Semester Project
EXPERIMENTATION OF TRANSFORMS
Department of Electrical and Computer Engineering
Missouri University of Science and Technology

EE215
5/6/2014

Abstract:

In the following report, the students compared the different Fourier transforms for their function and accuracy in MATLAB. The students compared the DTFT, DTFS, FFT and DFT algorithms for their usefulness under different criteria. For the experiment, the students changed the sampling frequency and time windows to see the accuracy of each transform and the time it would take to make these calculations

Table of Contents:

Introduction pg. 3

Background and Theory pg. 3-4

Procedure and analysis pg. 5-6

Conclusion pg. 7

Appendixes

- Appendix A – Graphs pg. 8-49
- Appendix B – MATLAB Code pg. 49-72

Introduction:

Over the past semester, the students have learned about transformations from the time domain into the frequency domain. In the following experimentation, the students used the DTFT, FFT, DFT, and Modified DTFT algorithms to find the transforms of a cosine and rectangle function. The students looked at the effect the sampling frequency had on the transform. The students, also, looked at the effect that the time window had on the transform. The Fourier transforms take the time domain signal given and change it into the frequency domain. The resulting graph in the frequency domain is called the spectral content of a signal. The students saw in this experiment how the accuracy of the spectral content changes with the changing of specific parameters.

Background and Theory:

The Discrete Time Fourier Series (DTFS) is a transform that takes the time domain signal and changes it into the frequency harmonic domain. In which you see results at the evenly spaced harmonics. It is seen as the discrete frequency domain. Just as the discrete time domain is a sampling of a continuous time function the discrete frequency is the sampling of the continuous frequency. The DTFS is not explored in the body of this experiment, but it is important to mention because it allows for transforms into the discrete frequency domain and is the root for the other transforms.

The Discrete Time Fourier Transform (DTFT) is used to take a discrete time signal and transform it into a continuous frequency domain. The DTFT extends the DTFS, so that it can handle aperiodic signals. The DTFT is shown as a sum from $-\infty$ to ∞ which is important because it shows that it can't be practically represented without error. In order to implement the DTFT, you can use an approximation of a large sample size. The DTFT of periodic signals will be an approximation because of the fact that even with a large sample size it won't be equal to infinity. However, the DTFT of aperiodic will be exact because all the data of the signal can be found in a finite range.

The Modified DTFT is used in this experiment was used to show the correctness of the DTFT. The Modified DTFT was found by taking the mathematically found DTFT and convoluting it with a windowing function. The window function will show only the values on mathematical description that are contained within the windowing function. Using the table of DTFT transforms, the students found the DTFT of the rectangle and cosine functions. The convoluting with the window function will give the mathematical description over a finite number of values. The window function that was used was a rectangle function which is the normal windowing function because it doesn't corrupt the values, but still gives the results over the desired time frame.

The Discrete Fourier Transform (DFT) is used as the same transform as the DTFT, but it is found only over one period. The DFT is very similar to the calculation for the DTFS. The key to the DFT is the choice of the window that will allow for accurate data, but not overwhelming time. The downside to the DFT is the time it takes to make the calculations. For an N-point DFT you will need to make N^2 calculations. This can lead to major problems when dealing with a finite about of processing speed. The DFT algorithm needs to have high sampling frequency (fs) and the spectral content of the signal will be shown from –fs/2 to fs/2. The sampling frequency is important to get accurate results. For low sampling frequencies, the results will be greatly different.

The Fast Fourier Transform (FFT) is almost identical to the DFT except it takes more efficient samples. The FFT is the practical application of the DFT. Due to the fact that the DFT takes N^2 computations; DFT is not used in Digital Signal Processing (DSP). However, the FFT made this possible by limiting the number of calculations to N*log(N) which is incredibly more effective. Fast Fourier Transforms allowed for accurate DSP on embedded systems with limited processing speed.

As mentioned in the DFT, the sampling frequency is important for these transforms. In the case of periodic signals, like sinusoids, there exists a criterion, the Nyquist Criteria. The Nyquist Criteria states that in order to get meaningful data form the transform you need to sample at a frequency two times greater than the fundamental frequency of the original signal. For samples below this range, the results are greatly different than the theoretical. This criterion is very important when calculating the transforms.

For all the transforms, periodic signals are difficult to get an accurate spectral content and leakage occurs. Leakage is the phenomenon that exists when the spectral content has stray values that don't exist in the theoretical. The calculations of the transforms need to be practical and finite and this representation truncates the periodic signal which can lead to leakage. The best way to remove this leakage is to increase the sampling frequency time window. The more samples that are obtain then the practical representation becomes more like the periodic signal.

Procedure and analysis:

The students started by sampling the discrete time function. For the rectangle function, the students took increasingly large sample sizes by varying the sampling frequency. The key to the sampling of the rectangle function was the shifting from the time to sampled time steps. (t->n/fs) The rectangle function took many more samples, but kept the ratio of the rectangle function the same. The cosine function shortened the period of the cosine giving more time values over a given time.

The students then moved to calculate the DTFT of the samples. The students took multiple sampling frequencies and pushed them through the DTFT algorithm. The rectangle function showed that the higher the sampling frequency was the more accurate the values for the spectral content were. They students, also, increased the time window in which the discrete time function was graphed. The increase of the time window made no major difference in the accuracy of the spectral content because the increase of the time window only padded the time window with zeroes. The sampling frequency for the rectangle function was the only criteria that made an impact. For the cosine function, the sampling frequency and the time window played a major role in the accuracy of the spectral content. The students took values below the Nyquist criteria which resulted in garbage values that had no real meaning. However, once the sampling frequency reached a value over two times the given frequency the values appeared in the correct places. The changing of the sampling frequency and time window increased the accuracy of the spectral content by decreasing the leakage and giving the proper output frequencies and amplitudes. The increase in the time window greatly decreased the amount of leakage in the system.

The students then moved into the Modified DTFT algorithm which took the theoretical result and convoluted the result with a window function. The students changed the size of the window function along with the sampling frequency. For both the rectangle and cosine function the students saw an increase in the accuracy of the spectral content with the increase of the windowing function and the sampling frequency. The Modified DTFT is another way of showing that the DTFT is a function that depends on the time window.

The students then moved on to the DFT algorithm which was modeled using a similar time window function. The students used the window function created for the Modified DTFT which was based upon the sampling frequency. For the rectangle function, the time window function needed to be adjusted from the previous version. For the cosine function, the window function was fine with moving from 0 to a specific time value because it is a periodic and it can be windowed by starting from an arbitrary value. However the rectangle window function needed to be adjusted, so that it would show the whole aperiodic signal. A window function starting at zero would only give half of the aperiodic signal. With the normal function, the students saw that the amplitude of the transform was exactly half of what it was required to be. So, the students adjusted the windowing function and got legitimate values.

The FFT algorithm was the last algorithm implemented by the students which was rather uneventful because it is a given MATLAB function. The FFT algorithm didn't need to be amplitude shifted due to the fact that it was accounted for the MATLAB code. It was obvious that the FFT code took an incredibly short amount of time compared to that of the other transforms. The FFT gave a spectral content that was very accurate for the aperiodic rectangle function, but had a small shift in some values of the periodic cosine function.

Conclusion:

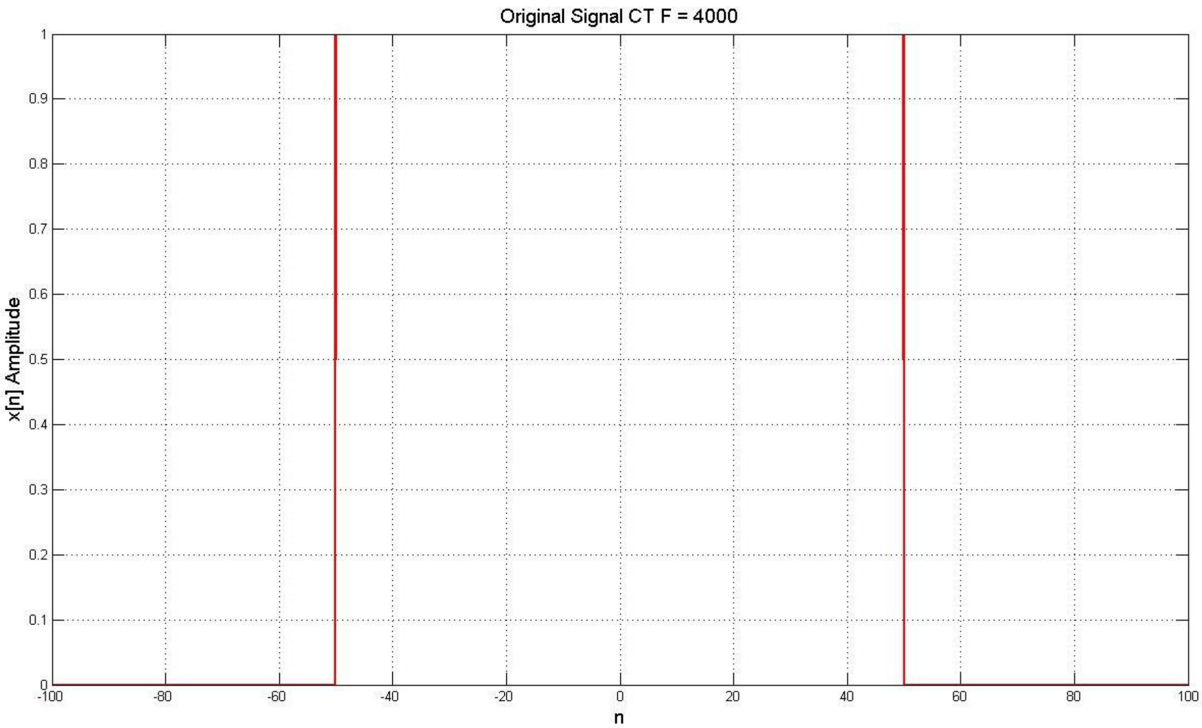
Cosine Function:

