathworks.com/products/matlab/matlab-vs-python.html>

Some remarks:

You will be asked to write the routines of generating discrete time sequences of cosine, sine, complex exponential from scratch – hence you are not allowed to call routines provided for by many available libraries, e.g as in [4-6].