1 Signals and Systems Review

1. Suppose x_1 and x_2 are two finite sequences defined as

$$\begin{aligned} x_1[n] &= [4,2,6,3,8,1,5] \\ x_2[n] &= [3,8,6,9,6,7] \end{aligned}$$

Let the starting index of $x_1[n]$ be -1 (i.e. $x_1[-1] = 4, x_1[0] = 2\ldots$) and the starting index of $x_2[n]$ be -2. Obtain the convolution of $x_1[n]$ and $x_2[n]$ using Matlab

   a) You should first try an implementation on your own without using any inbuilt functions.

   b) Now find out whether there are any inbuilt Matlab functions that can be used to compute convolution and use that.

2. Find out what is meant by auto-correlation of a discrete time signal $x[n]$. Find out the auto-correlation of $x_1[n]$ and $x_2[n]$ defined above using your own code. Find out whether there is a builtin Matlab function for finding out the auto-correlation. Find out the auto-correlation of $x_1[n]$ and $x_2[n]$ using the builtin function(s). Find out whether auto-correlation can be implemented using convolution.

3. Find out what is meant by cross-correlation between two discrete time signals $x_1[n]$ and $x_2[n]$. Find out whether cross-correlation is implemented as a builtin function in Matlab. Find the cross-correlation between $x_1[n]$ and $x_2[n]$ using your own code as well as builtin functions. Find out whether cross-correlation can be implemented using convolution.

4. Define what odd and even signals are. Write Matlab code to find out the odd and even parts of the following signals:

   a) $x_1[n] = [4,3,5,6,7,2]$; starting index is -2

   b) $x_2[n] = sin(2\pi 100n) + cos(\pi 100n)$ for all n (but choose an appropriate finite time extent for implementation)

5. Let $h[n]$ as defined below be the impulse response of a discrete time LTI system

$$h[n] = \begin{cases} n, & \text{for } n \in \{0,1,2,\ldots,10\}, \\ 0, & \text{otherwise.} \end{cases}$$

Using Matlab, plot the frequency response of this system (plot the magnitude and phase spectra separately). Note that the frequency response is the DTFT which is a continuous function of the frequency ω. Therefore, a sufficiently "smooth" discretization of ω is needed for obtaining these plots. Observe whether the frequency responses are periodic with ω or not.

6. Write a Matlab function to plot the frequency response (magnitude and phase) corresponding to any $h[n]$, rather than the specific case above. The function should take $h[n]$ and two variables "dbscale", "actualfreq" as input. The function should produce the magnitude and phase response as output. If the "dbscale" variable is 1 then the magnitude should be plotted in decibels. If the "actualfreq" variable is not -1, then the ω axis should be plotted in actual frequencies by interpreting the value of the "actualfreq" variable as the sampling frequency used to produce the discrete time signal (i.e., the time duration between two samples is $1/actualfreq$). Test this function with example inputs (you are free to choose non-trivial inputs) and demonstrate that your implementation is correct. The inputs should be chosen carefully to test the function.

7. We know that if a discrete time LTI system has an impulse response $h[n]$ and if the input to the system is $x[n]$, then the output (defined as $y[n]$) is given by the discrete time convolution of $x[n]$ and $h[n]$.

   a) Write a Matlab function "discreteTimeConvolve" that takes as inputs the impulse response and the input signal and produces the output signal as output. Please note that this function should not use any inbuilt Matlab functions to do convolution but rather implement convolution using basic array operations.

   b) Now find out whether any inbuilt Matlab function can be used to directly implement convolution. Demonstrate the use of this function using 3 examples.

8. A discrete time LTI system can also be specified by a constant coefficient difference equation as follows:

$$y[n] = \sum_{k=0}^{N} a_k x[n-k] + \sum_{l=1}^{M} b_l y[n-l].$$

   a) Implement a Matlab function "discreteTimeCCDE" that takes as input the set of $N+1$ coefficients (a_k) and M coefficients (b_l) and an input signal $x[n]$ and produces as output the signal $y[n]$. Please note that any initial conditions can be taken as input or assumed. Do not use any inbuilt Matlab functions for this part.

   b) Now find out whether any inbuilt Matlab function can be used to directly implement the above operation. Demonstrate the use of this function using 3 examples. (Hint: we have seen this in class when we studied IIR filters.)

