

BILKENT UNIVERSITY  
ELECTRICAL AND ELECTRONICS ENGINEERING DEPARTMENT  
Fall 2025–26

---

## EEE - 321: Signals and Systems Lab Assignment 4

---

Please carefully study this assignment before coming to the laboratory. You may begin working on it or even complete it if you wish, but you do not have to. There will be a short quiz in the lab session to test your understanding of the content of the assignment. Within one week, complete the assignment in the form of a report and upload it to Moodle. Some of the exercises will be performed by hand and others by using MATLAB. What you should include in your report is indicated within the exercises. After you upload your report, there will be another quiz to test your understanding of what you did in the report, given in the next lab session.

---

**Note:** Along with this pdf file, you will find a .rar file containing several bmp pictures. Unzip the archive and place all the images contained in it under the current directory of Matlab.

---

### Part 1 - Preparation

In this lab, you are going to do a bit of image processing. You will perform some simple operations on gray scale (white and black) digital images and see the results.

In this course up to now, we have mainly dealt with one-dimensional signals such as  $x(t)$  or  $x[n]$ . These signals are said to be one-dimensional because they are functions of one independent variable ( $t$  or  $n$ ).

Images, on the other hand are signals of two independent variables. Therefore they are two-dimensional signals.

A gray scale digital image can be represented with a 2D discrete function  $x[m, n]$  such that  $x[m, n]$  denotes the light intensity of the  $(m, n)$ th pixel of the image ( $m$  and  $n$  are integers). Since practical images are of finite size (such as  $512 \times 512$ ,  $1024 \times 1024$  etc.), we usually have

$$x[m, n] = 0 \quad \text{for} \quad m \notin [0, M_x - 1] \quad \text{or} \quad n \notin [0, N_x - 1]$$

where the image size is  $M_x \times N_x$ .

Recall that we store 1D signals in 1D arrays in Matlab. Since images are 2D signals, they are stored in matrices in Matlab. An  $M_x \times N_x$  image is stored in an  $M_x \times N_x$  matrix in Matlab.

In this first part, you will just practice reading images in Matlab and displaying them. Place the image named **Part1.bmp** under the current directory of Matlab. Now issue the command:

$$\mathbf{A} = \text{double}(\text{rgb2gray}(\text{imread}('Part1.bmp')));$$

You will see that a  $527 \times 791$  matrix of type double will be created and stored in the workspace. This matrix contains the image. The original image is a color image, but it is converted to a gray scale image by the Matlab function **rgb2gray**. What do the elements of matrix **A** represent?

Next type in the following commands:

$$\mathbf{MAX} = \max(\max(\mathbf{A}));$$

$$\mathbf{MIN} = \min(\min(\mathbf{A}));$$

What do MAX and MIN represent?

Next, issue the following commands:

$$\mathbf{A} = \mathbf{A} - \mathbf{MIN}; \quad \mathbf{A} = \mathbf{A}/\mathbf{MAX}; \quad \mathbf{A} = \mathbf{A} - 0.5;$$

Explain the effect of each command on matrix **A**.

Now display the image **A** using the following command:

$$\text{figure; imshow}(\mathbf{A}, [ \ ]);$$

A new figure window will open and the image will be displayed. What is the importance of an empty matrix,  $[ \ ]$  in this code? When is it needed and when is it not?

During the rest of this assignment, you will use these commands to read and display various images.

For this part, just make sure that you see the image without any problem. Do not provide the printout of the image or anything for the report.

## Part 2 - 2D Convolution

Recall that 1D discrete time (DT) systems map a 1D input signal  $x[n]$  to another 1D signal  $y[n]$  (that we name the output signal). (We say discrete time since the independent variable usually denotes time.)

Similarly, a 2D discrete space (DS) system maps a 2D input signal  $x[m, n]$  to a 2D output signal  $y[m, n]$ . (Now we say discrete space since the independent variables usually denote the space coordinates.) Recall that a 1D DT system is called linear time invariant (LTI) if it satisfies the following two properties:

- $\alpha_1 x_1[n] + \alpha_2 x_2[n]$  produces  $\alpha_1 y_1[n] + \alpha_2 y_2[n]$  for all  $x_1[n], x_2[n], \alpha_1, \alpha_2$ .
- $x[n - n_0]$  produces  $y[n - n_0]$  for all  $x[n]$  and  $n_0$ .

Similarly, a 2D DS system is called linear space invariant (LSI) if it satisfies:

- $\alpha_1 x_1[m, n] + \alpha_2 x_2[m, n]$  produces  $\alpha_1 y_1[m, n] + \alpha_2 y_2[m, n]$  for all  $x_1[m, n], x_2[m, n], \alpha_1, \alpha_2$ .
- $x[m - m_0, n - n_0]$  produces  $y[m - m_0, n - n_0]$  for all  $x[m, n]$  and  $(m_0, n_0)$ .

Now let us develop the input-output representation of 2D DS LSI systems by forming an analogy with 1D DT LTI systems.