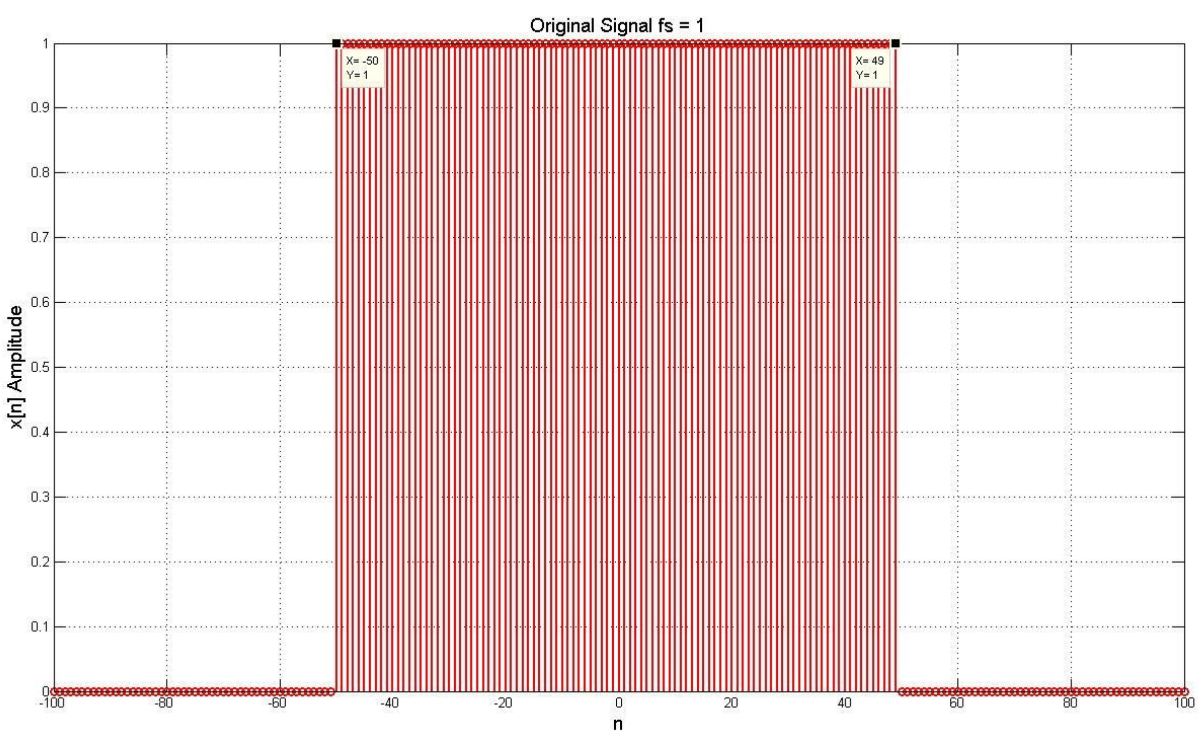
Overall the cosine function is a periodic function that will have a periodic nature in the transform to the frequency domain. According to the equation sheet the transform will be a repeating pair of impulses. The calculation in MATLAB showed that the function had impulse peaks at the given frequency in the cosine. The increase in the sampling frequency allows for the results to be more accurate. The Nyquist Criteria states that in order to get reliable results you need to choose a sampling frequency that is greater that two times the given frequency. Through the experimentation (shown below), the students saw that values below the Nyquist Criteria was not reliable or accurate. Some of the graphs had an increase in the amount of 'leakage' while others had values where they shouldn't be. The 'leakage' of a cosine function can be seen in the Sinc profile found in the transform. Based on the sample size and sampling frequency, the spectral content of the original signal may have a more defined Sinc profile. The students saw that increasing the sampling frequency and time window would minimize leakage and allow for a more theoretical profile to form. The more that you increase the time window and sampling frequency the more that the Sinc profiles centered at the given frequency turn into impulses. The spectral content of the signal increases in accuracy as the time window and sampling frequency, but that takes time to calculate the increase in data points. So, for many embedded systems, the time needed to make these calculations is a very important parameter. For many of these transforms with the frequencies and time windows, the transform wouldn't be feasible to actually implement on a embedded system due to the fact that the signal would take too long to process accurately. The FFT algorithm is definitely the best ratio of accuracy and time taken to compute the spectral content of the signal.

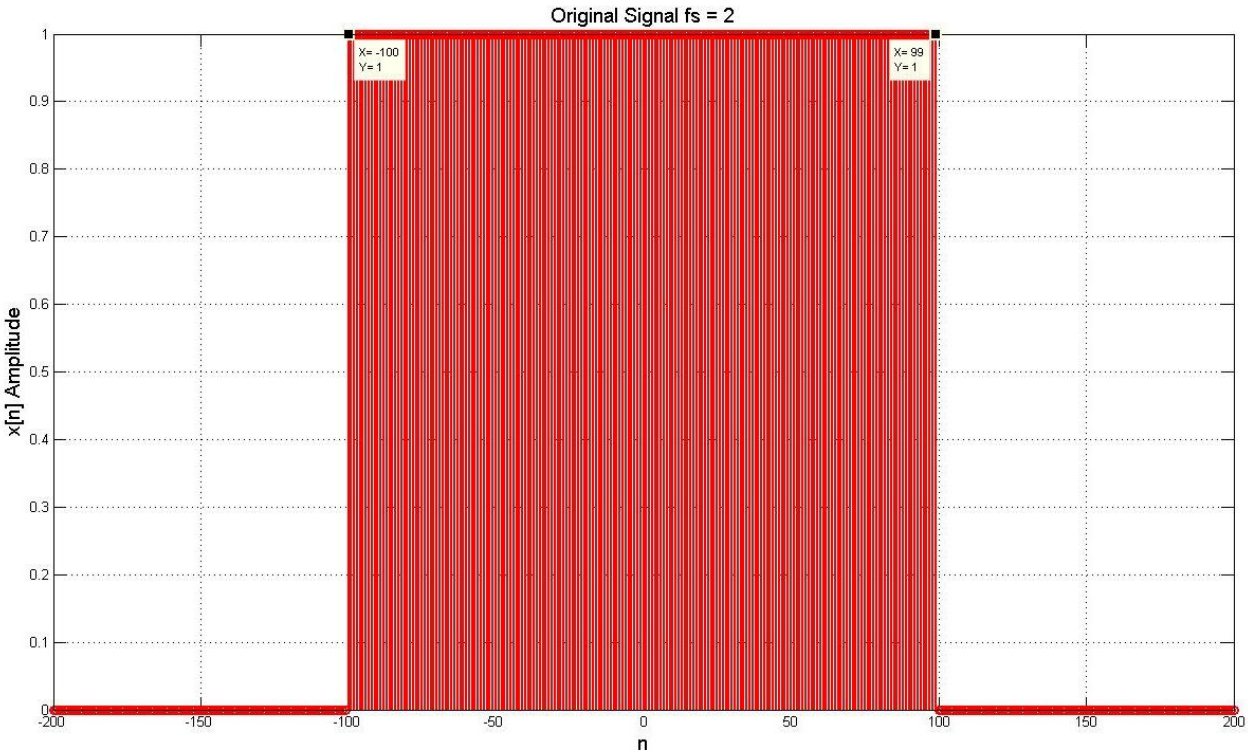
Rectangle Function:

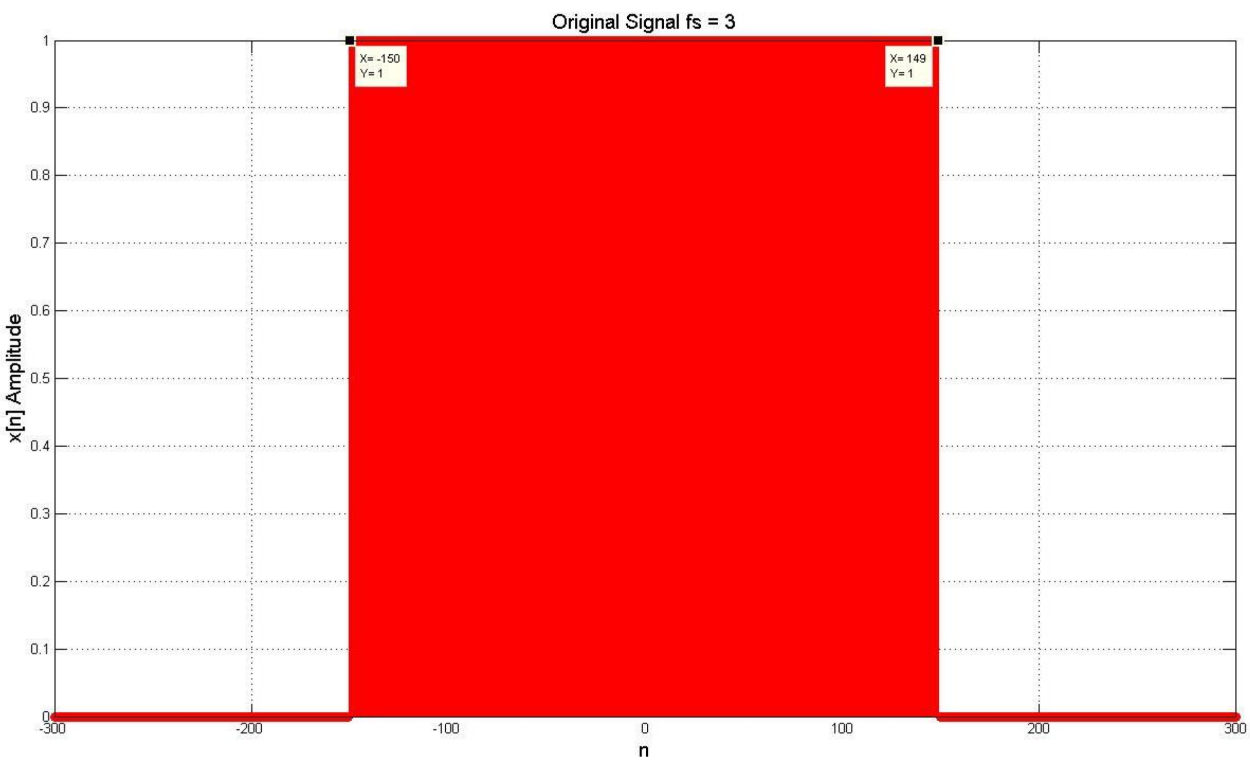
Overall the rectangle function is an aperiodic function that will, according to the table of transforms will be a Dirichlet function which can also be represented as a Sinc function convoluted with a comb function. The students calculated all of the transforms and found that the transform gave the spectral content from either –fs/2 to fs/2 or -.5 to .5. The FFT and DTFT gave its spectral content from -.5 to .5, so the signal needed to be zoomed in on to see the content accurately. The DFT and Modified DTFT gave the spectral content from –fs/2 to fs/2. For the DFT the students got a spectral content that did repeat and show the Dirichlet function, but the students only showed a representation period of –fs/2 to fs/2 which is only the Sinc profile. The rectangle function, due to its aperiodic nature, behaved differently than the cosine function when it came to some transforms. The students needed to use different amplitude scaling factors for the two functions due to which parameters would affect the single. The rectangle used the sampling frequency to amplitude scale the transforms because that is the main variable used to make the data more accurate. However, the cosine function used the length of the discrete time vector which was crucial to the periodic cosine function and the amplitude of the spectral content. The rectangle function had very little noticeable leakage because of two factors. The first factor would be the theoretical value of the transform allows for the leakage to be 'hidden' inside the Sinc profile. The second factor would be the aperiodic nature of the signal which would transform nearly perfectly with the DFT algorithm. The rectangle function gave more reliable results overall for all of the transforms due to the fact that all of the meaningful values were used in the transform. The time needed to make these calculations was a little more obvious because of the large sample sizes. The students chose relatively small sampling frequencies, compared to cosine, and were still able to get very large sample sizes. The sampling frequency for the rectangle function didn't need to be very high in order to get accurate results. I would assume that aperiodic functions would be easier for embedded systems to process. Systems with limited processing speed would be able to process rectangle functions with relatively low sampling frequencies and still obtain accurate data.

