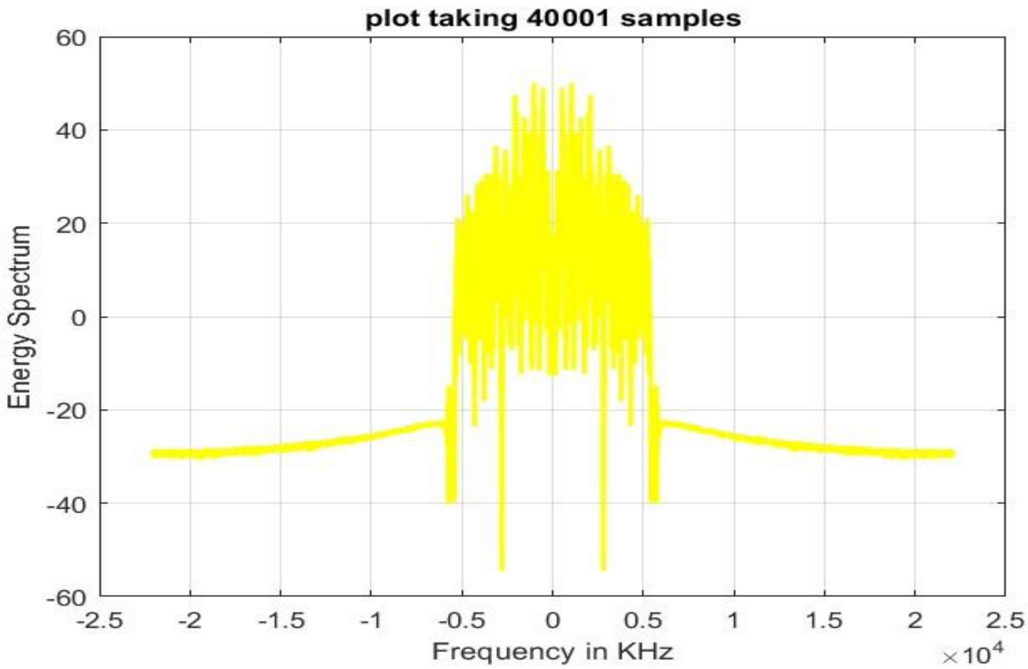
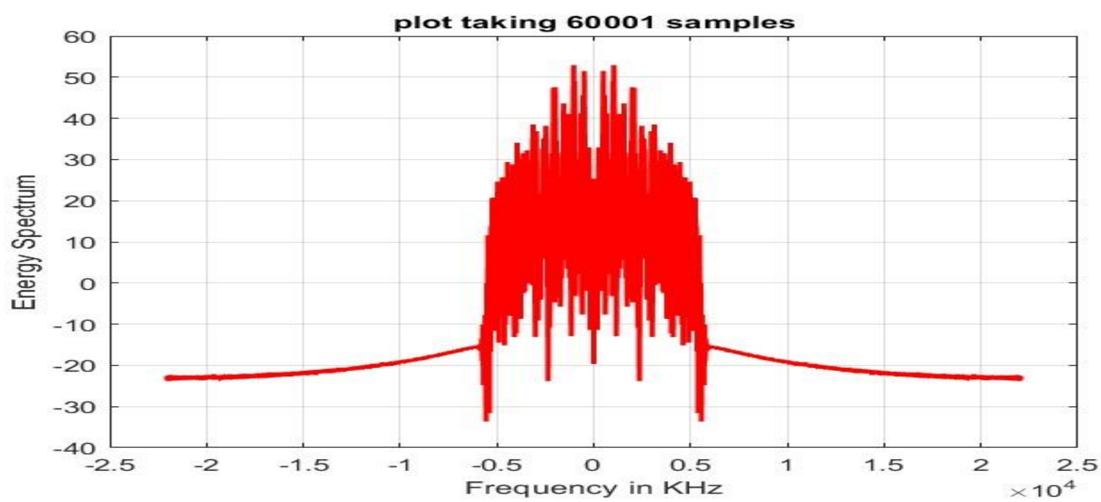
Lab Report for LAB 3 – Signals and Systems