Recall that a 1D discrete impulse signal is defined as:

$$\delta[n] = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{otherwise} \end{cases}$$

While developing the input-output relation for a 1D DT LTI system, we first wrote the input signal  $x[n]$  as a superposition of shifted impulse signals as:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k] \delta[n - k]$$

where we interpreted  $x[k]$  as the coefficient of the impulse shifted by  $k$  units (that is,  $x[k]$  is the coefficient of  $\delta[n - k]$ ). Then, we named the response that the system gives to  $\delta[n]$  as  $h[n]$  (impulse response). Using the time invariance property of the system, we recognized that the response of the system to  $\delta[n - k]$  should be  $h[n - k]$ . Then, using the linearity property of the system, we wrote the input-output relation of the 1D DT LTI system as:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] h[n - k]$$

We named the above operation as the convolution of  $x[n]$  and  $h[n]$  and used the shorthand notation:

$$y[n] = x[n] * h[n]$$

to denote it.

Now, define the two dimensional discrete impulse signal as:

$$\delta[m, n] = \begin{cases} 1 & \text{if } m = 0, n = 0 \\ 0 & \text{otherwise} \end{cases}$$

and going through the same steps, derive the input-output relation of a 2D DS LSI system, or derive the formula for 2D convolution of  $x[m, n]$  and  $h[m, n]$  that we denote as:  $x[m, n] * h[m, n]$ . In other words, derive the 2D version of Eq. 2. Show—explain your steps.

$$\begin{aligned} y[m, n] &= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x[k, l] h[m - k, n - l] \\ &= x[m, n] * h[m, n] \\ &= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} h[k, l] x[m - k, n - l] \end{aligned}$$

## Part 3 - 2D LSI Systems

Recall that a 1D DT LTI system is called FIR if the impulse response  $h[n]$  contains a finite number of nonzero values such that  $h[n] = 0$  for  $n \notin [0, M_h - 1]$  where  $M_h \in \mathcal{Z}^+$ . Similarly, a 2D DS LSI system is called FIR if the impulse response  $h[m, n]$  contains a finite number of nonzero values such that  $h[m, n] = 0$  for  $m \notin [0, M_h - 1]$  and  $n \notin [0, N_h - 1]$  where  $M_h, N_h \in \mathcal{Z}^+$ .

In this part, you will write a Matlab function that computes the output when a finite sized input image  $x[m, n]$  (of size  $M_x \times N_x$ ) is input to a 2D FIR DS LSI system whose impulse response  $h[m, n]$  is of size  $M_h \times N_h$ . From another perspective, your code will compute the 2D convolution of two finite-size 2D signals  $x[m, n]$  and  $h[m, n]$ . We assume that

- $x[m, n]$  can only be nonzero within  $0 \leq m \leq M_x - 1$  and  $0 \leq n \leq N_x - 1$ .
- $h[m, n]$  can only be nonzero within  $0 \leq m \leq M_h - 1$  and  $0 \leq n \leq N_h - 1$ .

Under these conditions Eq. 4 reduces to:

$$y[m, n] = \sum_{k=0}^{M_h-1} \sum_{l=0}^{N_h-1} h[k, l] x[m - k, n - l]$$

Based on the above equation, show that  $y[m, n]$  can only be nonzero within  $0 \leq m \leq M_y - 1$  and  $0 \leq n \leq N_y - 1$ . Determine  $M_y$  and  $N_y$  in terms of  $M_x, N_x, M_h$  and  $N_h$ . Include your work to your report.

Next, write a Matlab function of the following form

**function** [y] = DSLSI2D(h, x) where

- **h** of size  $M_h \times N_h$  denotes the impulse response of the system, such that  $\mathbf{h}(1, 1) = h[0, 0]$ ,  $\mathbf{h}(1, 2) = h[0, 1]$ ,  $\dots$ ,  $\mathbf{h}(k, l) = h[k - 1, l - 1]$ .
- **x** of size  $M_x \times N_x$  denotes the input signal  $x[m, n]$ , such that  $\mathbf{x}(1, 1) = x[0, 0]$ ,  $\mathbf{x}(1, 2) = x[0, 1]$ ,  $\dots$ ,  $\mathbf{x}(k, l) = x[k - 1, l - 1]$ .
- **y** of size  $M_y \times N_y$  denotes the output signal, such that  $\mathbf{y}(1, 1) = y[0, 0]$ ,  $\mathbf{y}(1, 2) = y[0, 1]$ ,  $\dots$ ,  $\mathbf{y}(k, l) = y[k - 1, l - 1]$ .

**Note:** Do not use any built in command of Matlab. Directly implement the convolution formula.

Check your function: If you take

$$x = \begin{bmatrix} 1 & 0 & 2 \\ -1 & 3 & 1 \\ -2 & 4 & 0 \end{bmatrix}$$

and

$$h = \begin{bmatrix} 1 & -1 \\ 0 & 2 \end{bmatrix}$$

you should get

$$y = \begin{bmatrix} 1 & -1 & 2 & -2 \\ -1 & 6 & -2 & 3 \\ -2 & 4 & 2 & 2 \\ 0 & -4 & 8 & 0 \end{bmatrix}$$

Include your code to your report.

