Digital Signal Processing (DSP)

DFT (leakage) and window functions

By Zavareh Bozorgasl

In this part, as mentioned in the problem statement, we Create two sinusoids v1 and v2 each with magnitude 1, frequency fc1 = 1000 Hz and frequency fc2 = 1000.5 Hz, both with duration Tdur = 10 seconds. Then create a signal v = v1 + v2.

Part 1:

At first, we take the fft of v (in the Matlab code). As the fft is symmetric for both positive and negative frequencies, we throw away the negative frequency values and just show the positive frequency side. Figure 1 shows fft of v.

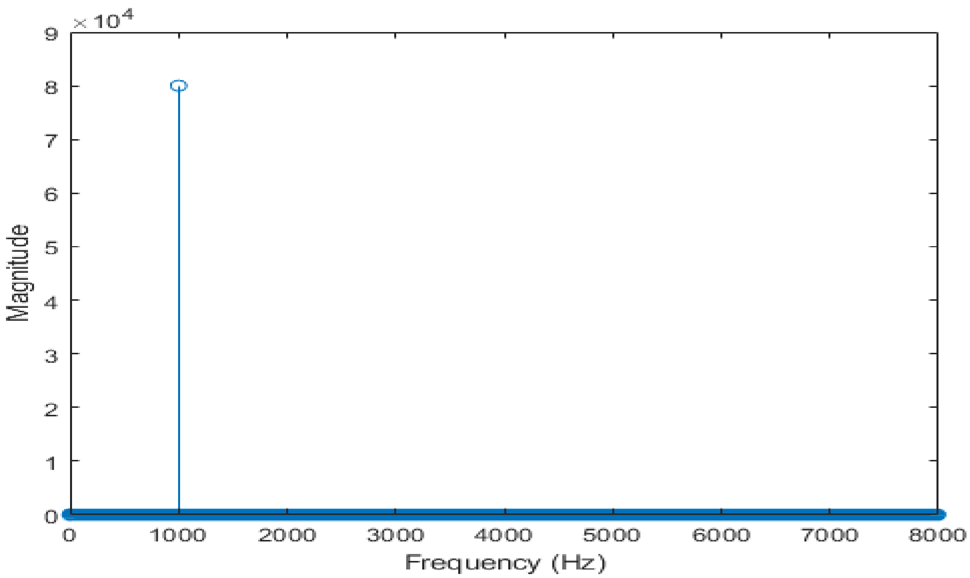


Figure 1. FFT of v versus frequency (Hz).

By zooming on a specific part of the frequency axis from 0.995*fc1 to 1.005fc1, we have the following figure (Figure 2).

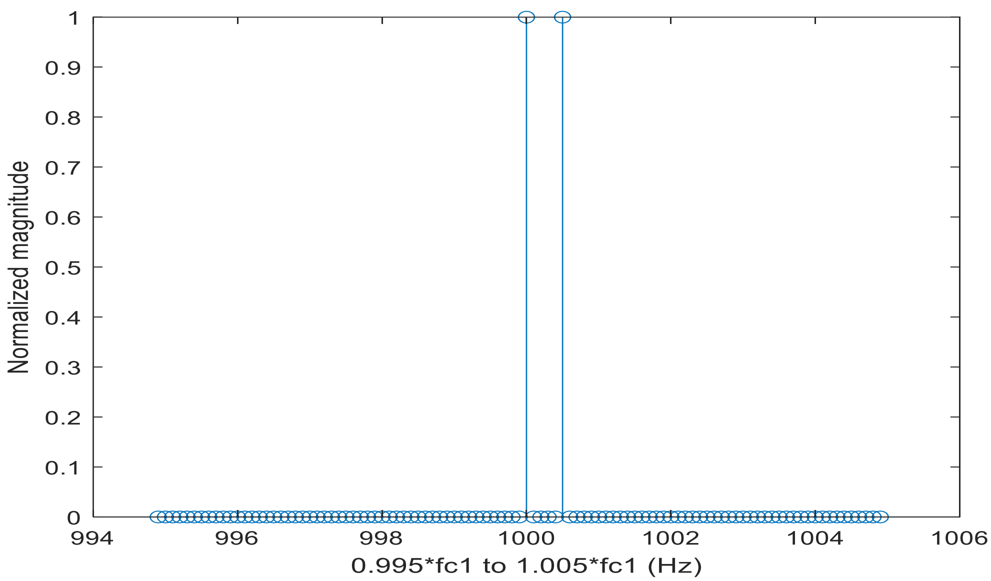


Figure 2. Normalized magnitude of FFT of v versus frequency (Hz);No windowing (rectangular windowing).

Figure 3 shows 20log10(abs(fft(v)/norm) versus frequency. As we expected, the two peaks have been separated to a good extent. Therefore as we have an integer number of cycles and also we have enough resolution for the frequencies, there is no leakage.

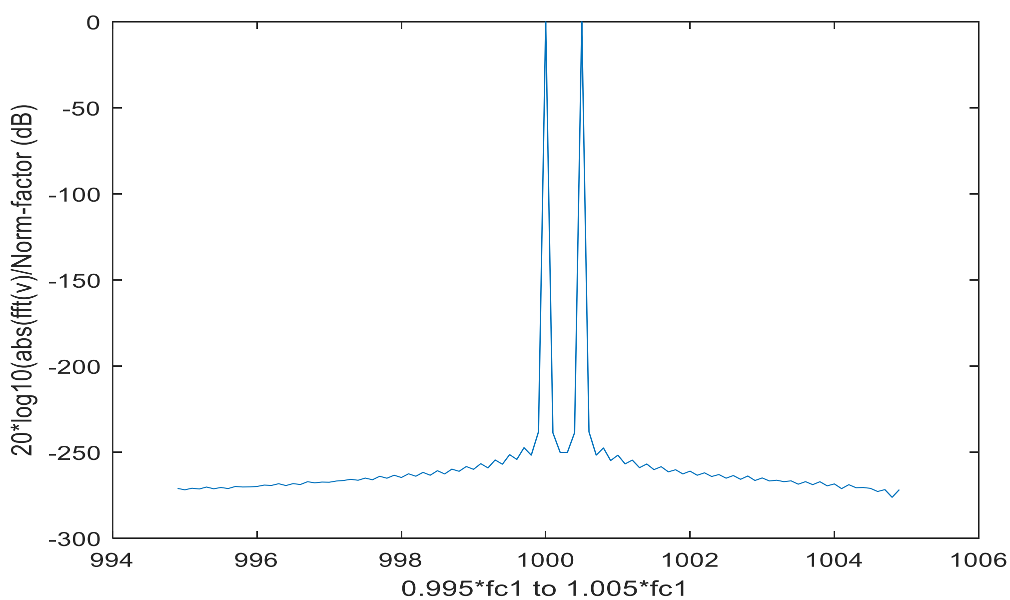


Figure 3. 20log10(abs(fft(v)/norm)) of v versus frequency (Hz); No windowing (rectangular windowing).

This case is similar to applying a rectangular (or flat window to the signal). This results in convolving signal with a sinc function in frequency domain. Indeed, the sinc function is the frequency domain of rectangular window. As we see in Figure 3, the peaks are similar to sinc function.

In the following we will try different windows on the signal. At first, by applying Hann window, we get normalized magnitude of FFT in the specified part of the frequency axis as in Figure 4. Also, 20log10(abs(fft/norm)) is shown in Figure 5.