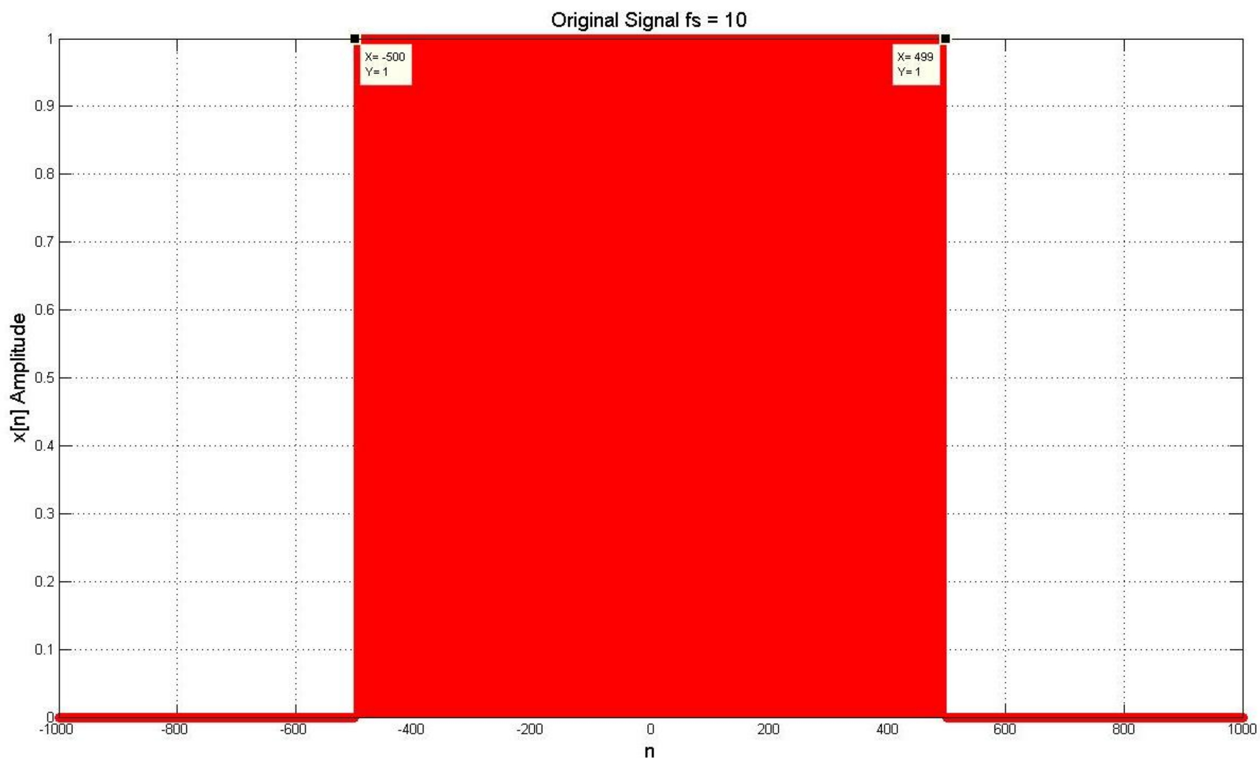
Appendix A (Graphs):