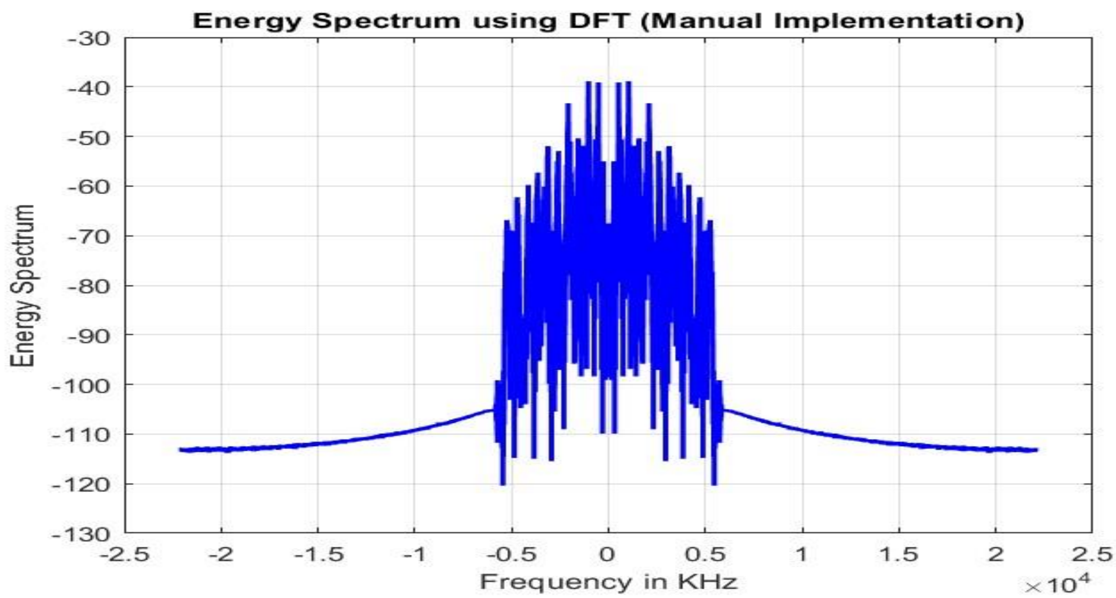
Exercise 1

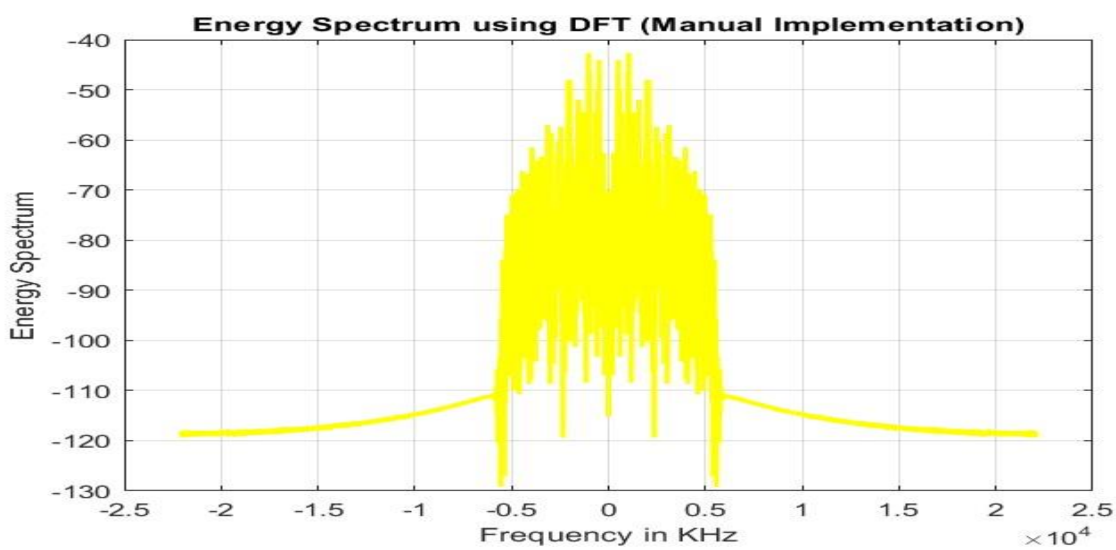




Energy spectra using built-in FFT:

A subset of figures obtained from plotting energy spectra using built-in FFT is shown above. Initially sample sizes are limited to 20001, 40001,60001 and different windows are chosen to perform the Fourier analysis. Then FFTSHIFT is used to put significant portion of the graph in the middle and formula is used to calculate energy spectra. Then the energy spectra are plotted using suitable frequency base. Time taken to perform FFT by MATLAB is found using tic, toc functions, which is about 0.006574 seconds for the sample size of 60001.