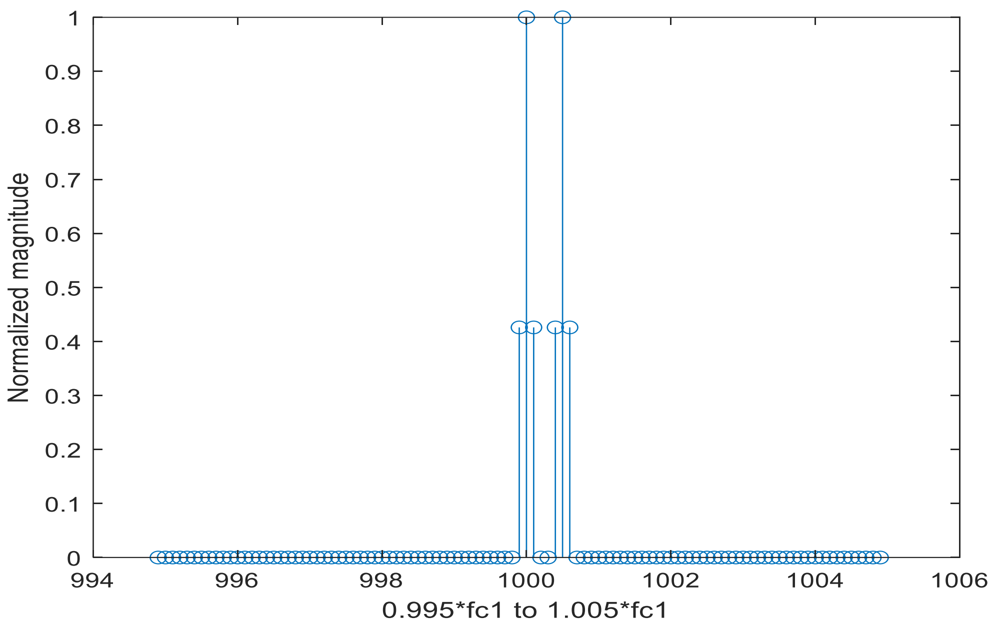


Figure 4. Normalized magnitude of FFT of v versus frequency (Hz); Hann windowing.

we zoom in on this part around the 1000 Hz frequency.

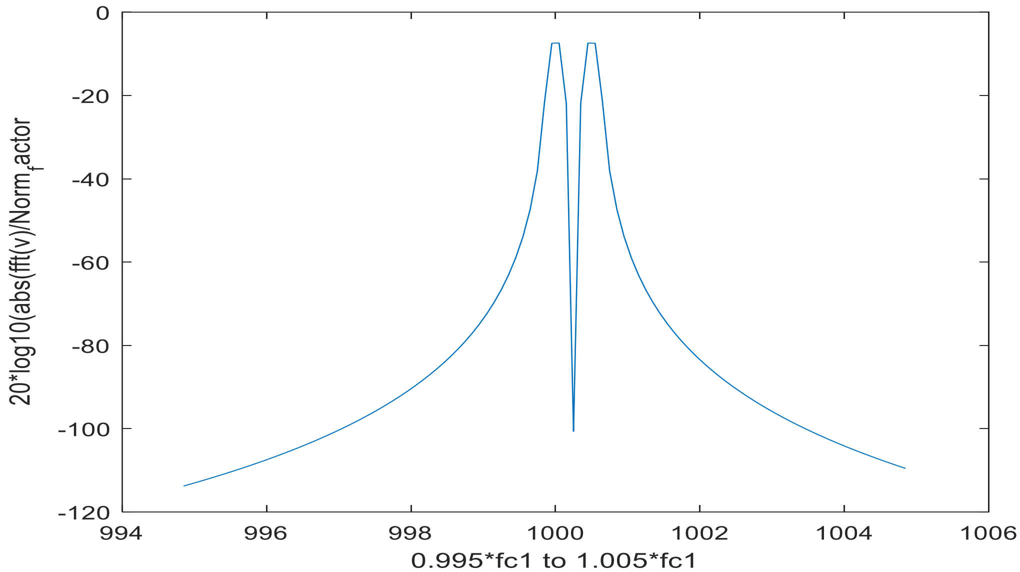


Figure 5. 20log10(abs(fft(v)/norm)) of v versus frequency (Hz); Hann windowing.

After applying Hamming window, we have Figure 6, which is magnitude against part of frequency axis (0.995*fc1 to 1.005*fc1).

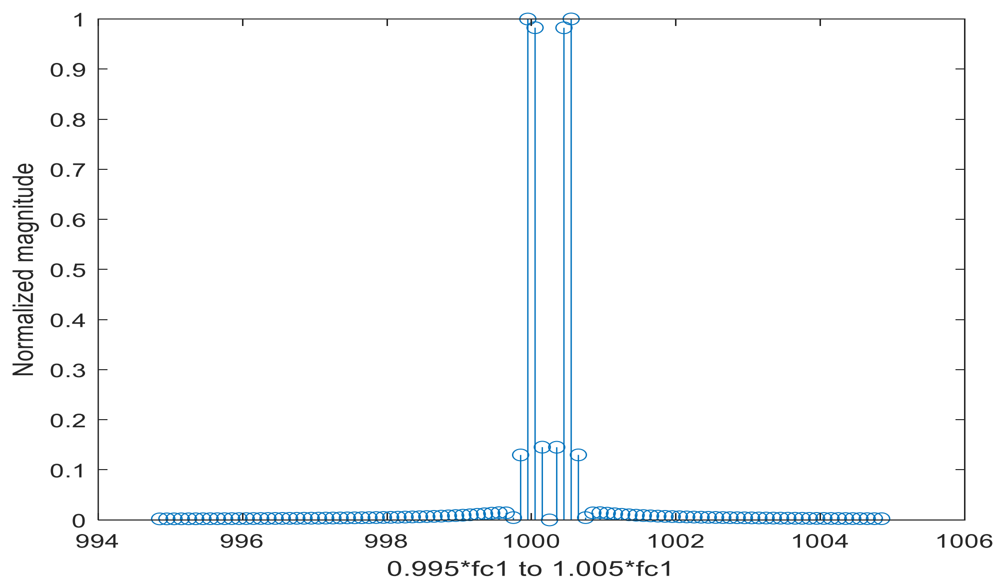


Figure 6. Normalized magnitude of FFT of v versus frequency (Hz); Hamming windowing.

By applying chebwin window, we have

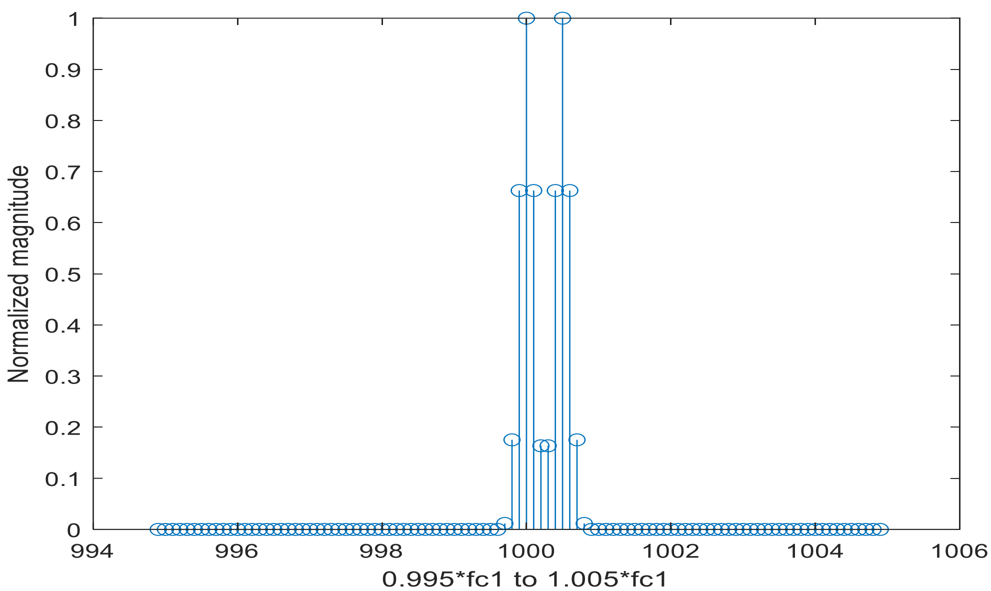


Figure 7. Normalized magnitude of FFT of v versus frequency (Hz); chebwin windowing.