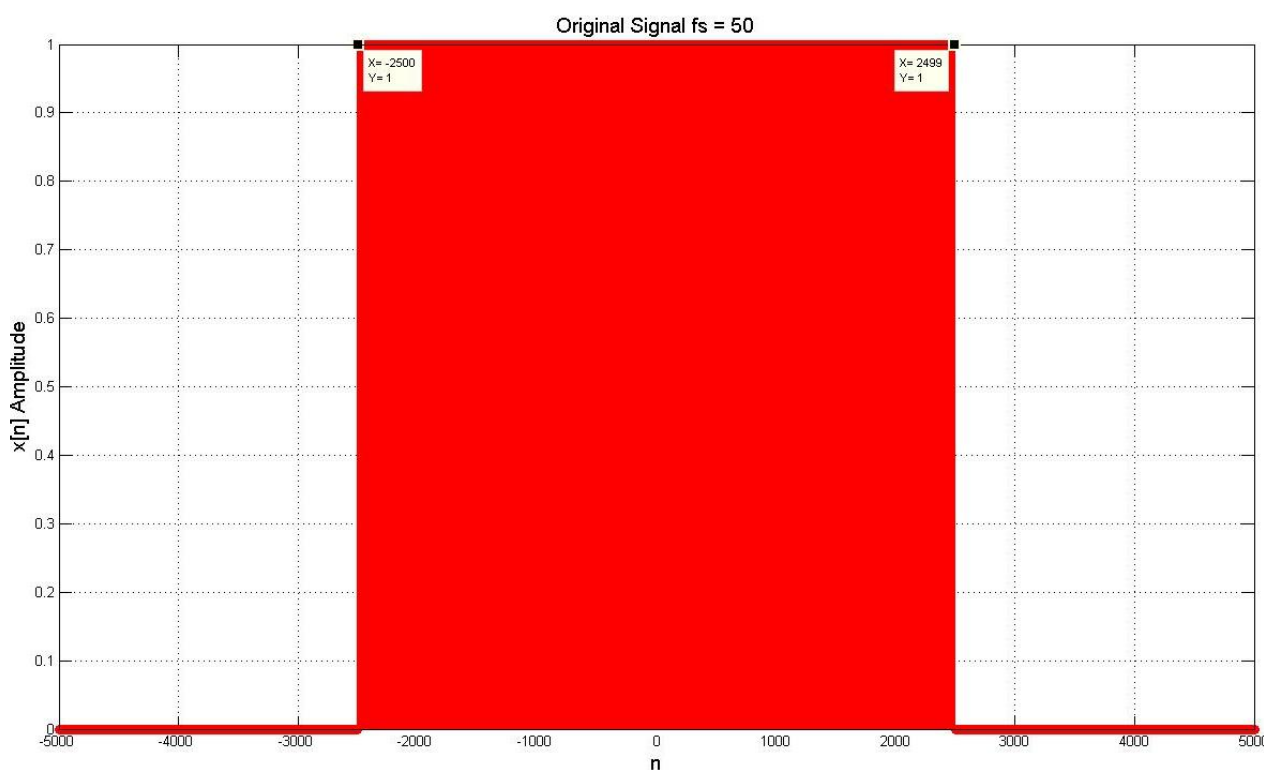


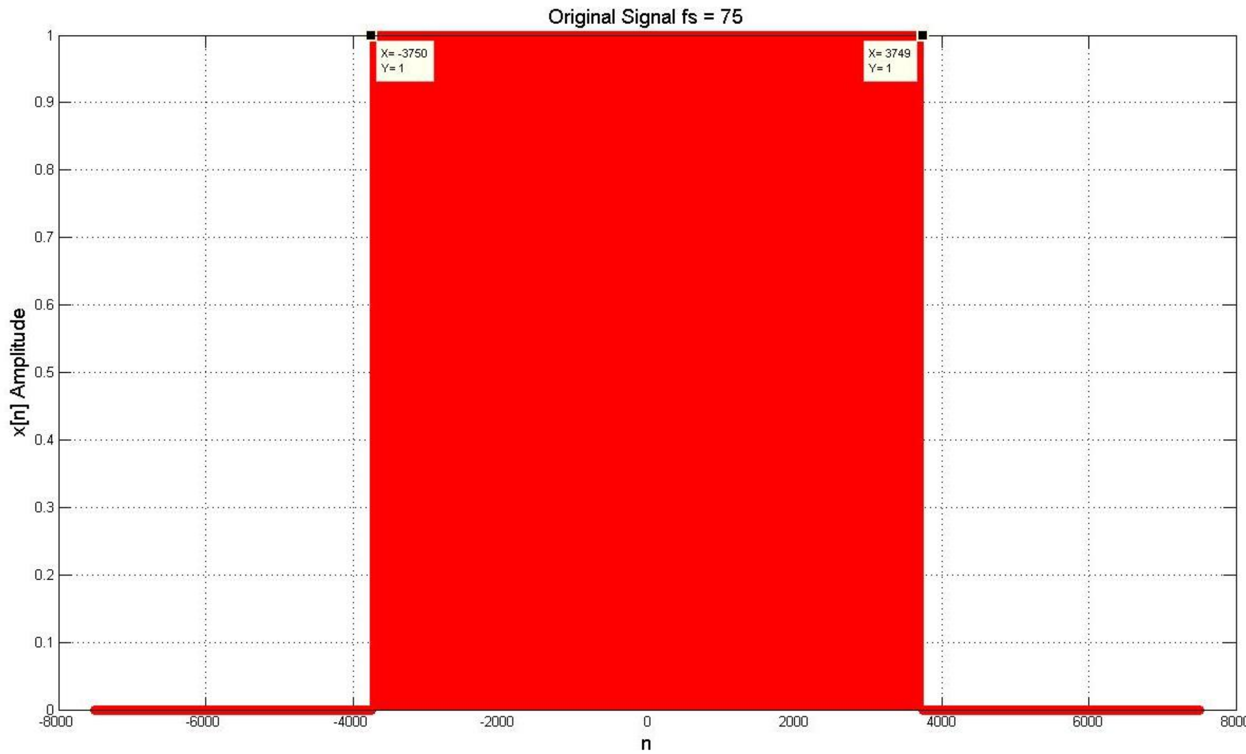


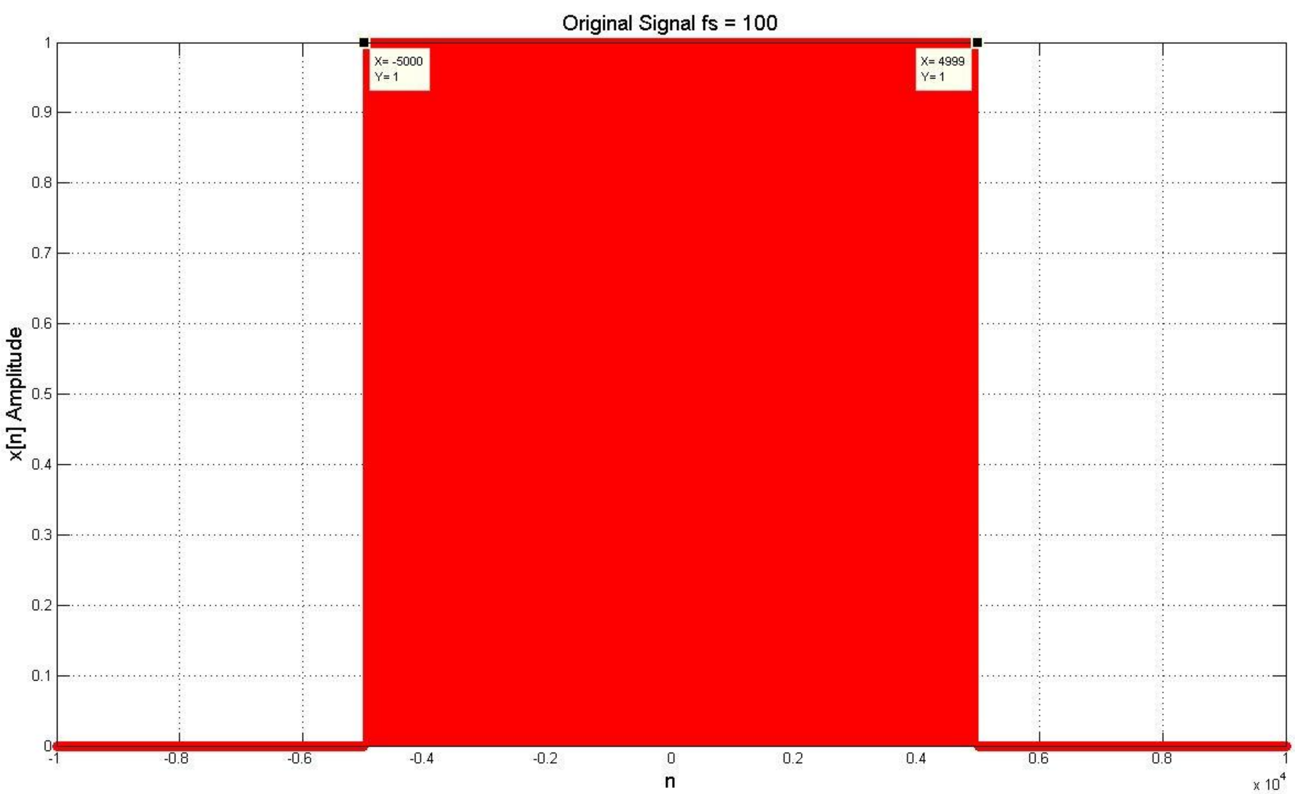

Original Signal fs = 2
x[n] Amplitude
n
X= -100
Y= 1
X= 99
Y= 1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-200
-150
-100
-50
0
50
100
150
200

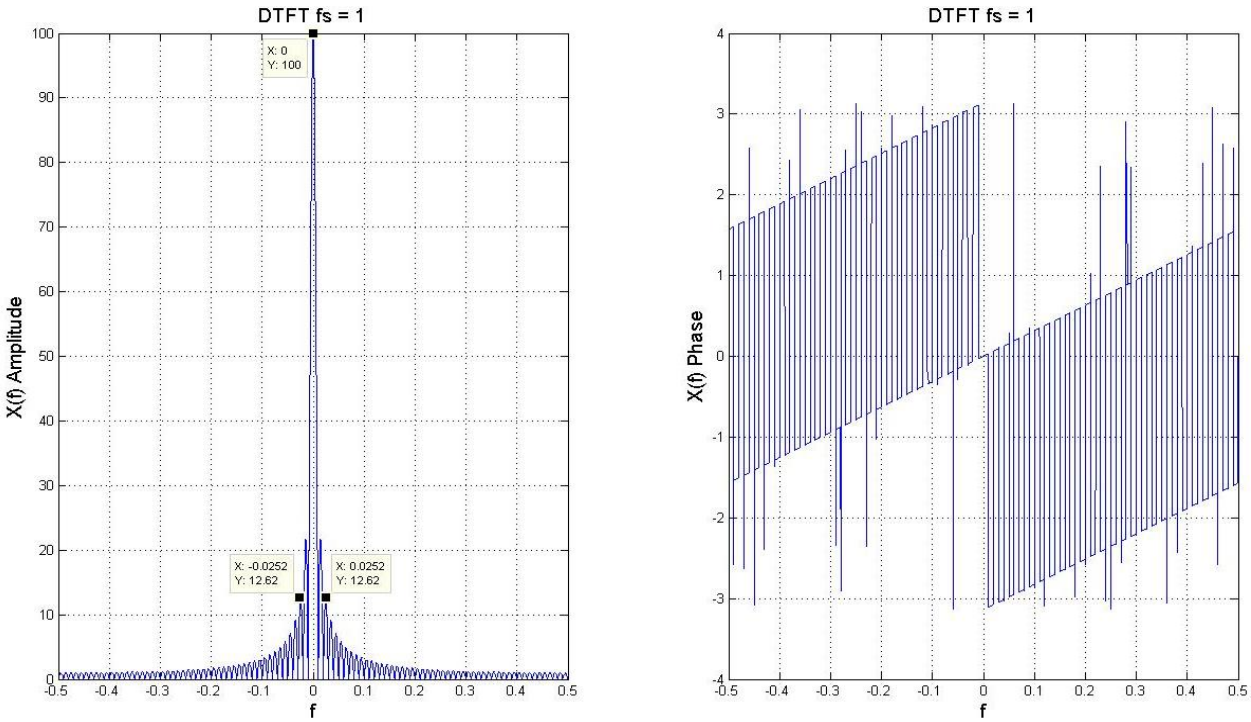

Original Signal fs = 3
x[n] Amplitude
n
X= -150
Y= 1
X= 149
Y= 1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-300
-200
-100
0
100
200
300

