UNIVERSITY OF THE PHILIPPINES DILIMAN
ELECTRICAL AND ELECTRONICS ENGINEERING INSTITUTE

EE 274 / COE 197E: Discrete Time Signals & Systems (DSP1)
Programming Exercise 01
DEADLINE: 02-OCT-2020

INSTRUCTIONS. This is an open-notes, open-books exercise, and can be done solo or by pair (For those under EE 274, you are required to do this individually). Write your name/s and student number/s on the topmost part of the lab report. The exercises are designed to be done in MATLAB or Octave. For convenience, use the **Publish** to **DOC/PDF** feature of MATLAB. Comment your answers to the questions.

Submit copies of your well-commented codes and automatically generated lab report in a compressed folder via UVLE no later than the designated deadline. Use the filename EE274_ProgEx01.zip or CoE197E_ProgEx01.zip. Anything submitted beyond the deadline will get a 0.6 multiplier penalty to the final score.

A. SIGNAL GENERATION

In this exercise, you will demonstrate your coding skills in MATLAB by generating the following signals:

`[x,n] = impseq(n0,n1,n2)`	Unit-impulse sequence generation
`[x,n] = stepseq(n0,n1,n2)`	Unit-step sequence generation
`[y,n] = sigadd(x1,n1,x2,n2)`	Addition of two sequences
`[y,n] = sigmult(x1,n1,x2,n2)`	Multiplication of two sequences
`[y,n] = sigshift(x,m,n0)`	Shifting operation on a sequence
`[y,n] = sigfold(x,n)`	Folding operation on a sequence
`[xe,xo,m] = evenodd(x,n)`	Decomposition or real sequence into even and odd components

B. SIGNAL REPRESENTATION

Generate and Plot the following signals. You may use your functions in Part A.

Function	Domain
$x_1(n) = \sum_{m=0}^{10} (m+1)[\delta(n-2m) - \delta(n-2m-1)]$	$0 \leq n \leq 25$
$x_2(n) = n^2[u(n+5) - u(n-6)] + 10\delta(n)$ $+20(0.5)^n[u(n-4) - u(n-10)]$	$0 \leq n \leq 25$
$x_3(n) = (0.9)^n \cos\left(0.2\pi n + \frac{\pi}{3}\right)]$	$0 \leq n \leq 20$
$x_4(n) = 10\cos(0.0008\pi n^2) + w(n),$ $w(n)\ is\ a\ random\ sequence\ uniformly\ distributed\ betwen\ [1,-1]$	$0 \leq n \leq 100$
$x_5(n) = \{\ldots,1,2,3,\mathbf{2},1,2,3,2,\ldots\}$ $(repeating\ sequence)]$	$0 \leq n \leq 20$

C. SAMPLING

Sampling is done by periodically obtaining samples from a continuous time signal. The period also known as the sampling period is the reciprocal of sampling frequency. Using up-sampling and down-sampling, information can be added or removed from a discrete time signal.

1. Load **signal1.wav** file in your workspace;
2. Using **[y,fs] = audioread ( )**, import the audio and sampling rate information in your workspace.
3. Using **upsample ( )** and **downsample ( )**, create the following signals:

Signal	Resampling
y1	y, up-sampled by 2
y2	y, down-sampled by 2
y3	y2, up-sampled by 2
y4	y3, up-sampled by 2

4. Is **y3** the same as **y**? Is **y4** the same as **y1?**

D. ALIASING

The following exercise investigates the effect of improper sampling.

1. Generate two 1 kHz sine signals (2 seconds duration), first signal at 8 kHz sample frequency and second signal at 1.2 kHz sample frequency;
2. On the same graph, use the plot function to display the two signals versus t in the range $0 \le t \le 5$ msec.;
3. Listen to the two signals one after another using the function **soundsc (x, fs)**; and
4. Compare the two signals. How does the sampling rate affect the digitized sound?

E. QUANTIZATION

Quantization is done by replacing each value of an analog signal x(t) by the value of the nearest quantization level. To exemplify this operation, let's simulate a unipolar ADC (Analog to Digital Converter) having the technical specifications: R = 10 Volts (full-scale range) and B = 3 (number of bits).

1. Write a MATLAB function **y = adc_uni (x, R, B)** where x and y are vectors containing the input signal and the quantized signal, respectively;
2. Test your function with an input ramp signal ranging from -5 to 15 Volts (1 volt per step); and
3. On the same graph, use the plot and stem functions to display the input signal and quantized signal, respectively.

UNIVERSITY OF THE PHILIPPINES DILIMAN
ELECTRICAL AND ELECTRONICS ENGINEERING INSTITUTE

EE 274 / COE 197E: Discrete Time Signals & Systems (DSP1)
Programming Exercise 01
DEADLINE: 02-OCT-2020

F. AUDIO FILE FORMATS

The following exercise will demonstrate the effects of using quantization and sampling on audio signals.

1. Load **music1.flac** provided in UVLe folder. Can also be downloaded here;
2. Using the MATLAB functions you have created in parts A-E, quantize, up/downsample using the following configurations:

Signal	Unipolar-ADC Quantization	Sampling Rate
x1	16-bits	48000
x2	8-bits	48000
x3	4-bits	48000
x4	16-bits	16000
x5	8-bits	16000
x6	4-bits	16000
x7	16-bits	8000
x8	8-bits	8000
x9	4-bits	8000

3. Listen to the generated signals using **soundsc ( )** and compare with the original
4. Compute the SQNR of the signals with respect to the original file. Comment on the results
5. Which has more audible effect: using half the sampling rate or using half bit resolution?

References:

https://www.mathworks.com/matlabcentral/fileexchange/54530-lab-1-digital-signal-processing-sampling-and-quantization