Energy spectra using manual DFT: In the above, Two of the figures for energy sepctra obtained from manual implementation of dft is provided. To manually implement dft on matlab, two loops have been utilized, first one going from 0 to N-1 for variable ‘i’ and then the second one going from same range 0 to N-1 for variable ‘k’, and then dft is computed using this formula:

y_dft(i+1) = y2_dft(i+1) + y2(k+1)*exp(-1j * 2 * pi * k * i / N2)

Then normalization has been done and using the formula for energy, energy is calculated and energy spectra is plotted against suitable x-coordinates. This process took significantly longer to finish, the manual dft implementation for 20001 samples took 57.840973 seconds to complete while the one with 60001 samples took 527.838680 seconds to complete.

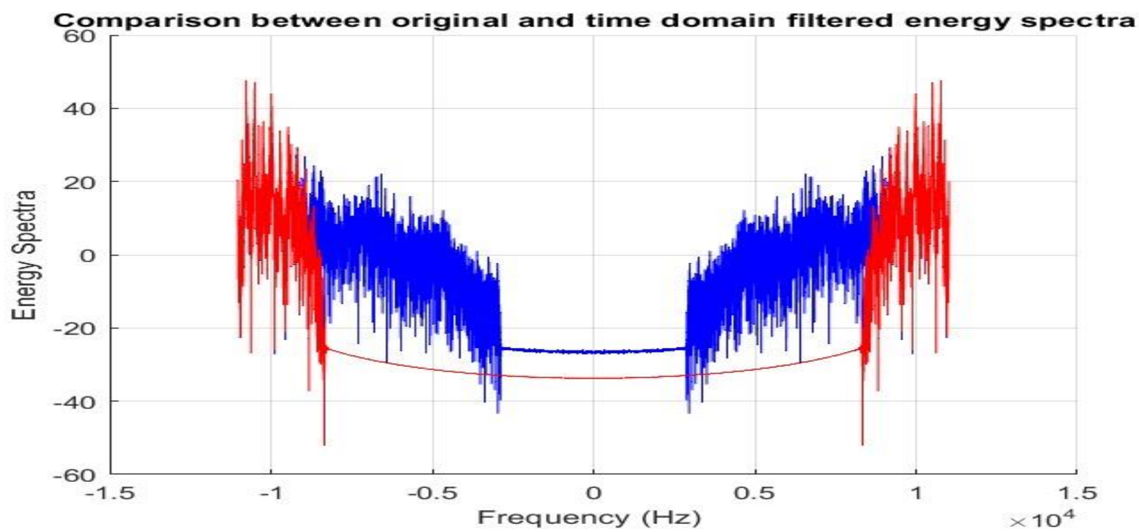
Time Comparison:

Comparing the timing for both the cases, FFT and manual DFT, the largest sample took only 0.006574 seconds for FFT while the largest sample for DFT manual implementation took 527.8386 seconds to complete. This actually verifies the thoeretical estimates as FFT has order of complexity O(nlogn) while DFT has O(n^2) due to usage of two loops.

Exercise 2

Filter in Time domain using convolution:

In this exercise, a portion of the audio file used for exercise 1 is again used to implement different filters on it. In the first case, the proivded impulse response filter h[n] is used which is a raised cosine with bandwidth equal to 10% of the total bandwidth. The first case of filter implementation is done in time domain using the convolution, through matlab built-in conv function. The arguments for conv is the original signal and the given filter. Then energy spectra is found using the same formula as before in exercise 1 and the filtered signal along with the original signal is plotted on the same graph to highlight the contrasts. The figure is proived below:



Filter in Frequency domain using circular convolution:

In the second case, a frequency domain filtering is used. FFT of the signal and given filter is taken. Then circular convolution in frequency domain is utilized to implement the filtering on the signal. After calculating the energy spectra in this manner before and aftet the filtering is done, both the signals, before and after the filtering is plotted on the same graph to highlight the contrasts. The graph is provided below: