Sabancı University
Faculty of Engineering and Natural Sciences

EE 312 - DISCRETE SIGNALS & SYSTEMS

Laboratory Assignment No. 1

Spring 2020 - 2021

Issued: Wednesday, 10 March 2021 **Due:** Friday, 19 March 2021, 11:59 pm

- After each lab session, you are required to submit a lab report within a week. Please submit a **single** zip file through SUCourse containing the following:
  - A **single** pdf file containing your solutions to the problems, explanations and comments on the results, i.e., your report
  - All of the m files you have written and used for each of the problems
  - All of the images and plots you have generated

  Please test the submission process beforehand so you do not have problems at the last minute. (You can submit a preliminary version and update it later).
  Hint: You can write your report in a Word file and then convert this file to pdf. You can use PDFCreator to convert Word files to pdf. You can download this program from `http://www.pdfforge.org/pdfcreator`.

- Attendance to the laboratory sessions is mandatory. The reports of those who did not attend to a laboratory session without a valid and documented excuse will be evaluated out of 50. Submissions within three days after the 1-week-deadline will be accepted with 10% penalty. No late submission beyond this.

In this laboratory assignment, you are going to work with basic speech and image signals. You will study simple manipulation of these signals on MATLAB.
There are two sound files, i.e., `bird.wav` and `gong.wav`, provided to you. You can load them into MATLAB using the `audioread` command.
Here, you are given some example codes for some of the operations you are expected to perform. Example codes can be your guide to implement other operations on MATLAB.

Problem 1.1 Load the sound file `bird.wav` into a variable $x[n],\ 0 \leq n \leq N-1$ on MATLAB and perform the following operations:

(a) $y[n] = x[N-1-n],\ 0 \leq n \leq N-1$

```
close all;clear all;clc;
[x,fs]=audioread('bird.wav');

% Read the wav file named 'bird.wav' and set the sampling rate

y=x(end:-1:1);
```

```
% note: no need for \for" loop

sound(x,fs)

% Play the original sound, you can use sound command pause;
% Wait for you to press any key

sound(y,fs);% Play the sound that you processed

% What does this process do?
```

(b) $y[n] = x[n] + 0.5x[n-1000]$

```
close all;clear all;clc;
[x,fs]=audioread('bird.wav');

% Read the wav file N=length(x);

prm=1000; % Parameter for the process
x=x(:)'; % make x row vector
y = [x zeros(1,prm)] + 0.5*[zeros(1,prm) x];

% or do the long way (a little different then above,
% what is different?)

y2=x;
for i=prm+1:N
y2(i)=y2(i)+0.5*x(i-prm);
end

sound(x,fs);
pause;
sound(y,fs);
pause;
sound(y2,fs);

% What does this process do?
```

(c) $y[n] = x[n] + 0.5 \cdot \sum_K x[n-K],\ K = 1000, 2000, \ldots, 6000$ that is a multiple delayed echo with linear variation. Write a MATLAB code to perform this operation using a single `for` loop for K.

(d) Repeat part (c) using the `filter` and `conv` commands on MATLAB. Consider $x[n]$ as the input to an LTI system and $y[n]$ as the output. First find and form the impulse response of this system $h[n]$, then filter/convolve the input x with h. How do the `filter` and `conv` command outputs differ? Do they produce results that have the same length? Check `help filter` and `help conv`.

(e) $y[n] = x[2n]$. Try to write a MATLAB code without a `for` loop for this operation. What does this process do?

(f) $y[n] = \begin{cases} x[n/2], & \text{n even} \\ 0, & \text{n odd} \end{cases}$
What does this process do?

For each case, listen to the output and comment.

Problem 1.2 Perform the following operations between the two sound files, i.e., `bird.wav` and `gong.wav`, ($x[n]$, $z[n]$) provided.

(a) $y[n] = x[n] + z[n]$

(b) $y[n] = x[n] - z[n]$

Be careful about the sizes of the sound files. Listen to $x[n]$, $z[n]$ and $y[n]$ for both cases and comment.

Problem 1.3 Write `load penny` into command window of MATLAB to load 128x128 penny image P into memory. You can view it using the function `imshow(P,gray(256))`. The second argument is the `colormap`. Write `help colormap` to learn more.
Now perform the following operations on this image, i.e., $P[n_1, n_2],\ 0 \leq n_1 =\leq N_1 - 1$ and $0 \leq n_2 =\leq N_2 - 1$.

(a) $y[n_1, n_2] = P[N_1 - 1 - n_1, N_2 - 1 - n_2]$

```
close all;clear all;clc;

load penny;
[h,w]=size(P);

% one trick is to use 0-based indices but add 1 (later) to
% each index
for i=0:h-1
    for j=0:w-1
        y(i+1,j+1)=P(h-i-1,w-j-1);
    end
end

%%%%% Shorter Way %%%%%%%
% y=P(end:-1:1,end:-1:1);
%%%%%%%%%%%%%%%%%%%%%%%%%

imshow(P,gray(256));
figure;imshow(y,gray(256));
```

(b) $y[n_1, n_2] = P[n_1, n_2] + P[n_1 - 10, n_2 - 10]$

(c) $y[n_1, n_2] = P[n_1, n_2] + \sum_K P(n_1 - K, n_2 - K),\ K = 10, 20, \ldots, 100$

(d) $y[n_1, n_2] = P[2n_1, 2n_2]$

(e) $y[n_1, n_2] = \begin{cases} P[n_1/2, n_2/2], & n_1, n_2 \text{ even} \\ 0, & \text{otherwise} \end{cases}$

(f) $y[n_1, n_2] = 0.5 * (P[n_1, n_2] + Q[n_1, n_2])$ where $Q[n_1, n_2]$ is the image obtained in part (a).

For each case, explain the performed operations and comment.

Problem 1.4 Write a MATLAB script to simulate a transmission scenario as follows: For a given input signal $x[n]$, obtain the output $y[n]$ by delaying $x[n]$ by d samples and adding random noise to it using `randn` command. The delay can be simulated by putting zeros in front of $x[n]$. Find the cross correlation of the signals x and y using `xcorr` and extract the location of the peak using `max` to estimate delay amount $\hat{d}$. (*Hint:* Write `help max` into the command window to see how to use the `max` command to locate the maximum value in a vector.)

The input to the program should be the signal x, the delay d and the noise standard deviation σ. The output should be the estimated delay value $\hat{d}$. The program should also plot signals $x[n]$ and $y[n]$ and the cross correlation function $r_{xy}[n]$ in the same figure (using `subplot` command). Assume $x[n]$ is a rectangular pulse of length 100 samples. Try to the increase the noise standard deviation σ such that the method starts to fail. Try 5 times for each σ value, since the result depends on the noise signal which will change each time. Summarize your observations.

```
close all;clear all;clc;

d = ...; % define delay amount in samples
sigma =...; % define noise standard deviation
x = ...; % define x, a rectangular pulse of length 100 samples
y = [zeros(1,d) x]; % form noiseless y

% form the noise signal same size as y

noisey = sigma * randn(size(y));
y = y + noisey; % form noisy y
rxy = xcorr(x,y);

% continue from here to plot x, y and rxy

figure;
subplot(311);
plot(x);
...
...
...
```

```
% and try to estimate the delay from rxy

[val, ind] = max(rxy); % ind is the index of the maximum value

% now convert ind to the estimated delay amount dhat

% use \help xcorr" to understand xcorr output

...
...
...

% Compare the original delay value d to the estimated value dhat.
% Are they the same? Change the noise level (sigma value) and obtain
% cases where the delay is not the same as the estimated delay.
```