Another window is blackman window,

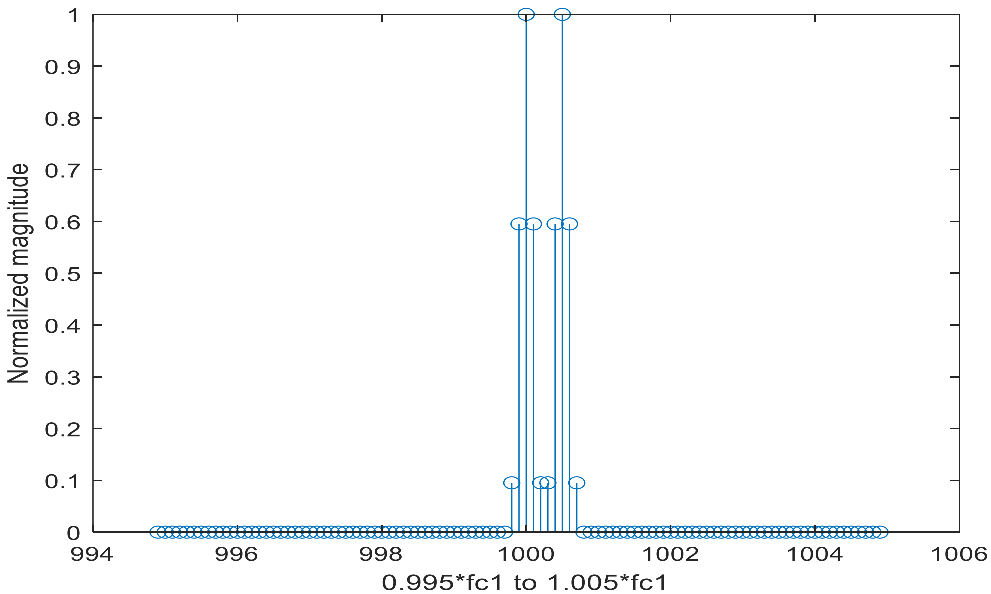


Figure 8. Normalized magnitude of FFT of v versus frequency (Hz); blackman window.

Finally, we plot all the windows in a figure (Figure 9).

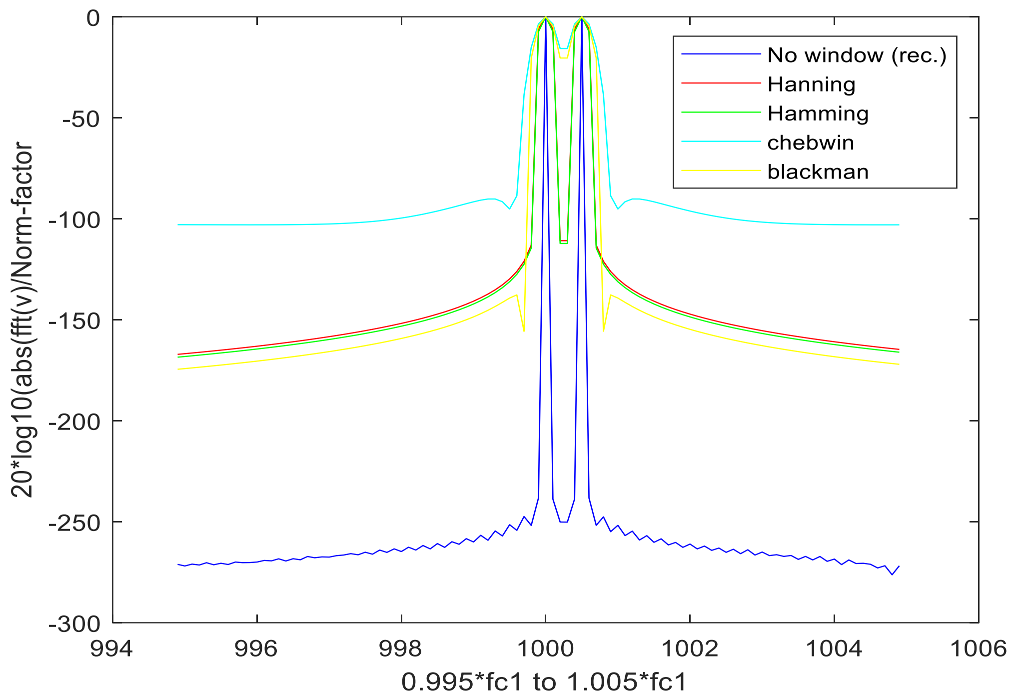


Figure 9. 20log10(abs(fft(v)/norm)) of v versus frequency (Hz).

As we expected, in this case windowing can worsen the situation. As we have two sinusoids with 2 different frequencies fc1 = 1000 Hz and fc2 = 1000.5 Hz and the frequency resolution is fs/N = 16000Hz/160000 = 0.1 Hz , we expect that we have both of these frequencies in the frequency domain (i.e., in the fft). Moreover, in general we have 160000 samples, hence 160000/1000 = 160 cycles of the signal with fc1, and 160000/1000.5 = 159.92 (which is close to 160), and its effect is not clearly obvious for frequency leakage.

we can see that amount of drop off (for frequencies adjacent to fc1 and fc2) are as- the first one is drop off for the sample left to fc1, the second one for the sample between fc1 and fc2, and the third on is for the sample right to fc2, all based on dB: No window: -250,-250,-250 -----------Hann: -110,-110,-110--------Hamming-110,-110,-110--------------chebwin:-100,-100,100 ------------blackman:-150,-110,-150.

As a summary, Spectral leakage due to FFT can happen by: mismatch between desired tone and chosen frequency resolution, and also time limiting an observation (not being an integer multiple of cycles of sinusoids). As we don't have these two problems in this part (just a bit for the second signal), hence, the case of no windowing gives the best answer; because, introducing windows which is indeed, a convolution in frequency domain, has its effects on the signal.

Part 2:

By changing Tdur = 10 seconds + 0.5*T, where T = 1/fc1, we have a fraction of cycles of sinusoids. This will cause having leakage in fft. After applying the windows, the captured signal won't be perfectly replicated, but the leakage is now confined over a smaller frequency range. There still may be leakage from the primary data bin into other bins [1]. Indeed, window functions create slightly modified signals. When we don't apply any window, Figures 10 and 11 show the results.

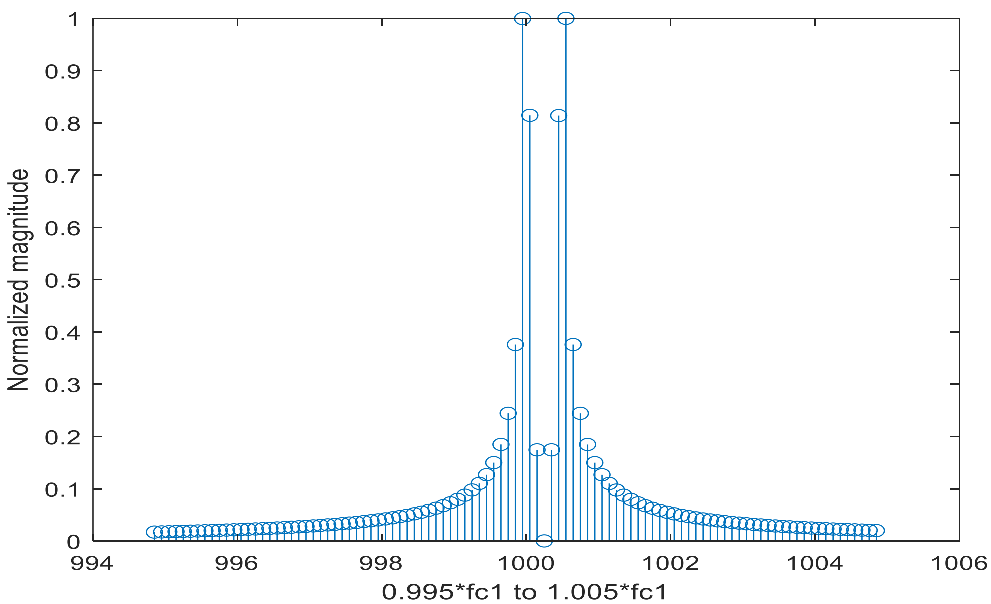


Figure 10. Normalized magnitude of FFT of v versus frequency (Hz); No windowing (rectangular windowing).