9. In this task, you will explore one of the most important properties of the DTFT. Suppose $h[n]$ is the impulse response of a discrete time LTI system. The LTI system has output $y[n]$ when $x[n]$ is the input. Let $H(\omega), X(\omega)$ and $Y(\omega)$ be the DTFTs of $h[n], x[n]$, and $y[n]$. We know that $Y(\omega) = H(\omega)X(\omega)$.

   a) Write a Matlab function to compute the DTFT of a signal; note that you have already done this in (2) in this labsheet. The DTFT that is being computed is a suitably discretized version of the actual DTFT (defined for the continuous ω).

   b) Using the above function compute $H(\omega)$ and $X(\omega)$ and their product $Y(\omega)$.

   c) Compute $y[n]$ by taking the inverse DTFT (think about how you will implement this on your computer; are there inbuilt Matlab functions that can help you implement this.)

   d) Compare the $y[n]$ that you have obtained with what is obtained via directly convolving $x[n]$ and $h[n]$. Are there any differences? If there are, then why do such differences arise.

10. Review the following from the textbook

    a) discrete Fourier transform (DFT) of a discrete time signal $x[n]$

    b) the inverse DFT

11. Using the internet

    a) https://en.wikipedia.org/wiki/Fast_Fourier_transform,

    b) http://www.dspguide.com/ch12/2.htm,

    or using the textbook by Oppenheim and Schafer (Chapter 9 and also Chapter 8), get an idea about what fast Fourier transform (FFT) is. Then using Matlab's documentation (or internet resources) study what the following inbuilt Matlab functions do. Pay attention to the inputs and outputs of these functions

    a) dftmtx

    b) fft

    c) ifft

d) fftshift

12. Implementation of DFT and IDFT.

    a) Write a Matlab function named "mydft1" that computes the N-point DFT of a discrete time signal $x[n]$ where $n \in \{0, 1, \dots, N-1\}$.

        i. The input $x[n]$ is assumed to extend from 0 to $N-1$ and can be a Matlab vector, and

        ii. the function should implement DFT using loops, and

        iii. the function should return the N-point DFT as another Matlab vector.

    b) Write a Matlab function named "myidft1" that computes the N-point IDFT of a DFT $X[k]$ where $k \in \{0, 1, \dots, N-1\}$.

        i. The input $X[k]$ is assumed to extend from 0 to $N-1$ and can be a Matlab vector, and

        ii. the function should implement IDFT using loops, and

        iii. the function should return the N-point signal $x[n]$ as another Matlab vector.

    c) i. Using mydft1() compute the 16-point DFT of $x[n] = cos(2\pi(0.25)n)$ for $n \in \{0, \dots, 15\}$.

        ii. Test whether myidft1(mydft1($x[n]$)) $= x[n]$.

    d) Write a Matlab function named "mydft2" that computes the N-point DFT of a discrete time signal $x[n]$ where $n \in \{0, 1, \dots, N-1\}$.

        i. The input $x[n]$ is assumed to extend from 0 to $N-1$ and can be a Matlab vector, and

        ii. the function should implement DFT using dftmx,

        iii. the function should return the N-point DFT as another Matlab vector.

    e) Write a Matlab function named "myidft2" that computes the N-point IDFT of a DFT $X[k]$ where $k \in \{0, 1, \dots, N-1\}$.

        i. The input $X[k]$ is assumed to extend from 0 to $N-1$ and can be a Matlab vector, and

        ii. the function should implement IDFT using dftmx, and

        iii. the function should return the N-point signal $x[n]$ as another Matlab vector.

    f) i. Using mydft2() compute the 16-point DFT of $x[n] = cos(2\pi(0.25)n)$ for $n \in \{0, \dots, 15\}$.

        ii. Test whether myidft2(mydft2($x[n]$)) $= x[n]$.

13. Let $x[n] = cos(2\pi(0.25)n)$ for $n \in \{0, \dots, 15\}$.

    a) Compute the 16-point DFT of $x[n]$ using the fft function

    b) Assuming that $x[n]$ was obtained by sampling at a rate of 1 Hz, plot the DFT (magnitude and phase separately) with the frequency axis in the range $[0, 2\pi]$ radians/sec.

    c) Assuming that $x[n]$ was obtained by sampling at a rate of 1 Hz, plot the DFT (magnitude and phase separately) with the frequency axis in the range $[-\pi, \pi]$ radians/sec. Use the fftshift function for this task.

    d) Check if ifft(fft($x[n]$)) is $x[n]$.

14. Suppose x_1 and x_2 are two finite sequences defined as

$$\begin{aligned} x_1[n] &= [4, 2, 6, 3, 8, 1] \\ x_2[n] &= [3, 8, 6, 9, 6, 7] \end{aligned}$$

    Write a Matlab program to compute the circular convolution of $x_1[n]$ and $x_2[n]$. Modify your Matlab program such that it can compute the circular convolution of any x_1 and x_2 given as input - include in your program logic to check whether the circular convolution can be computed and return appropriate error messages.

15. Suppose we have two signals defined as follows:

$$\begin{aligned} x[n] &= \begin{cases} (0,1,2,3,4) \text{ for } n \in \{0,1,2,3,4\}\,, \text{ and,} \\ 0 \text{ otherwise} \end{cases} \\ h[n] &= \begin{cases} (1,1,1,1,1) \text{ for } n \in \{0,1,2,3,4\}\,, \text{ and,} \\ 0 \text{ otherwise} \end{cases} \end{aligned}$$

a) Calculate the circular convolution of $x[n]$ and $h[n]$ manually.

b) Using Matlab array operations or vector operations compute the circular convolution of $x[n]$ and $h[n]$

c) Using Matlab documentation study what the inbuilt function circshift does. Compute the circular convolution of $x[n]$ and $h[n]$ using circshift. Compare with the result obtained in (a) and (b)

d) Using Matlab documentation study what the inbuilt function cconv does. Compute the circular convolution of $x[n]$ and $h[n]$ using cconv. Compare with the results obtained in (a), (b), and (c)

e) Using fft and ifft compute the circular convolution of $x[n]$ and $h[n]$ and compare with the results obtained in (a), (b), (c), and (d)

16. Suppose we have two signals defined as follows:

$$\begin{aligned} x[n] &= \begin{cases} (0,1,2,3,4) \text{ for } n \in \{0,1,2,3,4\}\,, \text{ and,} \\ 0 \text{ otherwise} \end{cases} \\ h[n] &= \begin{cases} (4,5,3) \text{ for } n \in \{0,1,2\}\,, \text{ and,} \\ 0 \text{ otherwise} \end{cases} \end{aligned}$$

a) Calculate the linear convolution of $x[n]$ and $h[n]$ manually.

b) Using fft and ifft compute the linear convolution of $x[n]$ and $h[n]$ and compare with the result obtained in (a)

17. Suppose we have a periodic signal $x[n]$ with period 5 defined as follows:

$$x[n] = (0,1,2,3,4) \text{ for } n \in \{0,1,2,3,4\}\,.$$

So $x[n]$ is $(\ldots,0,1,2,3,4,0,1,2,3,4,\ldots)$. Also let $h[n]$ be defined as

$$h[n] = \begin{cases} (3,2,1) \text{ for } n \in \{0,1,2\}\,, \text{ and,} \\ 0 \text{ otherwise} \end{cases}$$

Let $y[n]$ be $x[n]$ considered for 5 periods from $n = 0$, i.e., $y[n] = x[n]$ for $n \in \{0,\ldots,24\}$. Also let $y[n] = 0$ for all other n.

a) Compute the linear convolution of $y[n]$ with $h[n]$ in time domain.

b) Compute the linear convolution of $y[n]$ with $h[n]$ using fft and ifft. Compare your result with that obtained above.

c) Review the overlap and add method taught in class from your class notes. Compute the linear convolution of $y[n]$ with $h[n]$ using overlap and add method. Choose at least 3 disjoint segments for the $y[n]$ signal (i.e., express $y[n] = y_1[n]+y_2[n]+y_3[n]$, such that each signal $y_i[n]$ has an extent which is disjoint with the extents of the other signals $y_j[n]$) . Check whether the linear convolution that you obtain matches with the results above.