Cairo University- Faculty of Engineering
Electronics and Communications Engineering Department
Analysis of Continuous-Time Signals (ELC 2030) – Fall 2023

Signals Project

It is required to use Matlab, or any other programing tool, to perform the following tasks:

I. Image filtering and restoration

In this task, we will explore fundamental image processing operations, such as filtering in the spatial (space) domain using 2D convolution and basic image restoration in the frequency domain.

Background: In 2D convolution, the system (often referred to as filter or kernel) is flipped about its origin in both x and y directions. Then it is shifted across the input image, and corresponding pixel values are multiplied and summed to give the output value. This is repeated for all output values. Through the selection of different kernels, various operations can be performed.

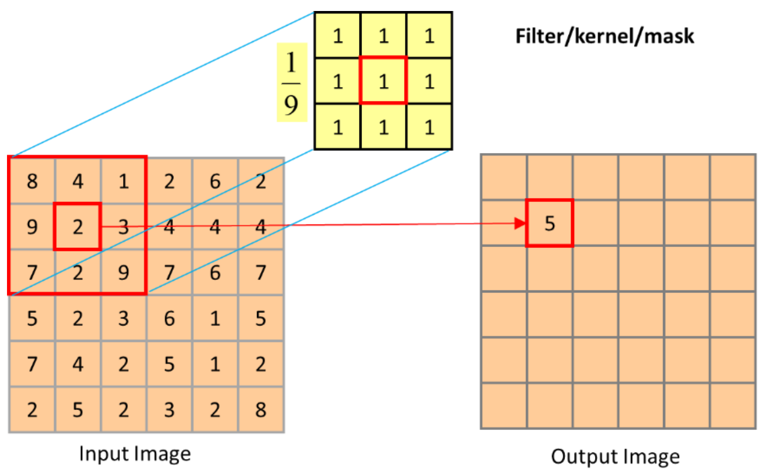

Fig. 1: 2D convolution

Task:

A. Read the image file ‘peppers.png’. Extract and display each of its three color components.

B. It is required to perform the following operations on the given image:
1. Edge detection: detect and display the edges in the image.
2. Image sharpening: enhance the image details and edges.
3. Blurring (averaging): hide the fine details in the image.
4. Motion blurring: simulate motion blur in the horizontal direction.

For each of the above four operations, do the following:

- Select an appropriate kernel to perform the operation.
- Perform 2D convolution between the input image and the kernel. Display the resulting image.

- Justify your choice of the kernel. Describe how the chosen kernel achieves the required operation.

Save the resulting images of the four operations as 'image1.png', …, 'image4.png', respectively.

C. Attempt to restore the original image from the motion-blurred image and the kernel in part B.4, using frequency domain. Note that convolution in the space domain corresponds to multiplication in the frequency domain [Lecture 10]. Save the restored image as 'image5.png'.

Useful Matlab functions: conv2, fft2, ifft2.

II. Communication system simulation

In this task, we will simulate a communication system based on amplitude modulation and frequency-division multiplexing. We will explore recording and filtering audio, as well as displaying its spectrum.

Background:

The spectrum of a discrete-time signal (the DTFT) is generally a continuous function of frequency. Therefore, it is not suitable for numerical computations. Instead, we will use the **Discrete Fourier Transform (DFT)**, which is a sampled version of the DTFT [Project lecture].

When a continuous-time signal is sampled at a rate f_s Hz, its spectrum is repeated at multiples of f_s Hz. The N-point DFT of a DT signal $x[n]$ gives N samples of one period of its spectrum. Since we have N samples in a range of frequencies from 0 to f_s Hz, the spacing between the samples is f_s/N. The N-point DFT X[k] is given in the range of $k = 0\ to\ N - 1$ and corresponds to frequencies $f = 0\ to\ (N - 1)f_s/N$. Note that N must be greater than or equal to L, the length of x[n].

We say that $X[k]$ is the DFT of $x[n]$ and $x[n]$ is the inverse DFT (IDFT) of $X[k]$. These two operations are implemented in Matlab using 'fft' and 'ifft' functions, respectively. Fast Fourier transform (FFT) is an efficient algorithm for computing the DFT.

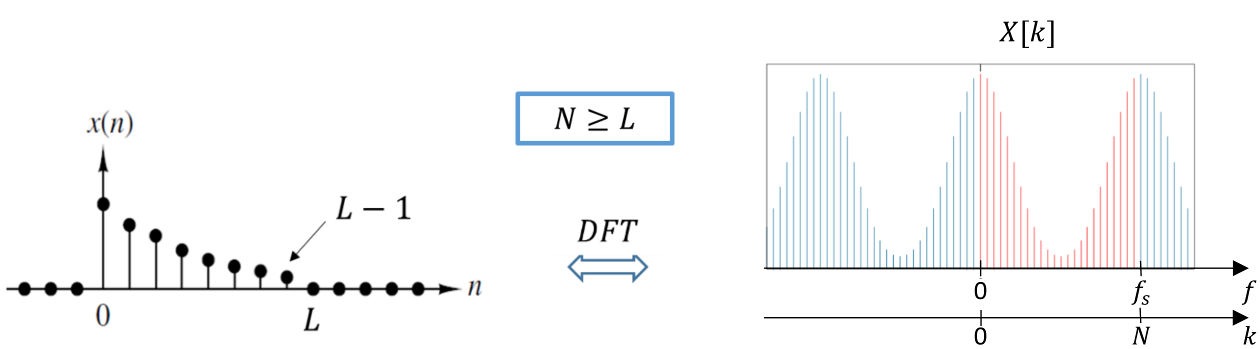

Fig. 2: DFT

Task:

a. Using Matlab, record two segments of your voice of about 10 seconds each. Select appropriate values for sampling frequency and bit depth. Justify your choice of such values. Save the audio files as ‘input1.wav’ and ‘input2.wav’.
b. Limit the maximum frequency of both signals to a suitable value. Design a LPF filter to filter each of the signals. You should test different cutoff frequencies, listen to the filtered audio, and select an appropriate value such that the quality of audio is not significantly affected. Plot the frequency response of the filter [Lecture 10].
c. Plot the magnitude spectrum of both signals before and after filtering against the frequency in Hz using FFT [Project lecture]. Use the function ‘fftshift’ to make the zero frequency in the center of the plot.
d. Perform amplitude modulation on both signals in a frequency-division multiplexing system, using suitable carrier frequencies [Lecture 10]. Justify your choice of carrier frequencies. Plot the magnitude spectrum of the transmitted signal.
e. Design a receiver to obtain each of the two signals from the transmitted signal. Save the audio files as ‘output1.wav’ and ‘output2.wav’. Plot the magnitude spectrum of each. Draw a block diagram of both transmitter and receiver. Explain the operation of the receiver with equations both in time domain and in frequency domain.

Hints:

1. Pay careful attention to the sampling theorem when considering the sampling rate in all stages. [Lecture 11]
2. You may use the Matlab filter designer tool “filterDesigner” to design the filter and export it as an object. Then use the “filter” function to filter the audio using the designed filter. If used, add a screenshot of the design page to the report. FilterDesigner can also be used to generate the response plots of the filter [Project lecture].

Useful Matlab functions: fft, fftshift, filter

Deliverables:

1. An **uncompressed pdf** project report containing:
   a. Explanation of your work.
   b. All the required results and answers to questions.
   c. All the required figures. Label your figures properly.
   d. All the codes, included at the end.
2. All the code, audio, and images files as uncompressed separate files.

Instructions:

- You can work in teams of up to 2 members per team.
- Any copied results or codes will result in zero grade for both teams.
- Code in the report should be supplied as text, not as screenshots.
- It is not allowed to use any form of Artificial Intelligence in this project.

Due date: December 30, 2023, at 11:59 pm.