Lab 10 – FIR filter design using windows

In this lab session you will design various N-tap low-pass and high-pass filters using the windowing method with a cut-off frequency $\omega_c = 0.4\,\pi$.

10.1 Windows

Consider a window function $w[n], 0 \leq n \leq N-1$, where N is odd and its corresponding DTFT, $W(e^{j\omega})$.

(a) For each of the window functions below, plot (in same figure) the time domain signal $w[n]$ and the frequency domain representation $W(e^{j\omega})$ over a fine grid of 1001 points. Use matlab inbuilt function `fft()` to compute the frequency transform and use `fftshift()` to display it appropriately. Display the spectrum in dB scale, i.e. plot $20\log|W(e^{j\omega})|$. Repeat this for different window lengths, $N = 51, 71, 91$.

(b) Compare the peak to side-lobe amplitude ratio in dB and width of the main-lobe for each window as N is varied and note this in tabular form. Verify these with the numbers given in the table below.

(c) Window functions – *Rectangular, Bartlett, Hamming, Hanning, and Blackman*. You can use inbuilt matlab functions to generate each of these windows.

(d) What is the effect of changing N?

10.2 Filter design using windows

Design the N-tap low-pass FIR filter as a function which takes window type and length of the filter as input. Plot (in the same figure) the impulse response and the frequency response (magnitude response and phase response) of the filter for each of the above window functions and different lengths of the window.

(a) Is N required to be of odd length? Why?

(b) Are the windows required to be symmetric? Why?

(c) Is the phase response linear for these filters?

(d) Comment on the trade-offs while choosing a specific window shape.

(e) Which window would you prefer? And why?

(f) By utilising the fact that multiplying the ideal low-pass filter impulse response by $(-1)^n$ gives a high-pass filter, design a high-pass filter using the windowing method.

10.3 Filtering of signals

(a) Generate a low frequency sinusoid (< $0.4\,\pi$) and a high frequency sinusoid (> $0.4\,\pi$) and add the two signals. Pass the combined signal through the FIR filters designed above (using `conv()` function in matlab) and plot the result in time domain and frequency domain. Do this for all the FIR filters designed using various windows.

(b) Generate a signal using the music synthesizer code you designed in Lab-5. Pass this signal through one of the designed low pass and high pass filters. Listen to the signals using the `sound()` command before and after filtering.