# Lab 3 - Resampling, Reconstruction and the DFT

| **Section** | L01 | L02 | L03 | L04 |
|---|---|---|---|---|
| **Lab Date** | Oct. 30 | Nov. 6 | Oct. 31 | Nov. 7 |
| **Due Date for Report** | Nov. 12 | Nov. 12 | Nov. 12 | Nov. 12 |

**Assessment:** 5% of the total course mark.

---

Objectives:

- To gain experience in resampling, reconstructing and analyzing digital signals within `MATLAB`. In particular, you will look at resampling and reconstruction of 2-dimensional images, and spectral analysis using the discrete Fourier transform (DFT).
- To gain experience in spectral analysis of speech signals using the fast Fourier transform (FFT) on the TMS 320 DSP processor.

Assessment:

- Your grade for this lab will be based on your ability to create and work with digital signals within `MATLAB` and the TMS 320 DSP processor, and on your reporting of the results.
- Clearly label all plots and their axes (points for style will be deducted otherwise).
- Please attend the lab section to which you have been assigned.
- You can complete this lab with one lab partner.
- By the end of the lab session, you must demonstrate to your TA the `MATLAB` code and the speech recognition results on `MATLAB` and the TMS 320 DSP processor.
- Each pair of students should complete one lab report together. The source code and the report have to be submitted by **11:59 pm on Nov. 12, 2023**. One of the group members can submit the source code and the report.

Pre-lab:

- Carefully read through this lab description so that you know what is required.
- Read through the lecture notes so that you know how to answer the questions.
- Familiarize yourself with the `MATLAB` commands that may be required for this lab – see the list at the end of this lab description for some hints.

Experiments:

1. **Images, resampling and reconstruction**
   So far in this course, we have primarily been considering discrete-time signals and systems. However, many of the same principles apply to discrete-space signals and systems and can consequently be applied to images.

   The 2D image you will be using in this lab is a 256-level gray-scale portable networks graphic (.png) file - `KillarneyPic.png`

(a) Import `KillarneyPic.png`, and report its size and bytes.

(b) Most `MATLAB` functions require arrays to be of the `double` class. Convert the imported picture into a double-class variable and display it in a `MATLAB` figure using the `imshow()` function.

(c) Set the colormap to `gray`. Next, try a colormap of (`1-gray`). What effect does this have on the image? Now return the colormap to `gray`.

(d) You will be using this image to explore different resampling and reconstruction effects. Produce new images by resampling, and in some cases reconstructing, the original as follows:

i. Impulse sampling at $\frac{1}{5}$ of the original rate (i.e., set 4 out of every 5 samples to zero).

ii. Downsampling by a factor of 5 (i.e., discard 4 out of every 5 samples).

iii. Zero-order hold reconstruction (back to the original rate) from the downsampled image created in part (ii) above.

iv. First-order hold reconstruction (linear interpolation—back to the original rate) from the downsampled image created in part (ii) above.

You **cannot** use MATLAB builtin functions to do any of the above operations.
This resampling and reconstruction must be done both horizontally and vertically. Since the X and Y directions are ORTHOGONAL, you can resample and reconstruct in each direction INDEPENDENTLY. Ensure that you have the expected number of rows and columns after each operation.

(e) Of the 4 resampled/reconstructed images plotted in gray-scale, which one most closely resembles the original picture? Rank the images in order of FIDELITY compared to the original.

2. **Introduction to the DFT**

The discrete Fourier transform (DFT) is <u>not</u> the same as the discrete-time Fourier transform from Lab 2. The main difference is in the resulting frequency resolution. When the DFT is computed for an array of length N, the frequency resolution is inherently determined. In contrast, the result of the DTFT has continuous or infinite frequency resolution and thus requires that frequencies be specified for the purpose of computation and plotting in `MATLAB`.

