Course: DSP Lab **(Banana Problem)**

Session 4: Filter Design using moving average of last 5 samples (Circular convolution, Linear convolution and Convolution performed in frequency domain)

Dept.: Electrical & Electronics Engineering V1.1

Notes:

a) **Use plot() for analog signal plots and stem() for discrete signal plot.**
b) **Every plot must have labels for X-axis, Y-axis and title.**

1 Objectives/Outcome:

1) Design impulse response for moving average
2) Develop function for circular convolution
3) Validate circular convolution and linear convolution equivalence
4) To check convolution in time domain is product in frequency convolution

2 Tasks

2.1 Application

Design a moving average filter for last 5 samples of input. Test the output of filter for input as a sinusoidal frequency of 400Hz with noise added.

Submit a report with code snippets and plotting the outputs: 1) Input signal with noise and play signal to speaker 2) Filter Output with filter realised as a convolution in time domain and play the sample to speaker. 3) Filter Output with filter realised in frequency domain.

Details on how you should approach is provided step by step as tasks below.

2.2 Task1: Develop Circular convolution function

Develop a function $y = \boldsymbol{myCconv}(\boldsymbol{x1}, \boldsymbol{x2})$ that computes circular convolution and returns the same. If the length of the sequences is not the same, then the function needs to append zeros to make them of same length.

Develop a function $y = \boldsymbol{myCconv}(\boldsymbol{x1}, \boldsymbol{x2}, \boldsymbol{N\_in})$ that computes circular convolution:

1) Taking **N_in** as the length of result of circular convolution and returns the resultant circular convolution sequence.
2) If the length of the input sequences is <= n, then the zeros are to be appended.
3) If the length of input sequences is >= n then the convolution is computed only for n number of sequences.

Test the function with the following input cases and plot x1, x2 and output y for different scenarios

1) x1= {1 2 3 4 5} x2 = {3 -1 2 4 5 6 7 8 9 10}, myCconv(x1, x2)

2) x1= {1 2 3 4 5} x2 = {3 -1 2 4 5 6 7 8 9 10}, myCconv(x1, x2, 15)
3) x1= {1 2 3 4 5} x2 = {3 -1 2 4 5 6 7 8 9 10}, myCconv(x1, x2, 5)
4) x1= {1 2 3 4 5} x2 = {3 -1 2 4 5 6 7 8 9 10}, myCconv(x1, x2, 10)
5) x1= {1 2 3 4 5} x2 = {3 -1 2 4 5 6 7 8 9 10}, myCconv(x1, x2, 4)
6) x1= {1 2 3 4 5 6 7 8 9 10} x2 = {3 -1 2 4 5 6 7 8 9 10}, myCconv(x1, x2)

2.3 Task2: Develop Linear convolution using circular convolution

Develop a function $\boldsymbol{y} = \boldsymbol{myLinconv}(\boldsymbol{x1}, \boldsymbol{x2})$ that computes Linear convolution sum using circular convolution function $\boldsymbol{myCconv}()$**.**

Test the function for the following input cases and plot x1, x2 and output y for different scenarios

1) x1= {1 2 3 4 5} x2 = {3 -1 2 4 5 6 7 8 9 10} myLinconv(x1, x2)
2) x1 = {3 -1 2 4 5 6 7 8 9 10} x1= {1 2 3 4 5} myLinconv(x1, x2)
3) x1 = {3 -1 2 4 5 6 7 8 9 10} x1= {1 1 1 1 1 1 1 1 1 1} myLinconv(x1, x2)

2.4 Task3: Define impulse response for moving average

a) Design an impulse signal for moving average of last 5 samples.
$$y[n] = (x[n] + x[n-1] + x[n-2] + x[n-3] + \; x[n-4])/5$$
b) Generate a 400Hz sinusoidal signal for 0 to 2s both inclusive. Plot the same (Sampling frequency: 8000Hz) and get the sound output
c) Add a noise signal using 0.25*rand(size(t)). Plot the same and get the sound output.
d) Perform convolution in time domain using $\boldsymbol{y} = \boldsymbol{myLinconv}(\boldsymbol{x1}, \boldsymbol{x2})$ **(Circular convolution based).**
e) Perform the convolution using product of DFT making use of the built functions fft() and ifft() for the two sequences. Compare against the convolution output to validate the property.
(Note: Plot only 200 samples of data so that you can clearly make out the output)

2.5 Task4: Plot Frequency spectrum

a) Plot DTFT of h[n]
b) Plot DFT of h[n] using FFT

Interpret the results and explain.

Additional Tasks:

You can try this for ECG signal or any other signal with noise. You can find discretised values for the same.