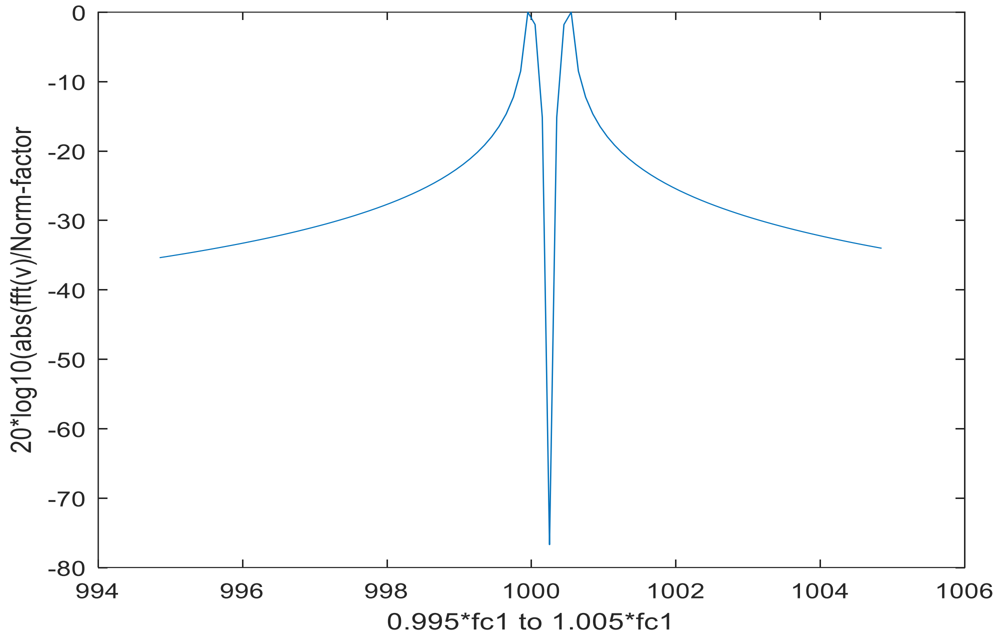


Figure 11. 20log10(abs(fft(v)/norm)) of v versus frequency (Hz); No windowing (rectangular windowing).

After applying Hann window, we get Figures 12 and 13.

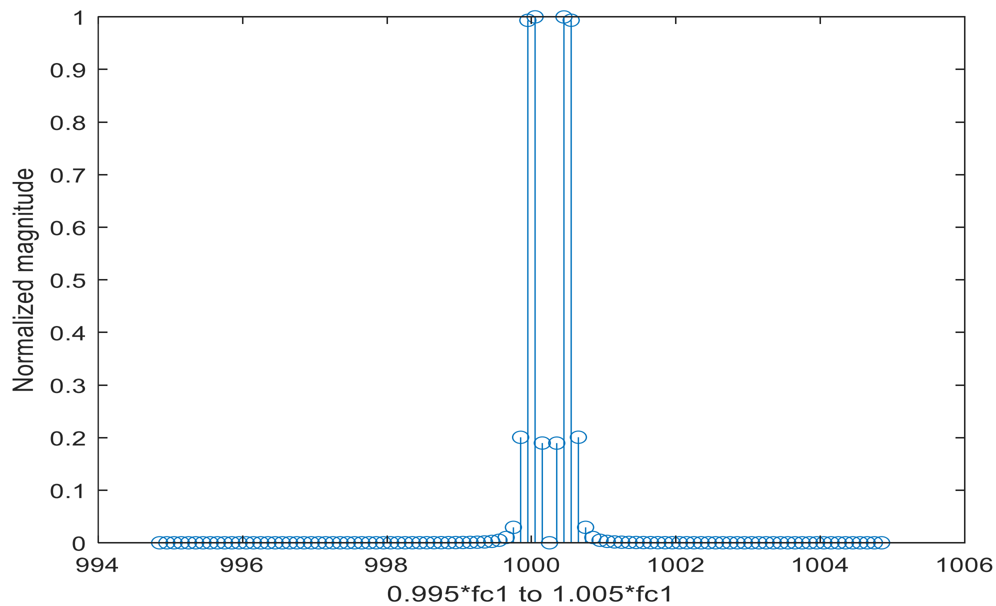


Figure 12. Normalized magnitude of FFT of v versus frequency (Hz); Hann windowing.

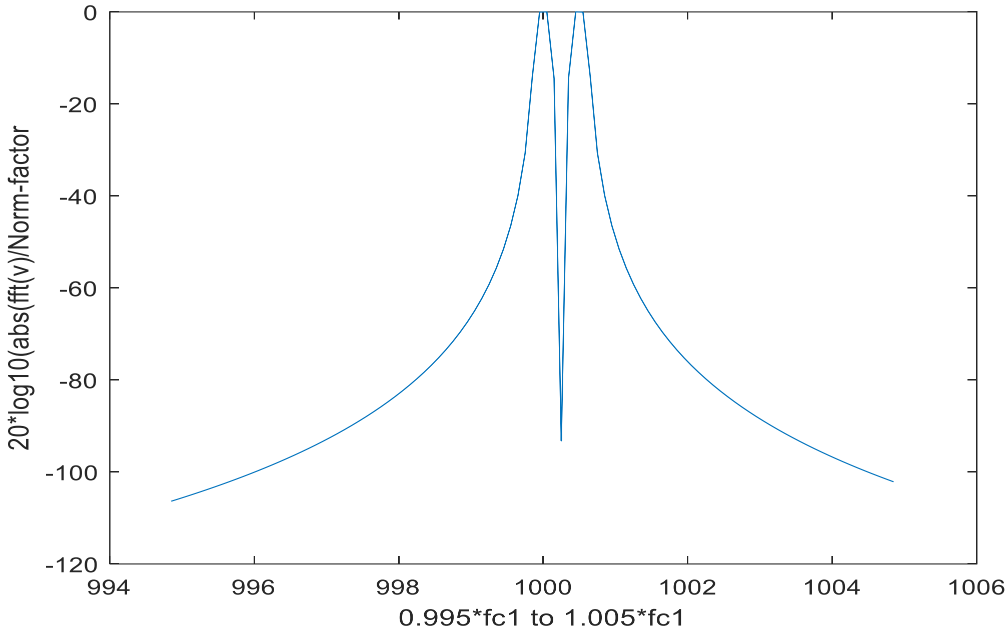


Figure 13. 20log10(abs(fft(v)/norm)) of v versus frequency (Hz); Hann windowing.

Figures 14 and 15 presents the signal which is resulted after applying Hamming window.

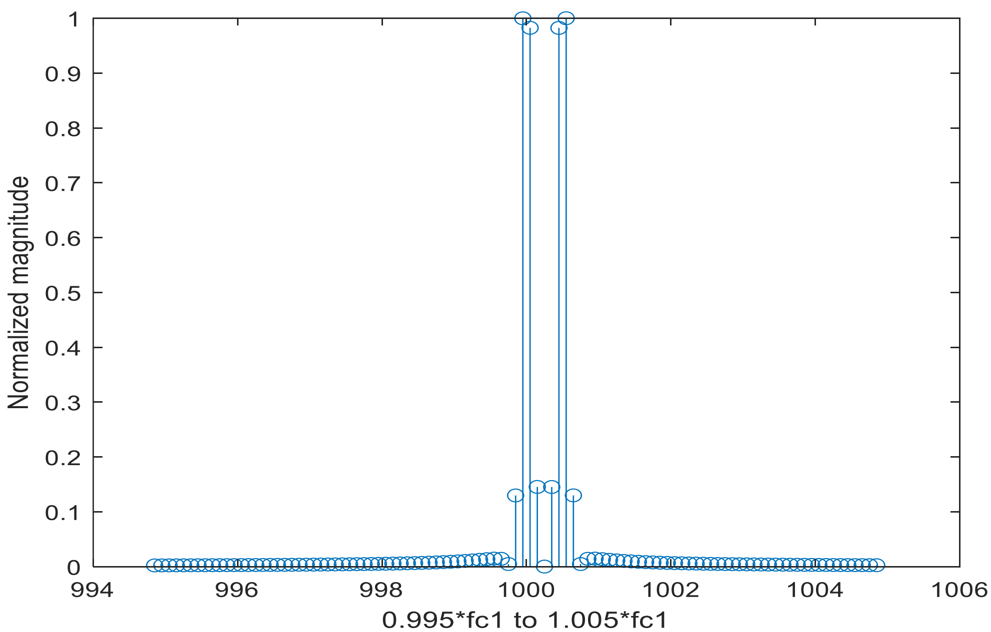


Figure 14. Normalized magnitude of FFT of v versus frequency (Hz); Hammin windowing.