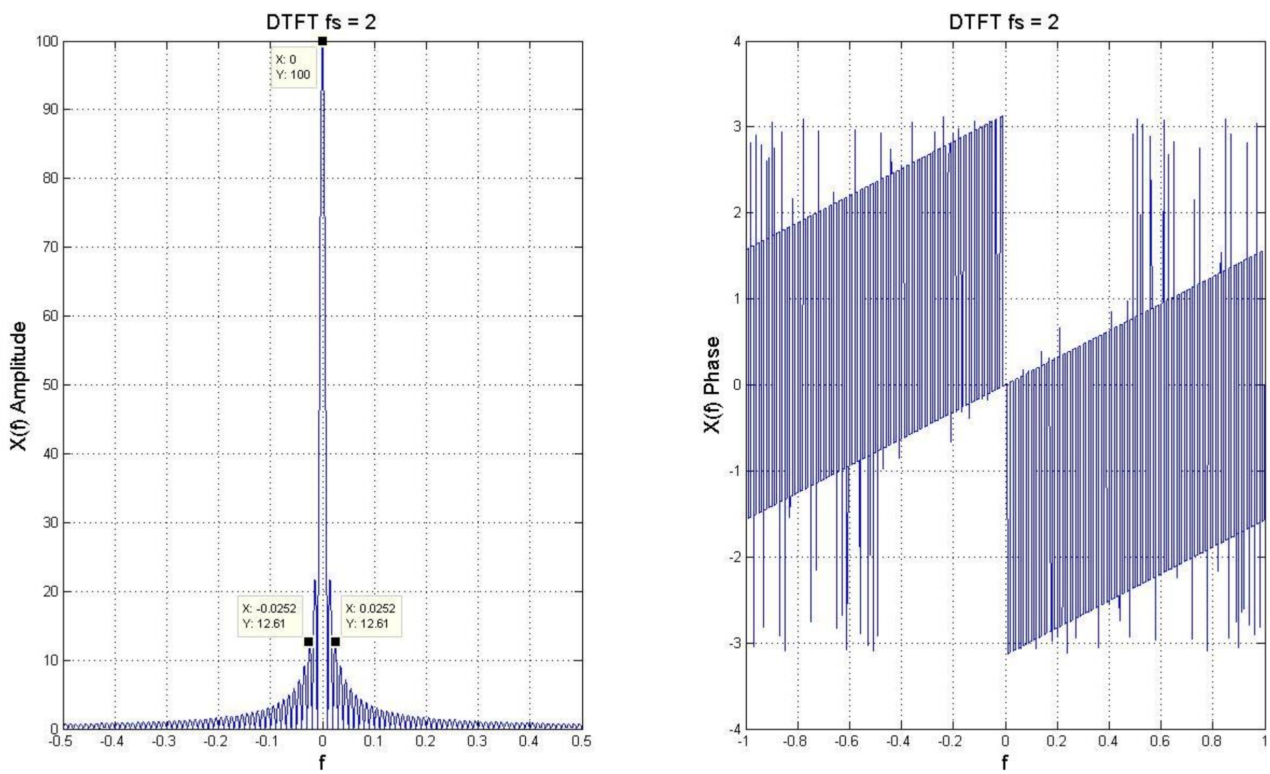

Original Signal fs = 10
X= -500
Y= 1
X= 499
Y= 1
x[n] Amplitude
n

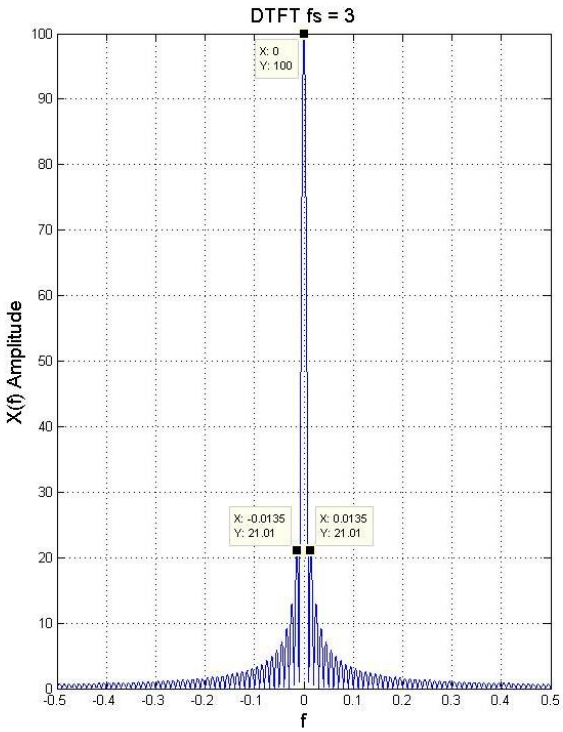

Original Signal fs = 50
X= -2500
Y= 1
X= 2499
Y= 1
x[n] Amplitude
n

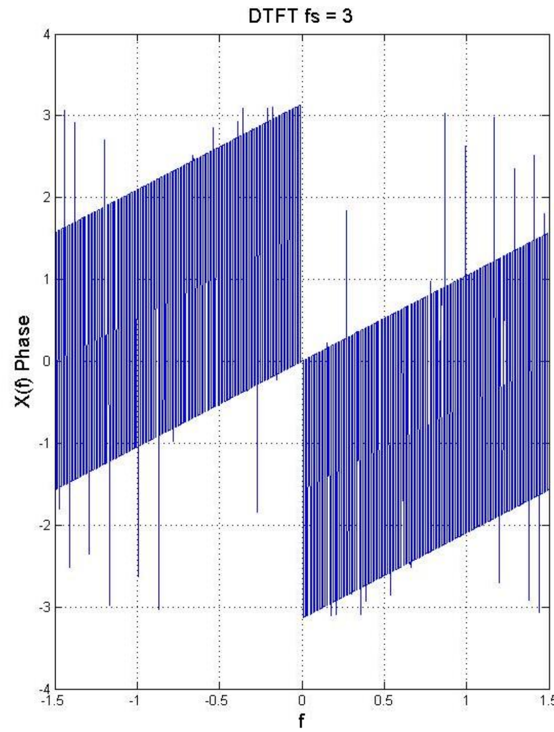

Original Signal fs = 75
X= -3750
Y= 1
X= 3749
Y= 1
x[n] Amplitude
n
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-8000
-6000
-4000
-2000
0
2000
4000
6000
8000

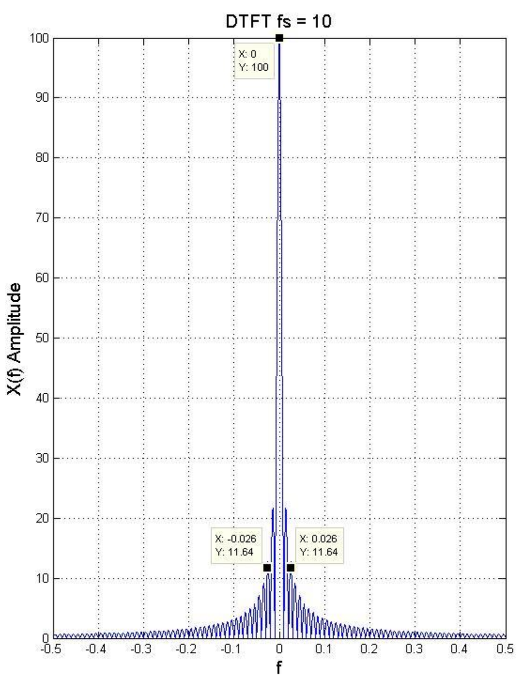

Original Signal fs = 100
X= -5000
Y= 1
X= 4999
Y= 1
x[n] Amplitude
n
x 10^4
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

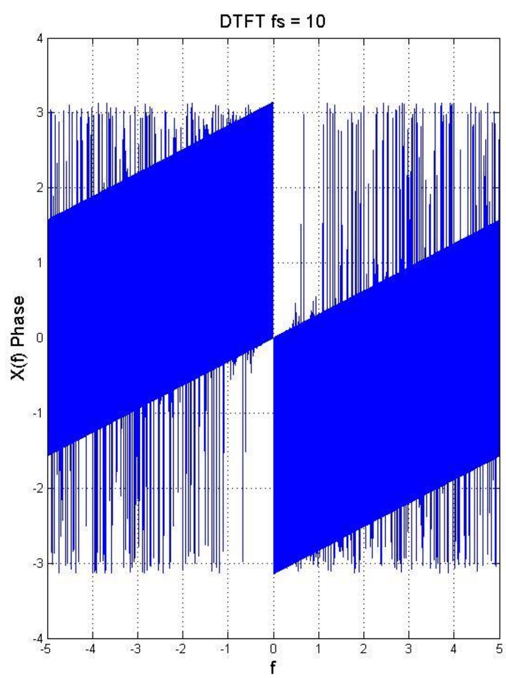

DTFT fs = 1
X: 0
Y: 100
X: -0.0252
Y: 12.62
X: 0.0252
Y: 12.62
X(f) Amplitude
f
DTFT fs = 1
X(f) Phase
f

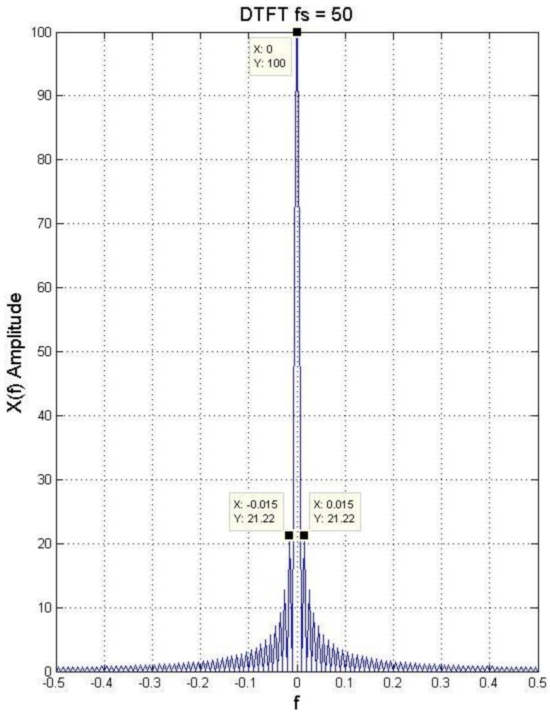

DTFT fs = 2
X: 0
Y: 100
X: -0.0252
Y: 12.61
X: 0.0252
Y: 12.61
X(f) Amplitude
f
DTFT fs = 2
X(f) Phase
f

