In this lab, you will learn how to design digital finite-impulse-response (FIR) filters by inverse discrete-time Fourier transform, followed by “windowing”. This method starts off with the frequency characteristics (i.e. discrete-time frequency response) of the desired digital filter of say order-M, adds a phase-shift corresponding to time-delay of M/2, computes the corresponding impulse response (which may have an infinite support and hence may be non-causal), and “truncates” this impulse response (to give it the desired finite support) using a finite-support window.

Prelab:
1. We aim at designing a bandpass FIR filter of order $M$ (of length $M + 1$), with central frequency $f_0 = 100 \text{ Hz}$ and bandwidth $B = 40 \text{ Hz}$.

2. Pick a sampling frequency $f_s$, which is appropriate for our filter. (The value $f_s = 300 \text{ Hz}$ will work well since max frequency under consideration is, $100 + 20 = 120 \text{ Hz}$.) The sampling period is now $T = \frac{1}{f_s} = \frac{1}{300}$.

3. Consider the ideal bandpass filter given by:
   $$H(e^{j\Omega}) = \begin{cases} e^{-j\Omega M}, & \text{for } \Omega_1 \leq |\Omega| \leq \Omega_2 \\ 0, & \text{otherwise} \end{cases}$$

   where $M = 50$, $\Omega_1 = 2\pi(f_0 - \frac{B}{2})T$, and $\Omega_2 = 2\pi(f_0 + \frac{B}{2})T$. Note that the delay is picked equal to $\frac{M}{2}$ such that the impulse response $h[n]$ of the filter is centered at $\frac{M}{2}$ (so truncation over $[0,M]$ preserves the essential features of $h[n]$). Derive the corresponding impulse response and trim it using a rectangular window of length $M + 1 = 51$. Plot the impulse response of your resulting FIR filter in MATLAB.
Laboratory Assignment:

1. Verify your calculations using the MATLAB function `fir1`. Note, to design the filter with a rectangular window, you must first create a window of size equal to the filter length (i.e. $M + 1$). You may use `window = rectwin(M+1)` and then `hr = fir1(M, [w1, w2], window)`. Also note that $w1$ and $w2$ denote normalized frequencies, i.e., $w1 = \frac{f_0 - B}{f_s/2}$ and $w2 = \frac{f_0 + B}{f_s/2}$. (Note that $f_s/2$ represents the maximum frequency when the sampling rate is $f_s$.)

2. Plot the impulse response $hr$ of your filter, and its frequency response. For the latter, you may use the function `freqz`.

3. On the same figure, represent the frequency characteristics of FIR filters designed with the rectangular window, at the same specifications as above, but with orders $M \in \{5, 10, 50, 100, 500\}$.

4. On another figure, represent the frequency characteristics of FIR filters of order $M = 50$ and the same specifications as above, designed with different windows. Use the rectangular window, the hamming window (`Hamming`), the Hann window (`hann`), and the Kaiser window (`kaiser`).

5. Using the Kaiser window, determine the order, normalized frequencies and parameter $\beta$ for designing a FIR filter with the Kaiser window, at the specifications of Figure 1. Design the filter and view its frequency characteristics. Verify if they satisfy the design requirements.
You may use:

\[
[M, \ Wn, \ beta, \ ftype] = \text{kaiserord}(\text{fcuts, mags, devs, fs}).
\]

For our particular example, we have:

\[
\text{fcuts} = [60, 80, 120, 140] \\
mags=[0,1,0] \\
devs=[1e-3,0.207,1e-3] \text{ (corresponding to a passband ripple of 2 dB and} \\
a \text{stopband attenuation of 60 dB, respectively).} \\
fs = 300.
\]

Then you can compute the filter by:

\[
hk = \text{fir1}(M, Wn, ftype, \text{kaiser}(M+1, beta), \text{’noscale’}).
\]

6. Save your code sources and plots for lab-reporting.