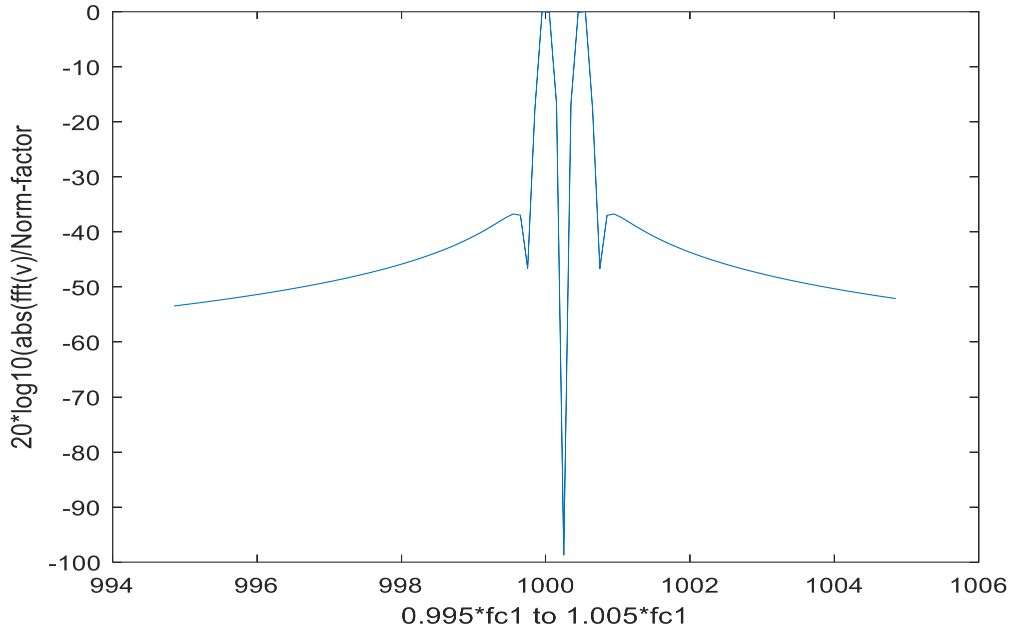


Figure 15. 20log10(abs(fft(v)/norm)) of v versus frequency (Hz); Hamming windowing.

Figures 16 and 17 presents the signal which is resulted after applying Chebwin Hamming window.

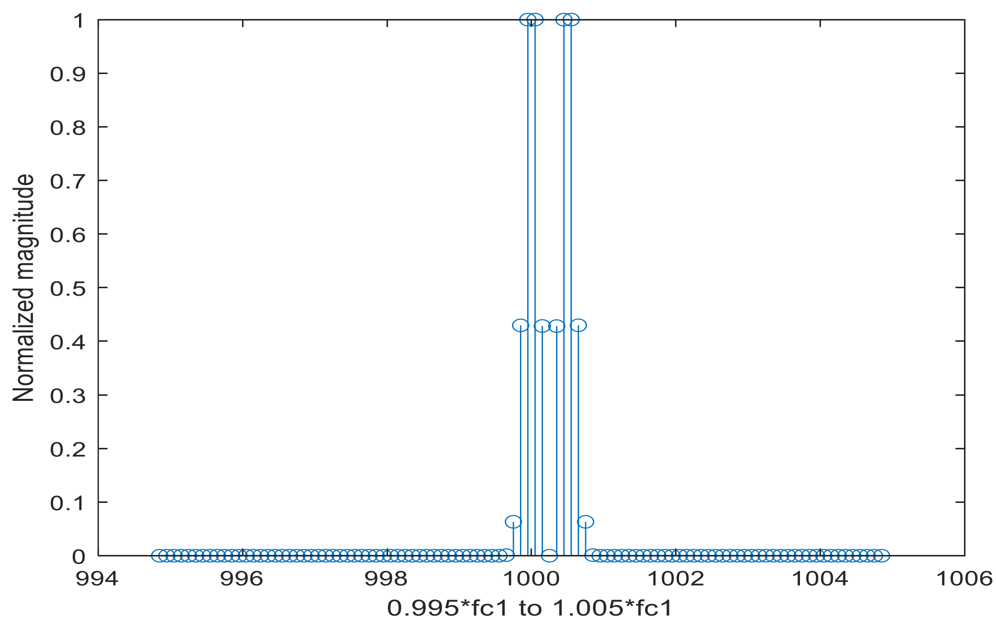


Figure 16. Normalized magnitude of FFT of v versus frequency (Hz); chebwin windowing.

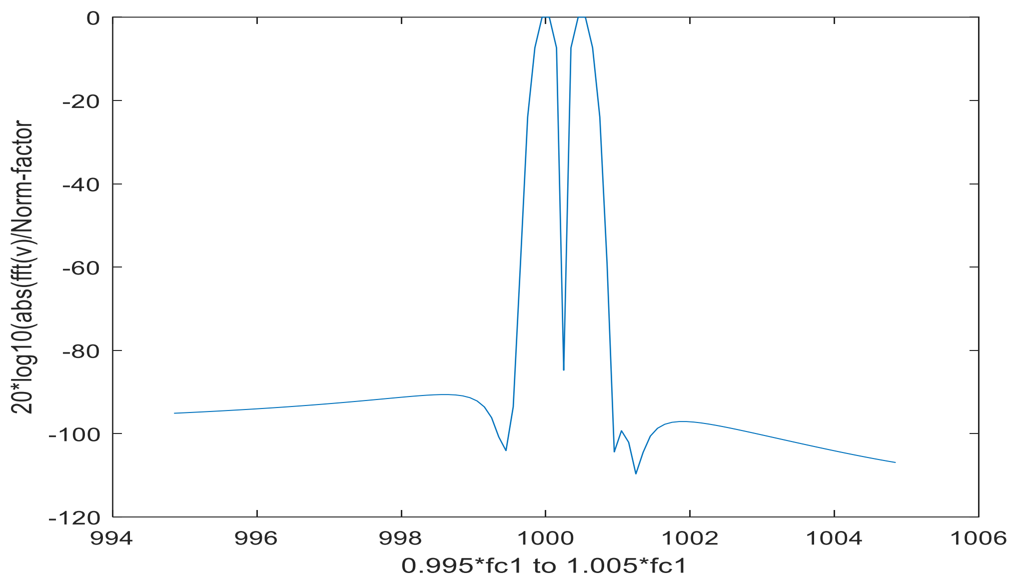


Figure 17. 20log10(abs(fft(v)/norm)) of v versus frequency (Hz); chebwin windowing.

Figures 18 and 19 presents the signal which is resulted after applying blackman window.

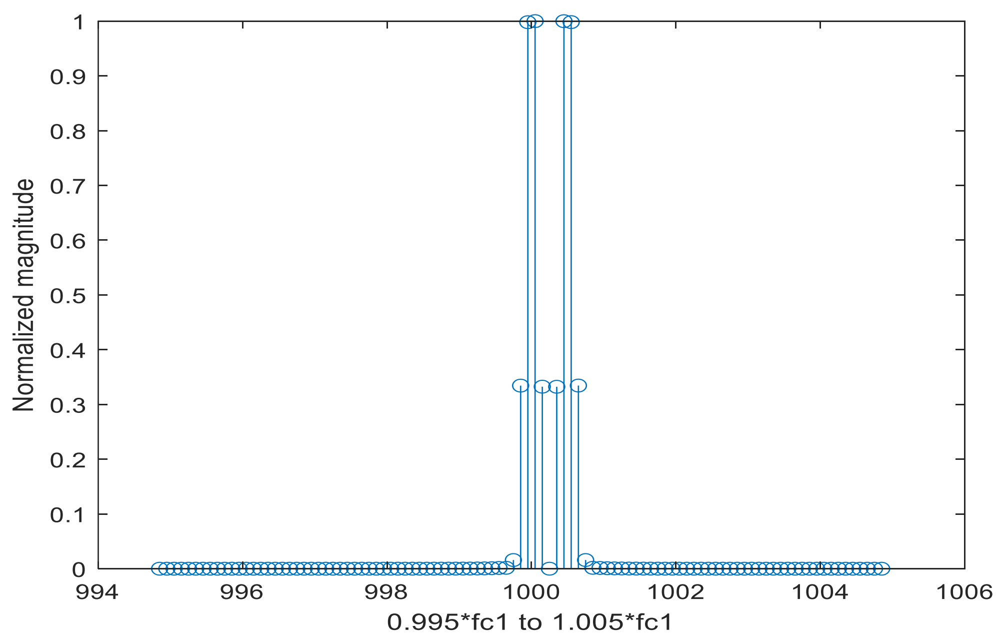


Figure 18. Normalized magnitude of FFT of v versus frequency (Hz); blackman windowing.