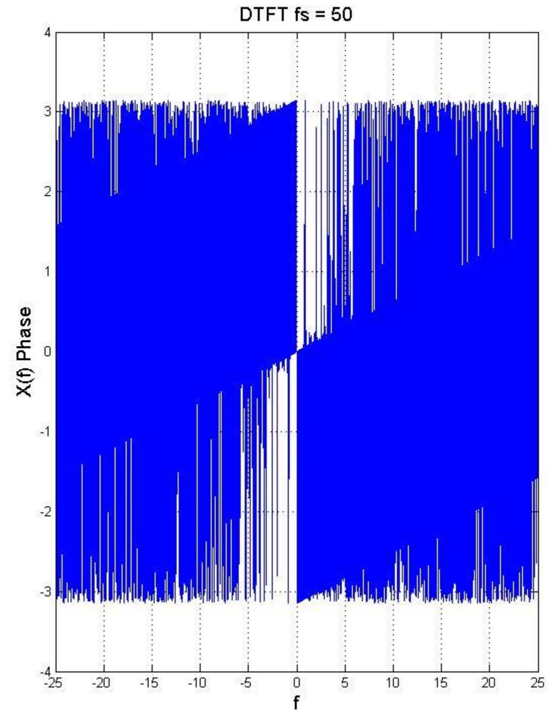

DTFT fs = 3
X(f) Amplitude
f
X: 0
Y: 100
X: -0.0135
Y: 21.01
X: 0.0135
Y: 21.01

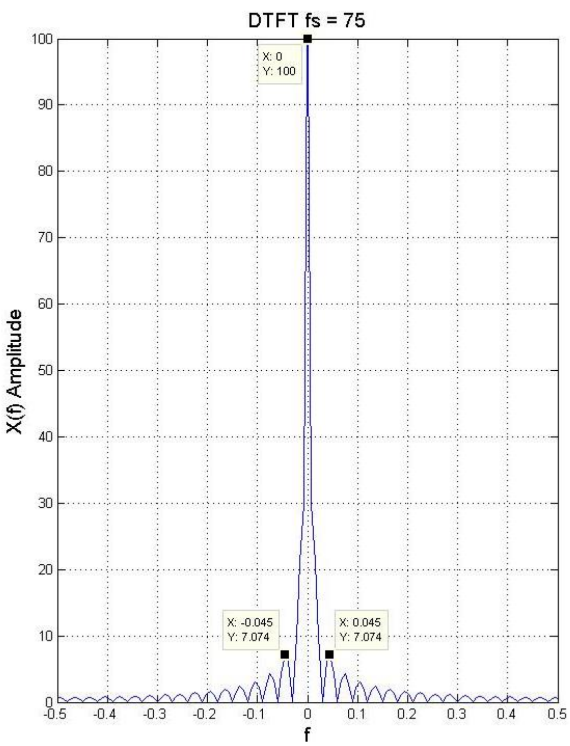

DTFT fs = 3
X(f) Phase
f

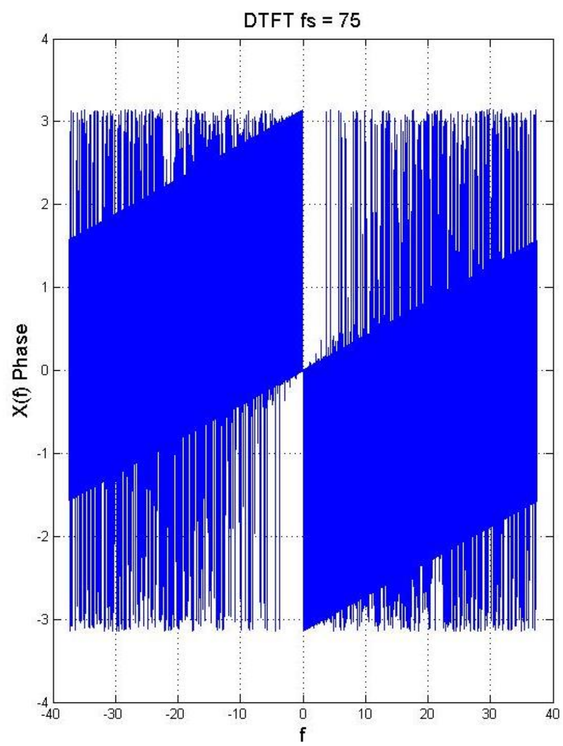

DTFT fs = 10
X(f) Amplitude
f
X: 0
Y: 100
X: -0.026
Y: 11.64
X: 0.026
Y: 11.64

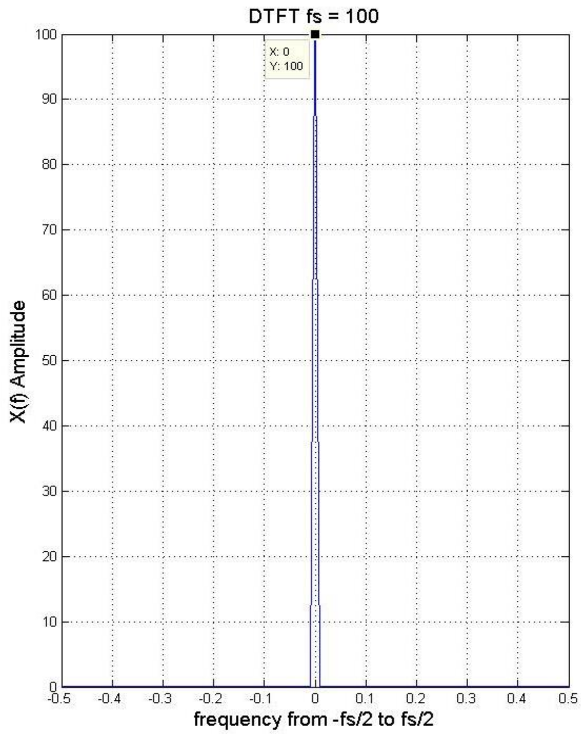

DTFT fs = 10
X(f) Phase
f

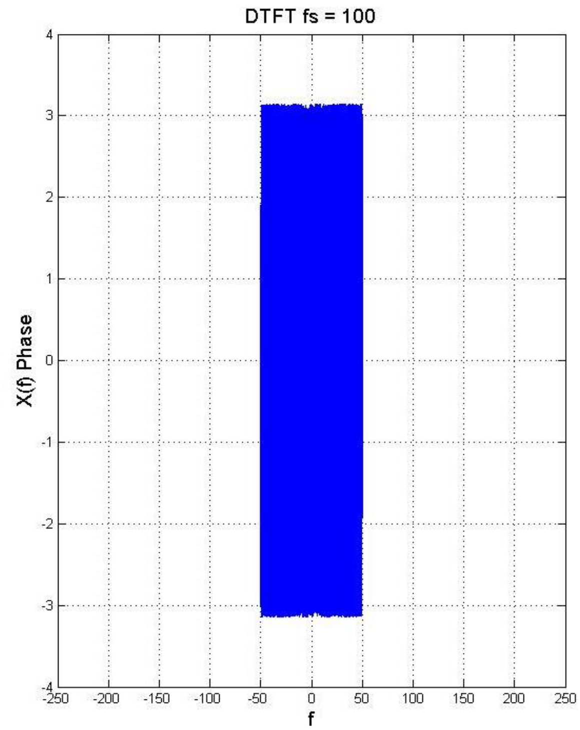

DTFT fs = 50
X(f) Amplitude
f
X: 0
Y: 100
X: -0.015
Y: 21.22
X: 0.015
Y: 21.22

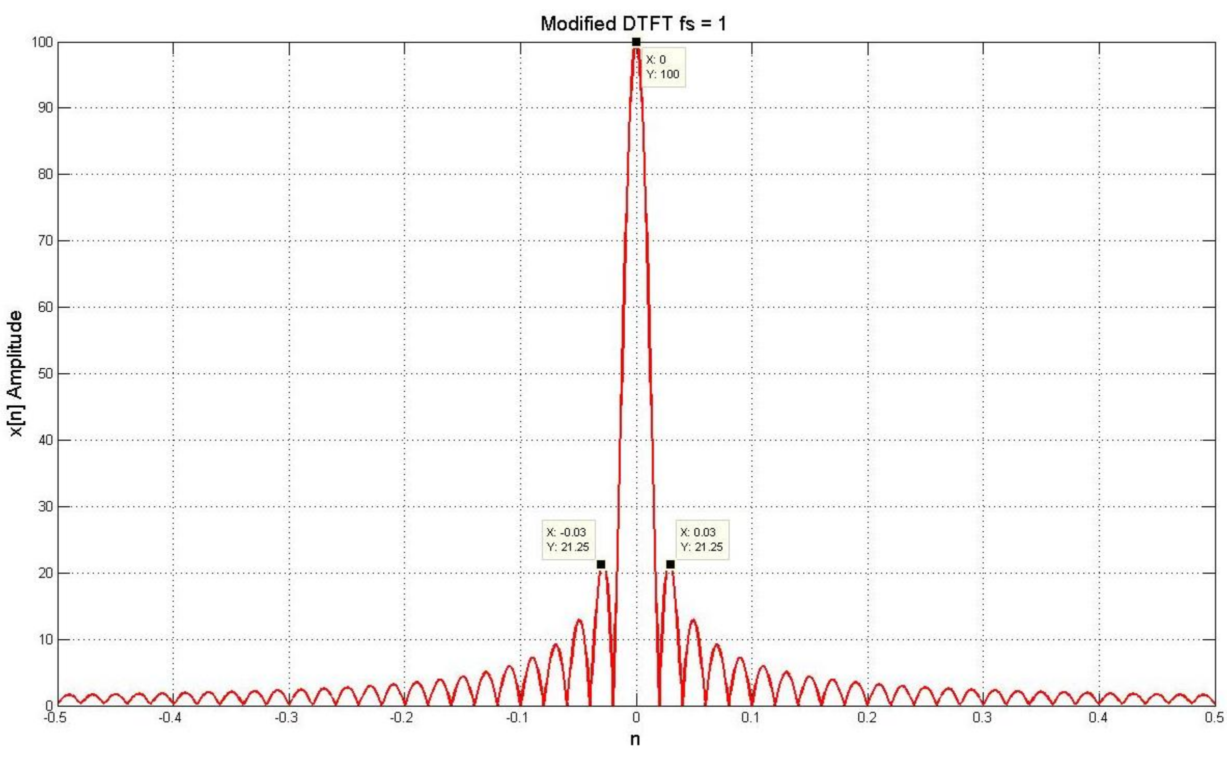

