Assignment 2

General Info

- The work can be done anywhere where MATLAB and the related toolboxes are available.
- A **written report** is required. The report is free form but should include results, figures, and any code you wrote, as well as discussions of what you observe.
- Reports should be submitted as an electronic copy in **PDF** format on **Canvas** before the due date.
- Late submissions will not be accepted.
- Better documentation and clearer discussions of your work improves our ability to fairly mark your report. Make sure your report is well structured, organized, and clear. Your report has to follow the order of questions. Give all relevant code right before the results and discussion of each part, not in an appendix.

1. Properties of Convolution

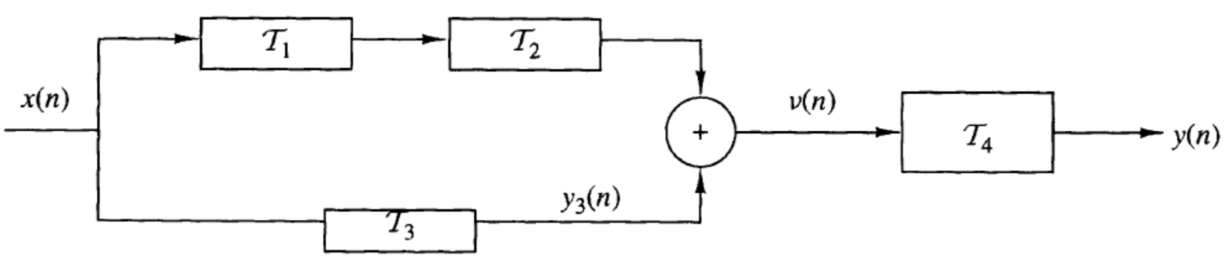
This is a convolution problem in 1D that demonstrates some convolution properties. Compute and plot the overall impulse response h(n) of the system shown in in the figure below for 0<n≤99. T_1, T_2, T_3, and T_4 and described by the following equations:

$$\mathcal{T}_1 : h_1(n) = \{\underset{\uparrow}{1}, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$$

$$\mathcal{T}_2 : h_2(n) = \{\underset{\uparrow}{1}, 1, 1, 1, 1\}$$

$$\mathcal{T}_3 : y_3(n) = \frac{1}{4}x(n) + \frac{1}{2}x(n-1) + \frac{1}{4}x(n-2)$$

$$\mathcal{T}_4 : y(n) = 0.9y(n-1) - 0.81y(n-2) + v(n) + v(n-1)$$



Tips: The following MATLAB commands may be useful: `filter`

2. Finding Periodicity with Autocorrelation

Measurement uncertainty and noise sometimes make it difficult to spot oscillatory behavior in a signal, even if such behavior is expected. The autocorrelation sequence of a periodic signal has the same cyclic characteristics as the signal itself. Thus, autocorrelation can help verify the presence of cycles and determine their durations.

(a) Consider a set of temperature data collected by a thermometer inside an office building that takes 2 measurements every hour. Do you expect it to have periodicities and in what ways? Why or why not?

(b) Load the raw data with

```
load officetemp;
```

Subtract the mean to concentrate on temperature fluctuations. Convert the temperature from Fahrenheit to degrees Celsius. Plot the temperature (^oC) vs. time (days).

(c) Compute the autocorrelation of the temperature. Is the calculated autocorrelation function periodic? Why or why not?

(d) If the signal is periodic, find the periods.

(e) Explain the results in (d) and if they match your expectations stated in (a).

3. Image convolution

Given the two spatial filters below:

$$h_1 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$h_2 = \begin{bmatrix} 0 & +1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

(a) Convolve the two filters h_1 and h_2 so as to create the filter $h_3 = h_1 * h_2$.
(b) What type of filter is h_3 (lowpass, highpass, etc)? Explain your answer.
(c) Load the image ‘mri.tif’, convolve it with h3. Display the original and convolved images and briefly describe what you see.
(d) Repeat (a), (b), and (c) but with the spatial filters $h_1$and h_2 given as:

$$h_1 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$h_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

4. Color Histogram and Edge Detector

(a) Write your own MATLAB function that computes the histogram of a color image:

hist = colorhist(image, n1, n2, n3)

where “hist” is a one-dimensional vector and n1, n2, n3 denote the number of bins for the R, G, B channel, respectively. The algorithm should scan the R channel first, then the G channel and finally the B channel. Then, concatenate the histograms of each color channel into the resulting vector, “hist”.
Perform your “colorhist” function on the images *TheCourtesan.bmp* and *TheHagueSchool.bmp*. Is it possible to discriminate these two images from each other based on their color histograms?

(b) Now you will explore a very simple edge detector:

(i) Load the image “evolution.jpg” and smooth it with a Gaussian filter to eliminate noise.

(ii) Use the *gradient* function to compute the 1st derivatives in the x and y direction:

[dy,dx]=gradient(I);

(iii) Look at the magnitude of the gradient image using *imshow, M=sqrt(dx^2+dy^2).*

(iv) Now use the command *quiver(dy,dx)* to view the actual gradient vectors; you may want to zoom in.

(v) Create a binary edge image by thresholding the gradient magnitude image. Choose an appropriate threshold value that gives you few gaps in the edges.

5. Two-Dimensional Fourier Transform

(a) Construct a binary valued image containing a rectangle (e.g. 60×20 pixels) in the middle. The background should be black and the rectangle should be white. Display the binary image. Display and study the Fourier transform of the image and make sure that you understand the overall relation between the image form and the power spectrum.
Tips: Check how the logarithm operation enables a better visualization (display) of the Fourier spectrum. Also, the function ‘fftshift’ can be useful for centering the spectrum.

(b) What happens in the Fourier domain when the object, i.e. the rectangle in (a), is rotated, translated, or made “thinner” (i.e. size decreased in one direction)? Display the results in both spatial and Fourier domain before/after the operations.

(c) The Fourier transform $X(\omega_1, \omega_2)$ of an image $x(n_1,n_2)$ is in general complex valued and the unique representation of an image in the Fourier transform domain thus requires both the phase and the magnitude of the transform. It is, however, often desirable to synthesize or reconstruct a signal from partial Fourier domain information. The Fourier transform phase $\theta_x(\omega_1, \omega_2)$ alone often captures most of the intelligibility of the original signal (image) $x(n_1,n_2)$. The Fourier transform magnitude contributes less to the image information. For the image lena512.bmp, illustrate the above statement by synthesizing the magnitude-only signal:

$$x_m(n_1,n_2) = F^{-1}[\,|X(\omega_1,\omega_2)|\,e^{j0}]$$

and the phase only signal:

$$x_p(n_1,n_2) = F^{-1}[1e^{j\,\theta x(\omega 1,\ \omega 2)}]$$

Tip: You might need to normalize the results for appropriate display.

6. Laplacian Operator in the Frequency Domain

(a) Show that the Laplacian spatial operator can be implemented in the frequency domain using the filter *$H(u,v)=-4\pi^2(u^2+v^2)$; that is,*

$$\Im[\nabla^2 f(x,y)] = -4\pi^2(u^2+v^2)F(u,v)$$

(b) Read the image blurry_moon.tif and implement the frequency domain filtering process expressed above.
(c) Generate a filter h = [0 1 0; 1 −4 1; 0 1 0]. Use this filter to process the moon image in the spatial domain and compare your results with (a). Comment on your findings.

Tips: The following MATLAB commands may be useful: `fft2`, `ifft2`, `fftshift`, `imresize`, `imrotate`, `imnoise`, `imagesc`, `imshow`.

End of assignment 2