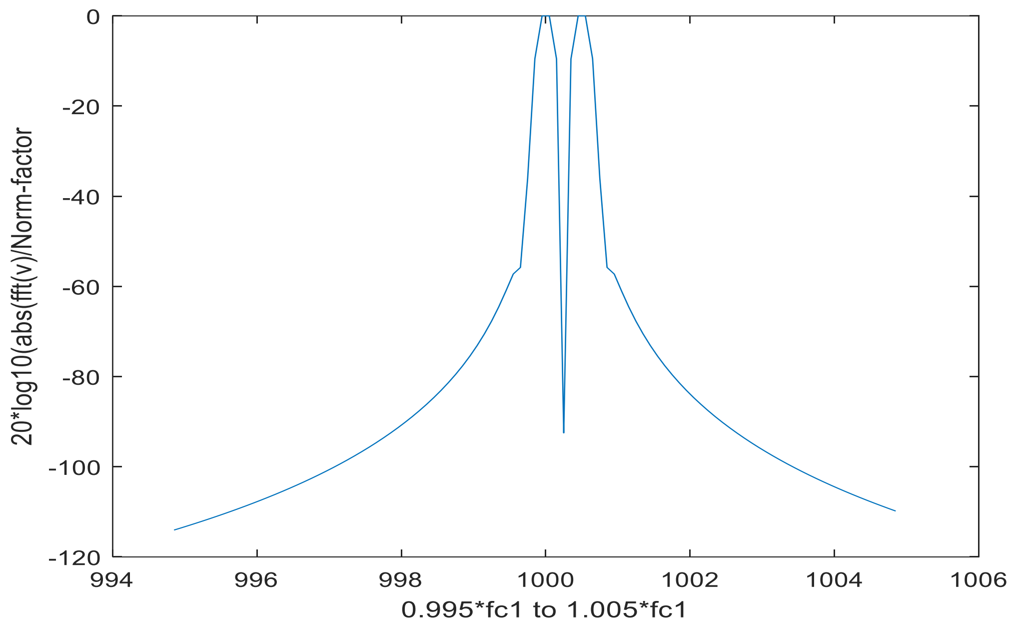


Figure 19. 20log10(abs(fft(v)/norm)) of v versus frequency (Hz); blackman windowing.

In Figure 20, a comparison of the all applied windows is presented.

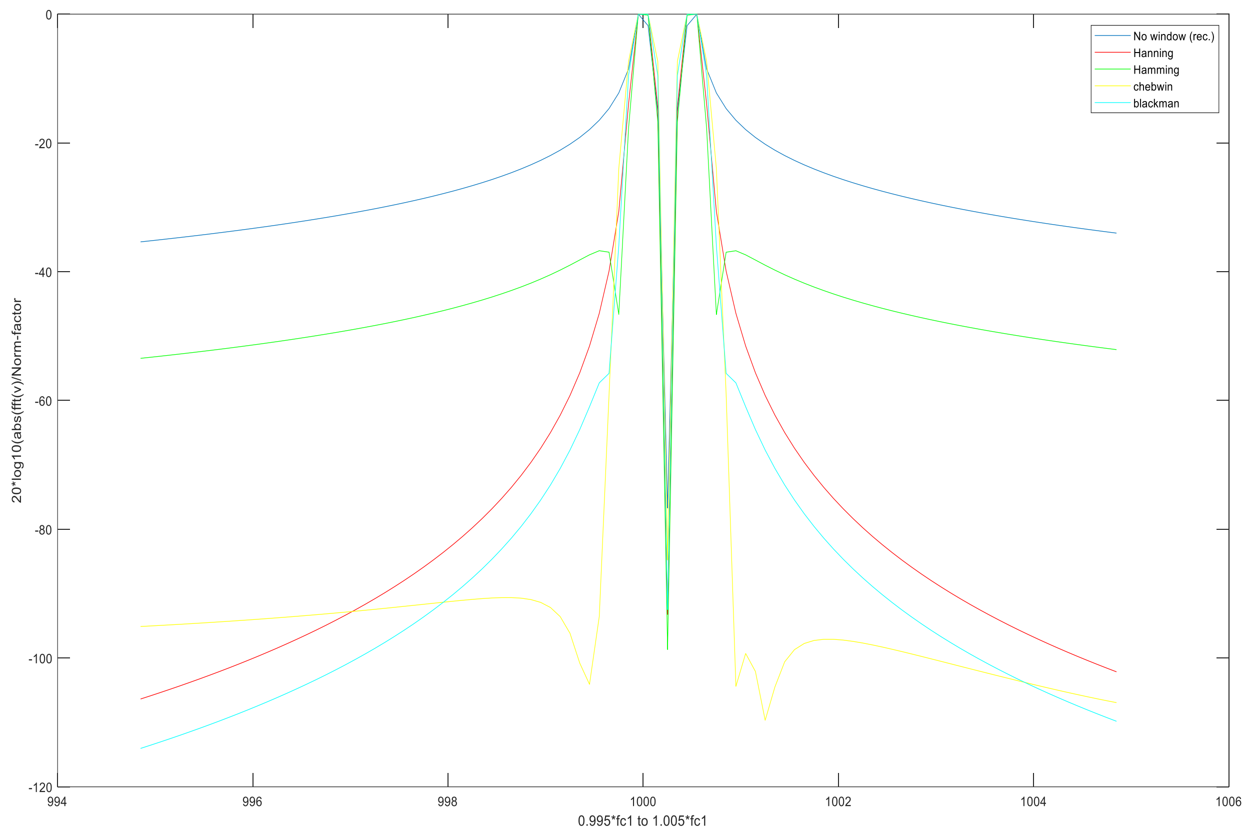


Figure 20. 20log10(abs(fft(v)/norm)) of v versus frequency (Hz).

As we have seen from above figures, chebwin windowing comparatively give the best performance if we concern a significant drop off after our mainlobes. We can see a comparison of different window functions as in the below figures. Figure 21 shows the case of rectangular window (the figures are taken from Matlab-I have plotted different windows in Matlab to see them in different lengths, etc.).

we can see that amount of drop off (for frequencies adjacent to fc1 and fc2) are as-the first one is drop off for the sample left to fc1, the second one for the sample between fc1 and fc2, and the third on is for the sample right to fc2, all based on Db): No window: -20,-100,-20 -----------Hann: -40,-100,-40--------Hamming:-50,-100,-50--------------chebwin:-110,-100,100 ------------blackman:-60,-100,-60.