DTFT fs = 50
X(f) Phase
f

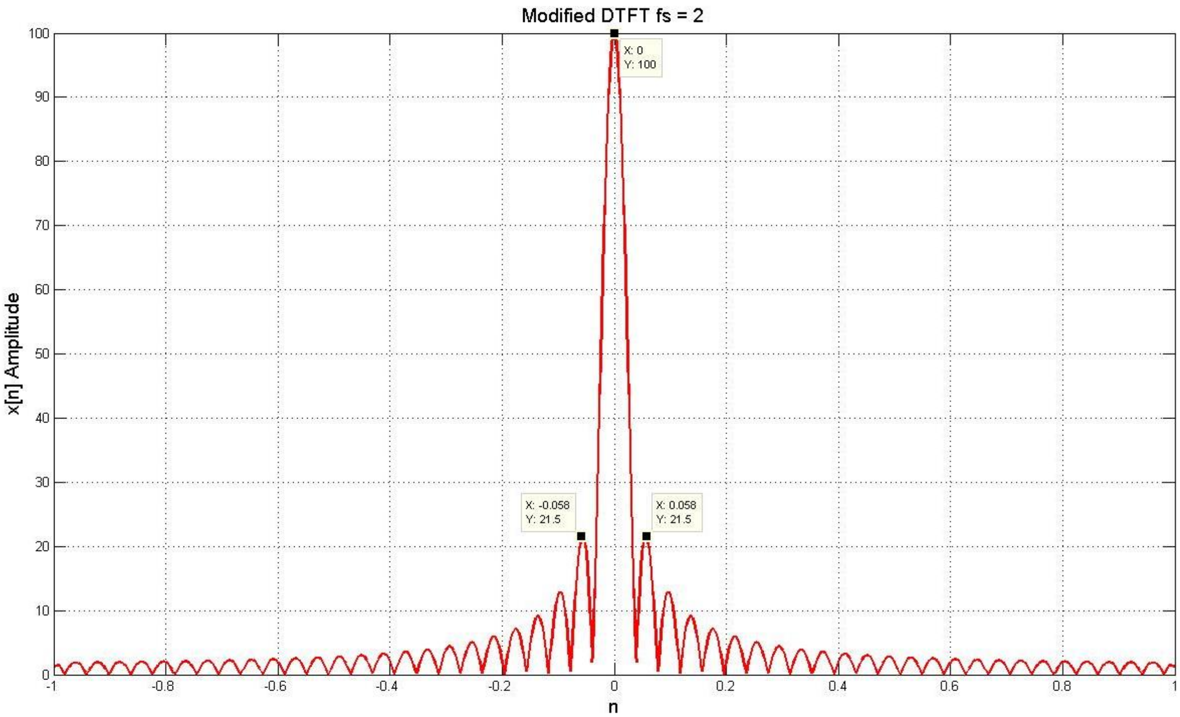

DTFT fs = 75
X(f) Amplitude
f
X: 0
Y: 100
X: -0.045
Y: 7.074
X: 0.045
Y: 7.074

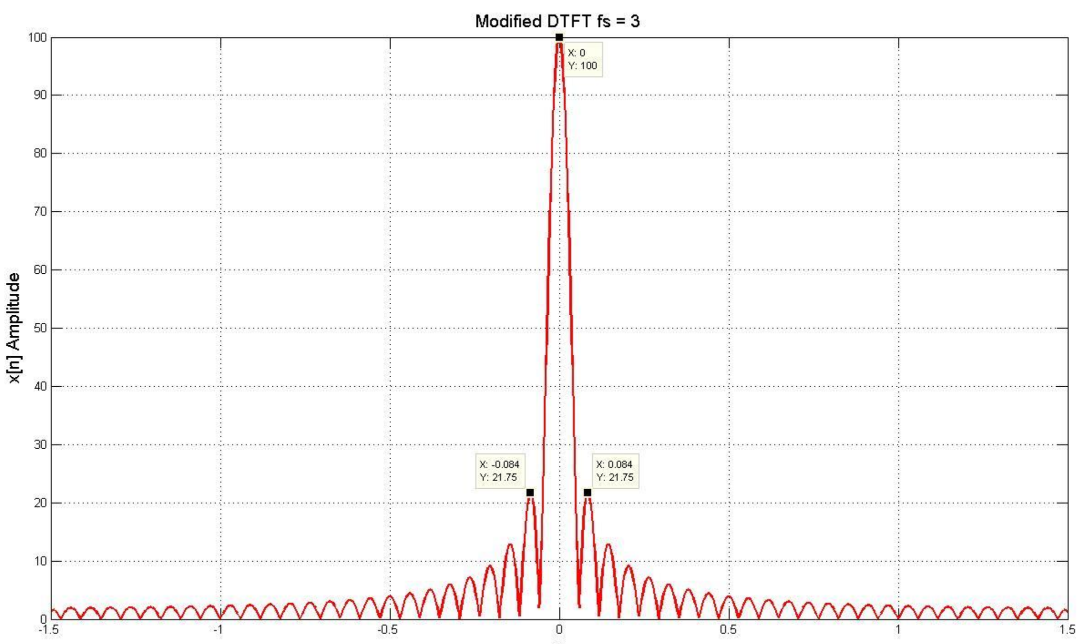

DTFT fs = 75
X(f) Phase
f

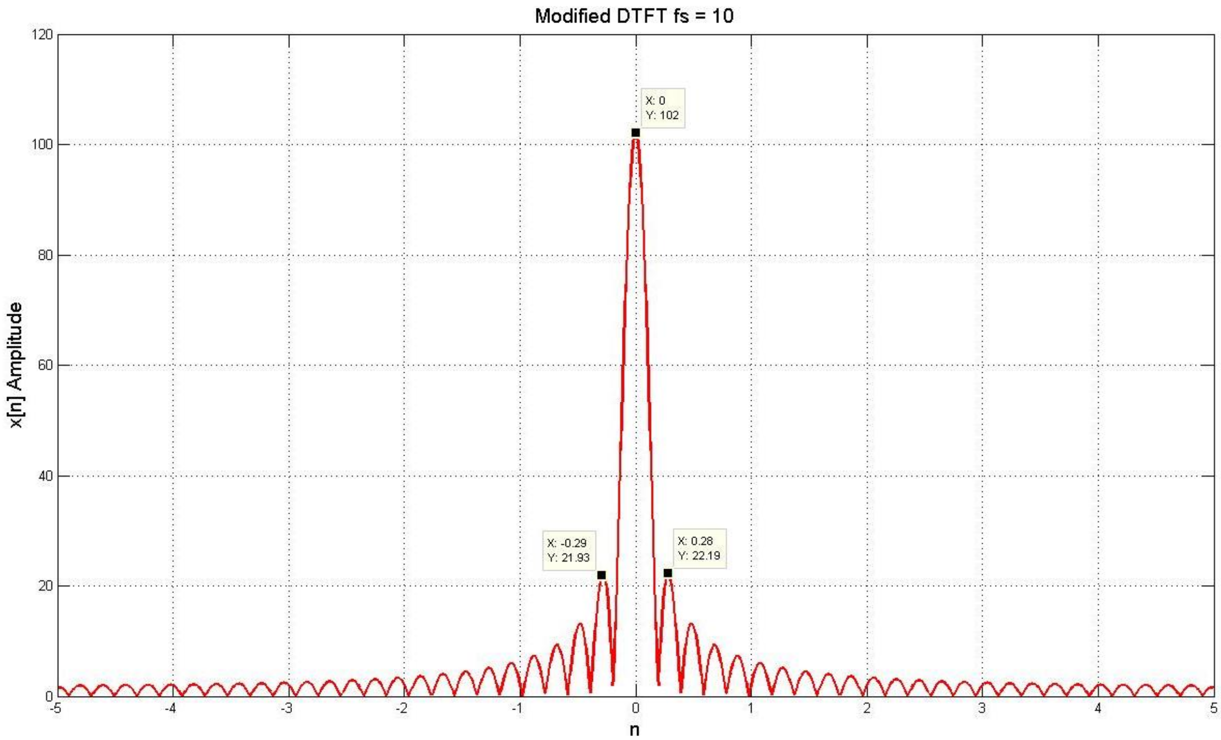

DTFT fs = 100
X: 0
Y: 100
X(f) Amplitude
frequency from -fs/2 to fs/2

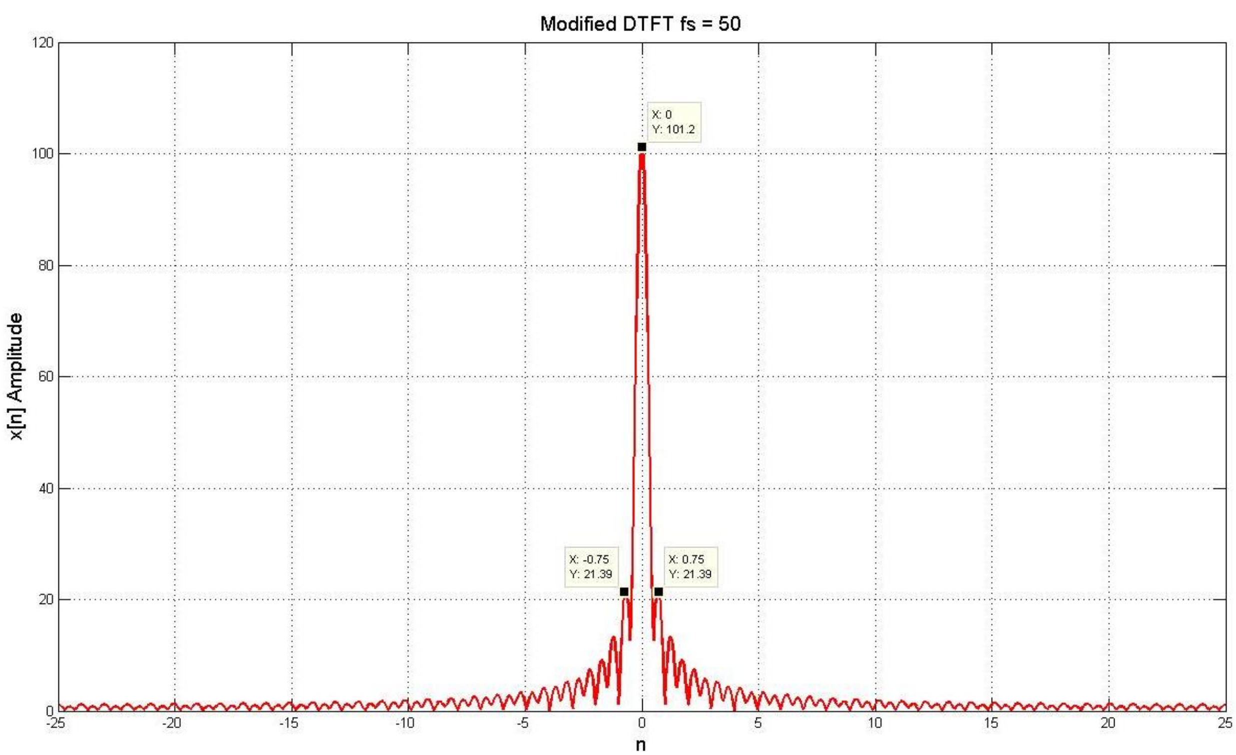

DTFT fs = 100
X(f) Phase
f

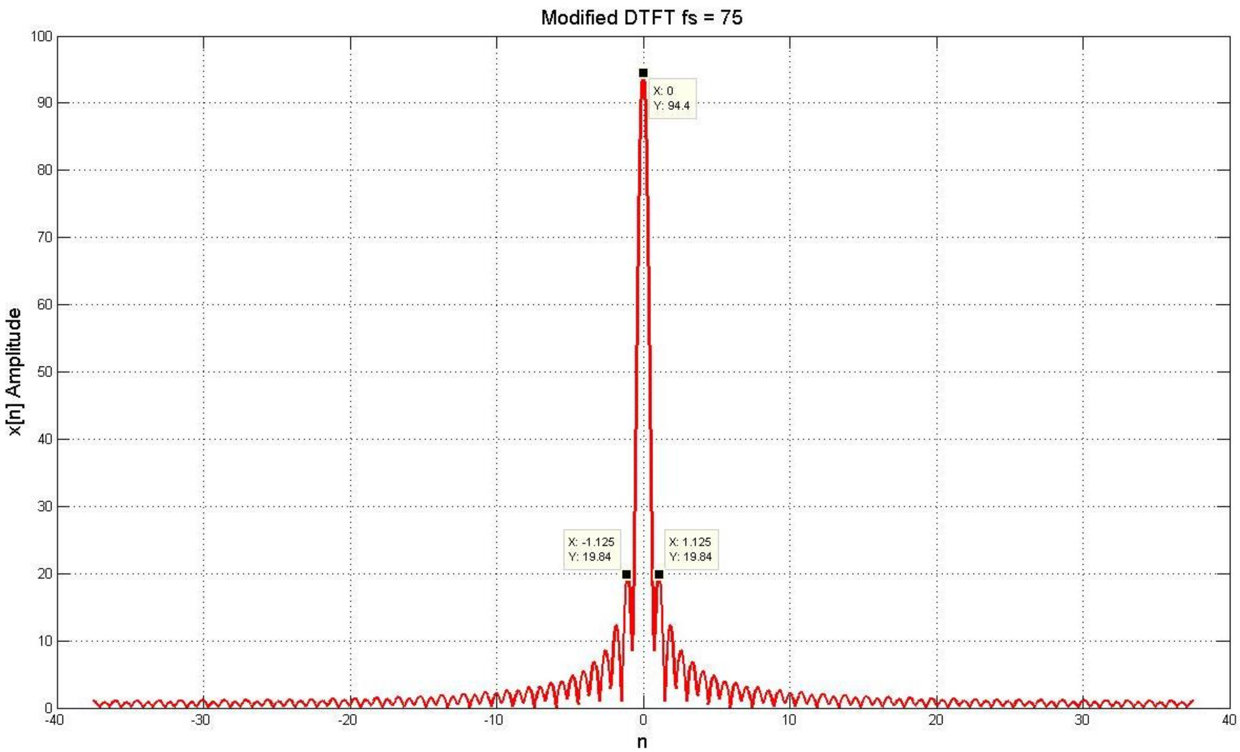

Modified DTFT fs = 1
X: 0
Y: 100
X: -0.03
Y: 21.25
X: 0.03
Y: 21.25
x[n] Amplitude
n