## Part 4 - Image Denoising

Now it is time to see some practical image processing examples. Read the picture named Part4.bmp in a matrix named **x** and display it. You will see a picture that is highly corrupted by noise. Noisy images may occur due to several reasons. For example, during capturing the image by a CCD camera, each CCD pixel may insert noise. Or, if the image is transmitted through a noisy communication channel, the received image can be corrupted by noise. In this part, you will try to rescue this image from the noise without disturbing the image itself as much as possible.

Suppose we know that the noise on the picture has high frequency content. We also know that typical daily life images taken by a typical camera have low-frequency content. Why should one apply a low pass filter to the noisy image and try to eliminate the noise? A typically used 2D FIR low pass filter has the following impulse response:  $h[m, n] = \begin{cases} 0 & \text{if } m < 0 \text{ or } n < 0 \text{ or } m > M_h - 1 \text{ or } n > N_h - 1 \\ \text{sinc} \left\{ B \left( m - \frac{M_h-1}{2} \right) \right\} \text{sinc} \left\{ B \left( n - \frac{N_h-1}{2} \right) \right\} & \text{otherwise} \end{cases}$  where  $B$  is a free parameter that determines the bandwidth of the filter. We should select  $B$  between 0 and 1 .

Let  $D$  denote your ID number, and let  $D_7$  denote your ID number in modulo 7 . That is

$$D \equiv D_7 \pmod{7}$$

with  $0 \leq D_7 \leq 6$ . You can compute  $D_7$  using the rem command of Matlab. To learn how rem works, type help rem in Matlab command window.

Now, let  $M_h = N_h = 20 + D_7$  and  $B = 0.7$ . Prepare the **h** matrix that represents  $h[m, n]$ . If you wish, you can use nested for loops over  $m$  and  $n$ , this time it is allowed. Recall that Matlab already has a built in function named sinc to compute the values of sinc(.) function. Include your code for preparing  $h$  to your report.

Using the code that you developed in Part 3, process the noisy image with this filter. Display the output image. Repeat the exercise with  $B = 0.4$  and  $B = 0.1$  as well. Include all the output images to your report. Use subplot command, and show the images on the same matlab figure. Which  $B$  value gives the best performance in terms of the image denoising? What is the drawback of choosing large or small value for  $B$  ?

Now try different combinations of  $B$  and  $M_h$ . By examining the output images, discuss the subject from different views, such as, denoising performance, excessive amount of smoothing, border effects etc. Put only the output image which shows the best performance to your report indicating  $B$  and  $M_h$  value for this image.

As you see, it is possible to greatly clarify the information in a corrupted signal using only the simple concepts of convolution, LSI systems, etc. In this part, we tried to remove the noise from a corrupted image. Such problems are called image denoising problems. Many algorithms have been developed by many researchers that try to clear the images from the corrupting noise while giving minimum damage to the original image.

## Part 5 - Edge Detection

Another widely studied interesting problem of image processing is the detection of edges in an image, which is called the edge detection problem. In a typical image, edges are the set of points in the close vicinity of which the pixel values change abruptly. Since changes are sudden and great, edges are inherently associated with high frequencies. Therefore, we can make use of high pass filters to detect edges.

Read the picture named Part5.bmp in a matrix named  $\mathbf{x}$  and display it.

Consider a 2D FIR DS LSI system whose impulse response  $h_1[m, n]$  is equal to:

$$h_1[m, n] = \begin{cases} 0.5 & \text{if } m = 0, n = 0 \\ -0.5 & \text{if } m = 0, n = 1 \\ 0 & \text{otherwise} \end{cases}$$

Prepare  $h_1[m, n]$  and process your image with this filter. Let  $y_1[m, n]$  denote the resulting image. Display the image which is defined as  $s_1[m, n] = y_1^2[m, n]$ . Include this image to your report. Which parts of the original image are emphasized? Include your answer to your report.

Now let

$$h_2[m, n] = \begin{cases} 0.5 & \text{if } m = 0, n = 0 \\ -0.5 & \text{if } m = 1, n = 0 \\ 0 & \text{otherwise} \end{cases}$$

As you see,  $h_2[m, n] = h_1[n, m]$ . Again, prepare  $h_2[m, n]$  and process your image with this filter. Let  $y_2[m, n]$  denote the resulting image. Display the image which is defined as  $s_2[m, n] = y_2^2[m, n]$ . Include this image to your report. Which parts of the original image are emphasized now? Comment on the difference with the image you obtained with  $h_1[m, n]$ . Include your answer and comments to your report.

Finally, process the original image with the filter  $h_3[m, n] = 0.5h_1[m, n] + 0.5h_2[m, n]$ . Let  $y_3[m, n]$  denote the resulting image. Display the image which is defined as  $s_3[m, n] = y_3^2[m, n]$ . Include this image to your report. Which parts of the original image are emphasized now? Comment on the difference with the images you obtained with  $h_1[m, n]$  and  $h_2[m, n]$ . Include your answer and comments to your report.