(a) Create a `MATLAB` function for computing the DFT of an arbitrary length `N` input array `x`, for the set of frequency sampling points `k = 0:length(x)-1`.
Hint: Either update one or two lines in your DTFT function to compute the DFT, or try writing a new function for the DFT that implements the matrix formulation of the DFT.

(b) Use your function to compute the DFTs of the 5 rectangular signals of different lengths `N`:

$$x[n] = \begin{cases} 1 & 0 \leq n \leq 15 \\ 0 & \text{otherwise} \end{cases}$$

with `N = 16,32,64,128,256`.

(c) Describe what happens to the magnitude of the output as the zero-padding at the tail end of the rectangular signal increases in length.

(d) Use the function that you created in Lab 2 to compute and plot the DTFT of the same input set. What frequency vector `w` do you need to provide as an input to the DTFT function to return the same outputs as the DFT?

(e) Use the built-in `MATLAB` command for the Fast Fourier Transform `fft()` to compute the DFT for the input set. Compare the results of these three Fourier transform methods.

3. **FFT based speech recognition**
The objective is to write a simple speech recognition program that can distinguish between the two words "yes" and "no".

(a) Load "yes.wav" and "no.wav" files. Plot the magnitude of the FFT (use the built-in `MATLAB` command `fft()`) of these two signals on the same plot. Do you observe any feature that can be used to distinguish both signals?

(b) Use the following code to see if you can come up with a value for **"threshold"** that can distinguish between "yes" and "no" from the provided audio files or your voice.

```
recording = false; % true or false

if recording == true % record audio data from a microphone
    fs = 16000;
    nBits = 16;
    nChannels = 1;
    ID = -1; % default audio input device
    recObj = audiorecorder(fs,nBits,nChannels,ID);
    disp('Start speaking. Recording ends in 3 seconds.')
    recordblocking(recObj,3);
    disp('End of Recording.');
    play(recObj);
    y = getaudiodata(recObj);
else
    filename = 'no.wav'; % change filename to test provided 'yes' and 'no'
    [y, fs] = audioread(filename);
    soundsc(y, fs);
end

N = length(y);
k1 = round(N/4); % FFT component corresponding to fs/4 Hz
k2 = round(N/2); % FFT component corresponding to fs/2 Hz

X = abs(fft(y));
f = sum(X(1:k1))/sum(X(k1+1:k2));

threshold = 0; % Set a value to distinguish words 'yes' and 'no'
if f < threshold
    disp('yes');
else
    disp('no');
end
```

(c) After testing the above program on `MATLAB`, use the provided "SpeechRecognition" program to demonstrate this simple speech recognition on the TMS 320 DSP processor. The threshold value in the provided sample program is not optimal, so your job is to set an

appropriate value that is close to optimal. When testing "yes" and "no" from the provided files, the ADC input should be configured as line, and when you test "yes" and "no" from your voice, the ADC input should be configured as mic. If the input signal is "yes", LED 0 should be on and LED 1 should be off; if the input signal is "no", LED 0 should be off and LED 1 should be on; if there is no input, both LED 0 and LED 1 should be off. Check the code for details.

REPORT: The report should contain

- Any mathematical calculations or derivations carried out
- `MATLAB` plots of results with brief descriptions
- Answers to questions
  - For part 3, you need to discuss the observations.

You **do not** need to include the `MATLAB` code in the report. However, you have to submit the `MATLAB` code separately.

POTENTIALLY USEFUL `MATLAB` COMMANDS:

Note that this is not an exhaustive list! You are not required to incorporate all of these in your scripts.

| | | | | |
|---|---|---|---|---|
| help topic | helpwin | figure | plot | stem |
| histogram | subplot | hold on | xlabel | ylabel |
| legend | title | function | clear | close |
| clc | zeros | ones | cos | exp |
| abs | round | max | min | find |
| if | for | end | real | imag |
| angle | unwrap | phase | audioinfo | audioread |
| audiowrite | soundsc | | | |