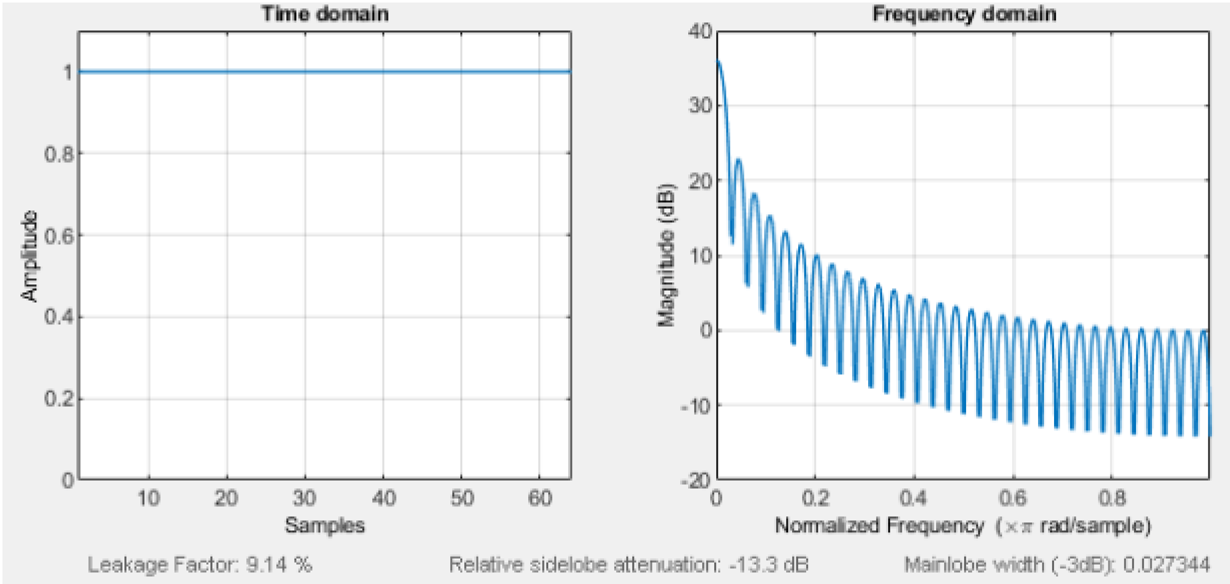
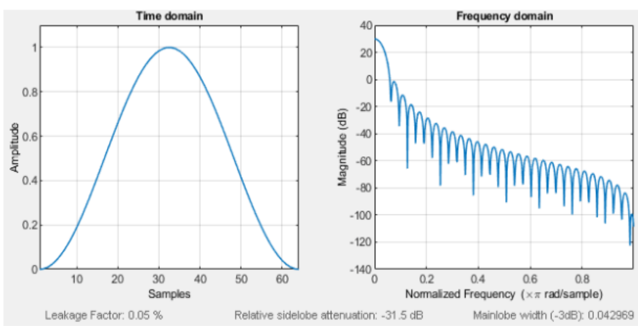
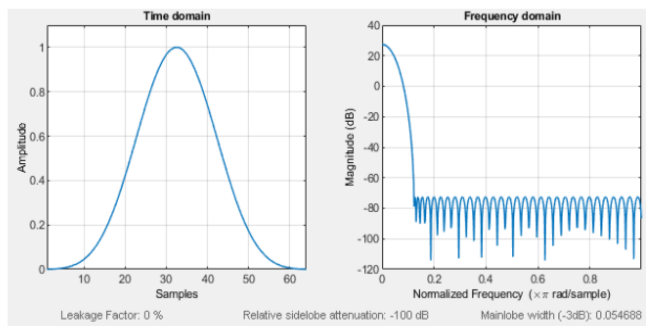


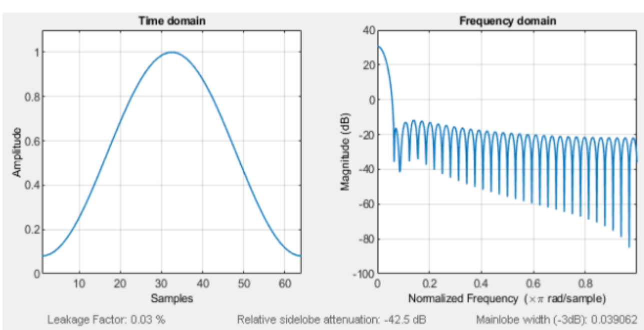
Figure 21. Rectangular window (time and frequency domain)



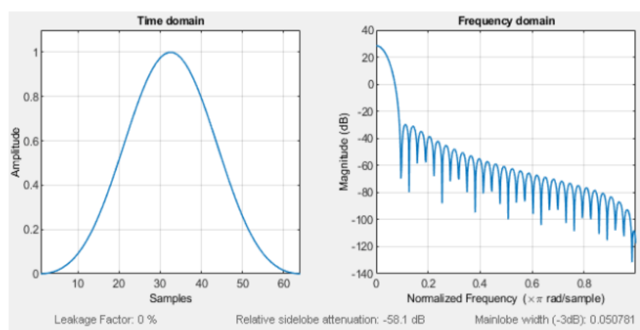
a. Hanning



b. Chebwin



c. Hamming



d. blackman

Figure 22. Comparison of different windows (time and frequency domain)

As it is obvious from the figure, the Hamming window and Chebwin has comparatively more drop off in their first sidelobes, and this shows itself in Figure

20. Moreover, the rectangular window has the least amount of sidelobe drop off, and the effect is obvious from Figure 20.

In windowing, the height of the side lobes represents the affect the windowing function has on frequencies around main lobes. Typically, lower side lobes reduce leakage in the measured FFT but increase the bandwidth of the major lobe [2].

The Hann window touches zero at both ends eliminating all discontinuity. The Hamming window doesn't quite reach zero and thus still has a slight discontinuity in the signal. This difference leads to Hamming window does a better job of cancelling the nearest side lobe but a poorer job of canceling any others.

There are different criteria for choosing window functions. According to [2], if the signal contains strong interfering signals near the frequency of interest, choose a window function with a low maximum side lobe level; If the frequency of interest contains two or more signals very near to each other, spectral resolution is important, hence, it's better to choose a smoothing window with a very narrow main lobe. For our problem, Chebwin is acting better than the other ones.

Part 3:

For this part, we have Tdur = 1.0 seconds. Hence, fs = 16000, and N = 16000 (number of samples). Therefore, resolution is fs/N = 1 Hz. As we have two frequencies at fc1 = 1000 Hz, and fc2 = 1000.5 Hz, the fc2 could not be sampled properly, because of the frequency resolution. Indeed, we have to increase the number of samples (cycles), but we have to have in our mind that the number of cycles has to be an integer multiple of our signal cycle.

In case of not applying any window, we get Figures 23-24.

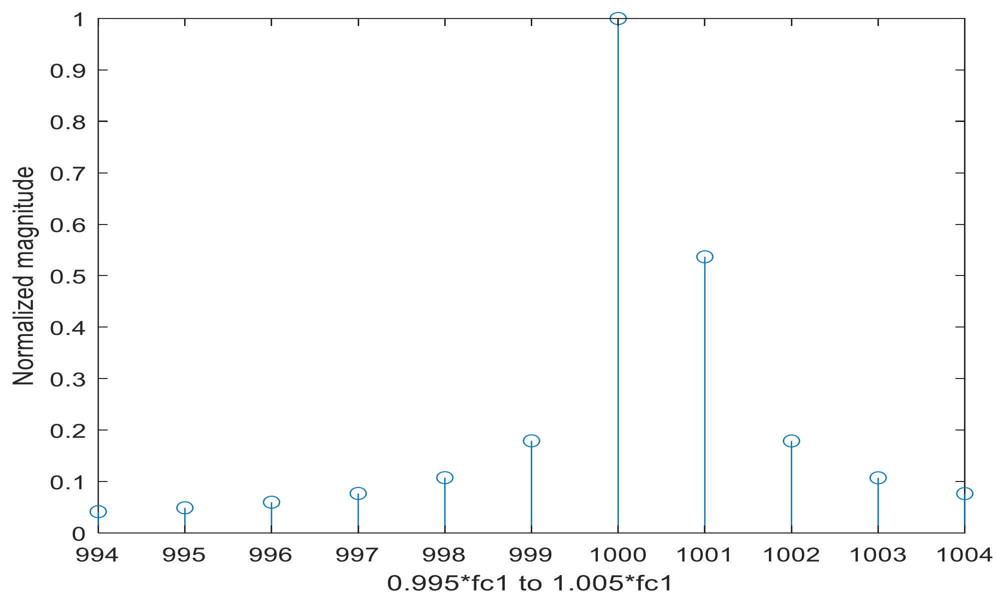


Figure 23. Normalized magnitude of FFT of v versus frequency (Hz); No windowing (rectangular windowing).

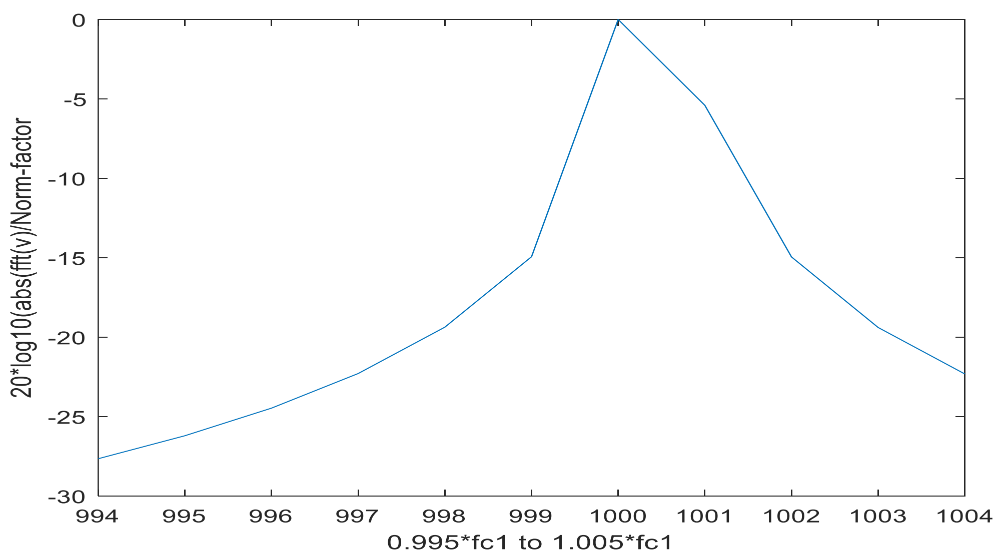


Figure 24. 20log10(abs(fft(v)/norm)) of v versus frequency (Hz); No windowing (rectangular windowing).

And by applying chebwin window, we get the following figures (Figures 25 to 27). Indeed, Figure 25 is zooming in part of the interval which is about 200 samples greater than what we zoomed on it (0.995*fc1 to 1.005*fc1); indeed, the frequency axis of Figure 25 is (0.995*fc1 - 100 to 1.005*fc1 + 100).

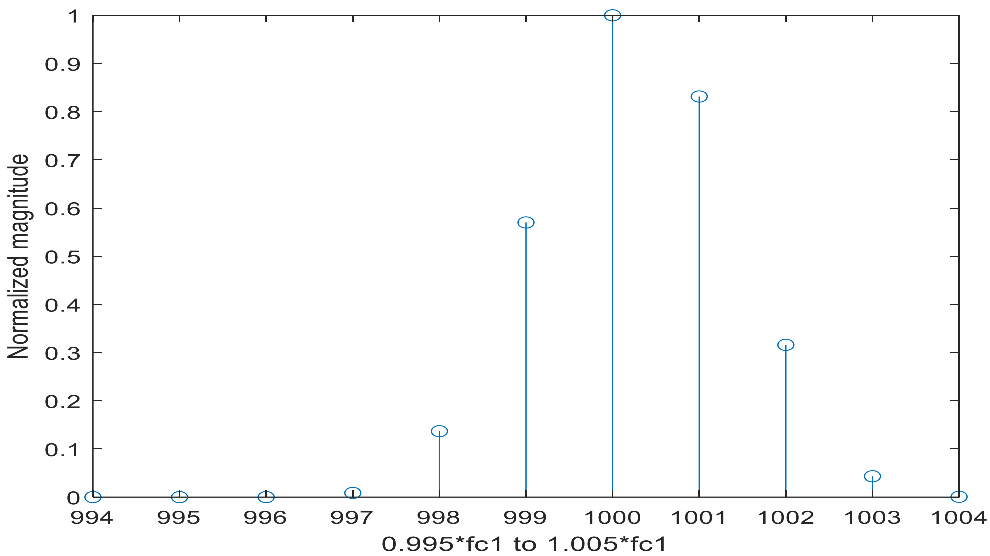


Figure 25. Normalized magnitude of FFT of v versus frequency (Hz); chebwin windowing.

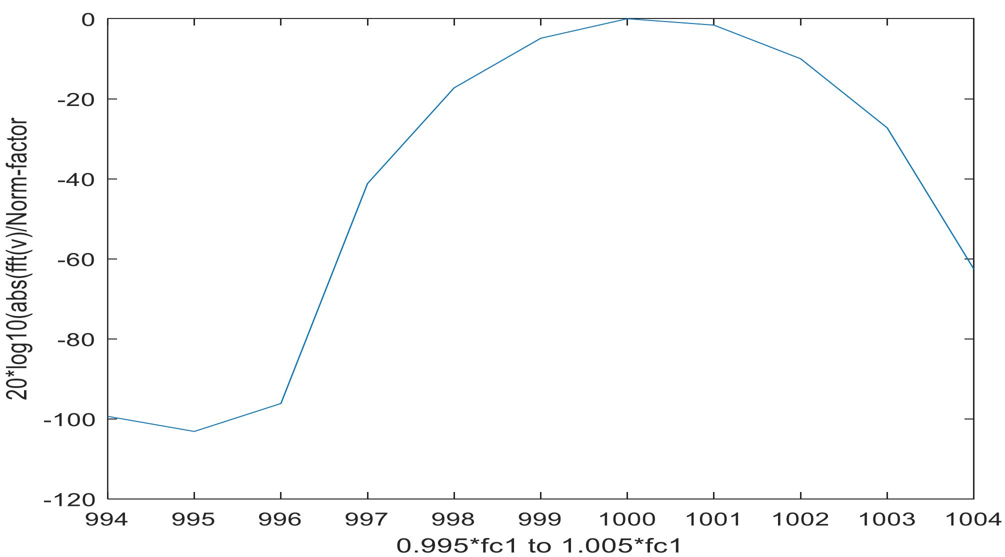


Figure 26. 20log10(abs(fft(v)/norm)) of v versus frequency (Hz); chebwin windowing.

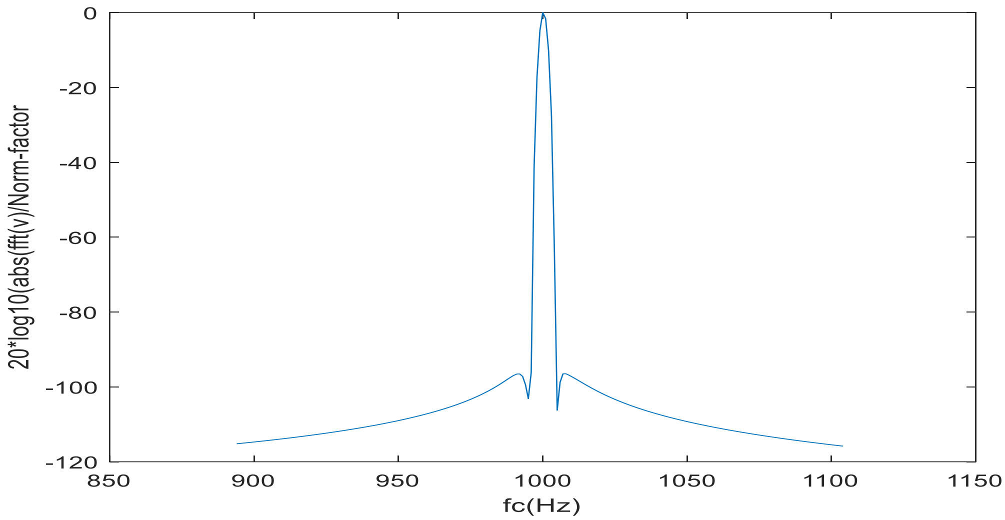


Figure 27. 20log10(abs(fft(v)/norm)) of v versus frequency (Hz); chebwin windowing.

It is obvious that even Chebwin window cannot separate the two frequencies. As the two frequencies are close (just 0.5 Hz difference), and also we do not have an integer multiple of the cycles (16000/1000.5 = 15.99), the leakage causes serious problems. Moreover, the low number of samples (and number of fft points) causes frequency resolution of 1 Hz, and with this frequency we do not expect to get fc = 1000.5 Hz (as our signal is sum of these two (fc1 and fc2 frequencies) and its period is LCM of these two signals).

[1] https://www.modalshop.com/rental/learn/basics/how-to-choose-fft-window

[2] https://www.ni.com/en-us/innovations/white-papers/06/understanding-ffts-and-windowing.html