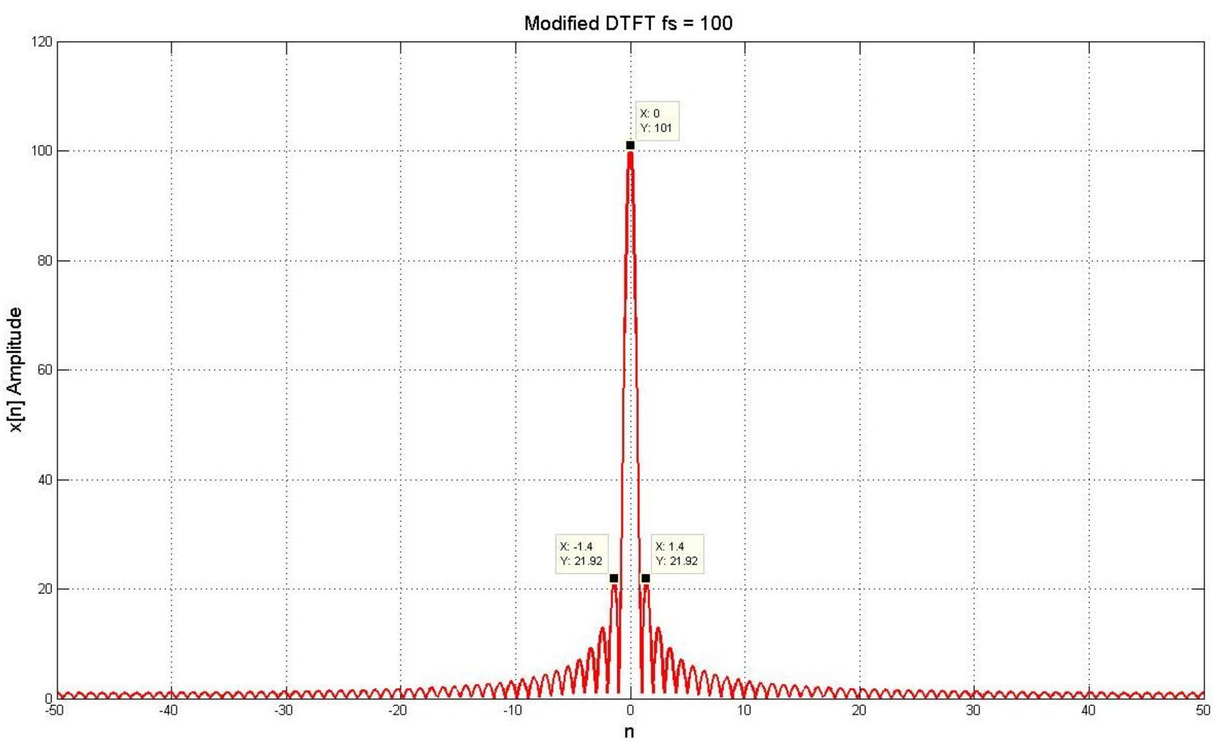

Modified DTFT fs = 2
x[n] Amplitude
n
X: 0
Y: 100
X: -0.058
Y: 21.5
X: 0.058
Y: 21.5

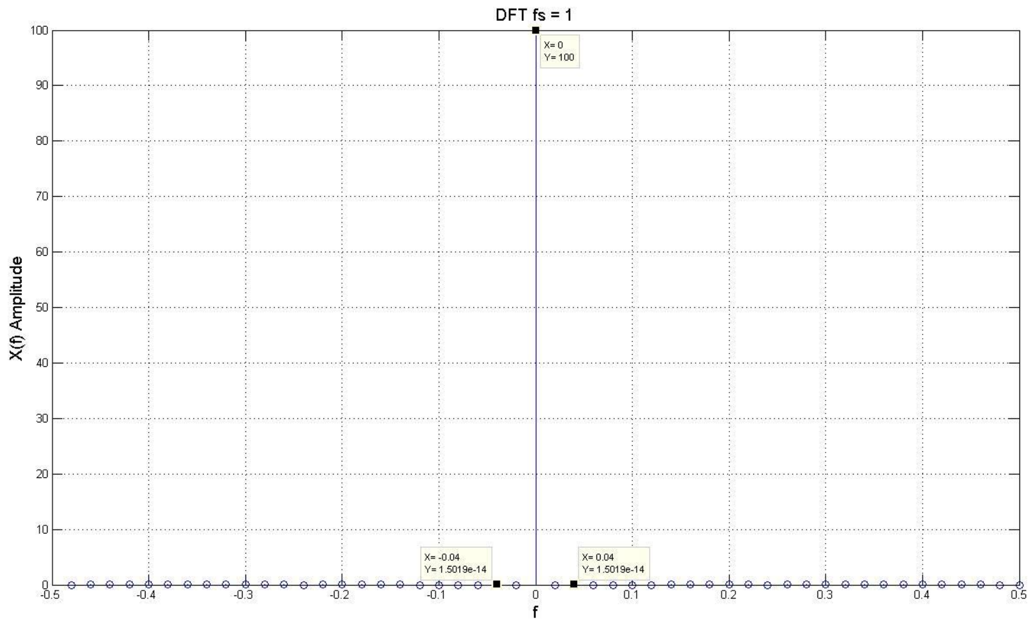

Modified DTFT fs = 3
x[n] Amplitude
n
X: 0
Y: 100
X: -0.084
Y: 21.75
X: 0.084
Y: 21.75

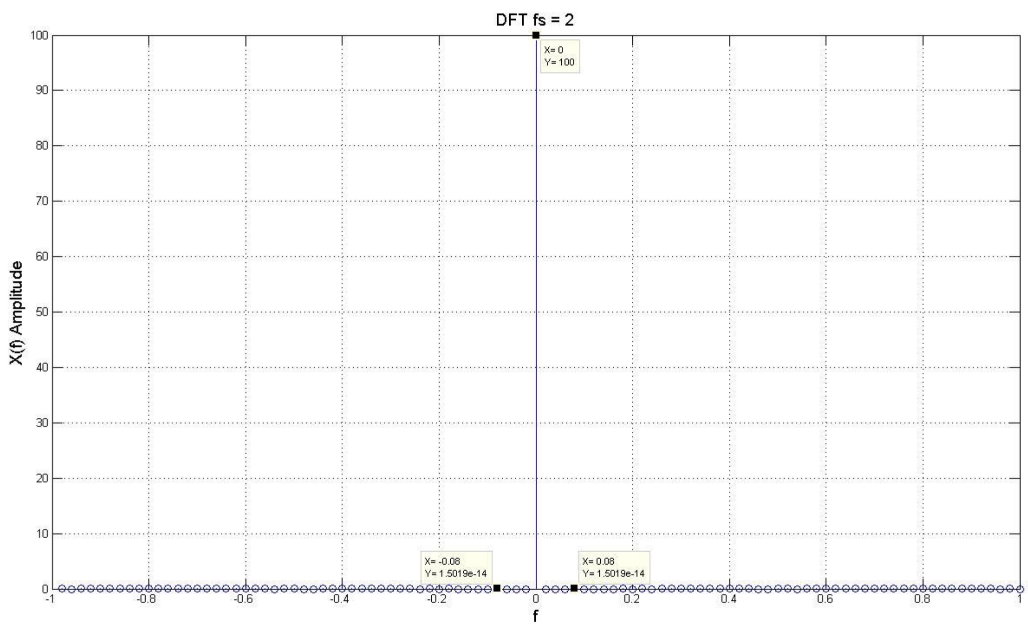

Modified DTFT fs = 10
x[n] Amplitude
n
X: 0
Y: 102
X: -0.29
Y: 21.93
X: 0.28
Y: 22.19

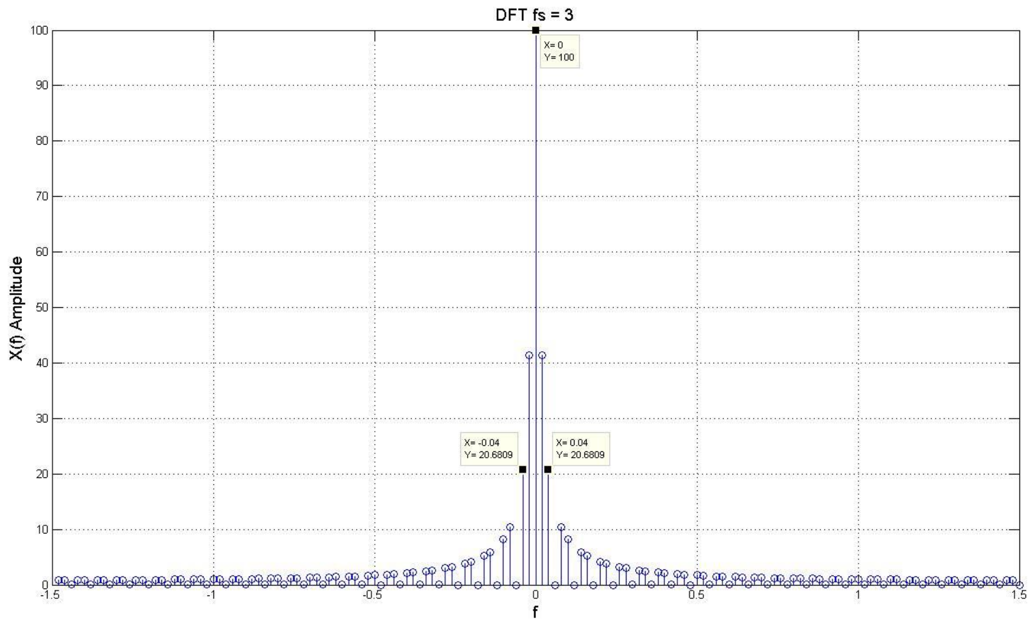

Modified DTFT fs = 50
x[n] Amplitude
n
X: 0
Y: 101.2
X: -0.75
Y: 21.39
X: 0.75
Y: 21.39

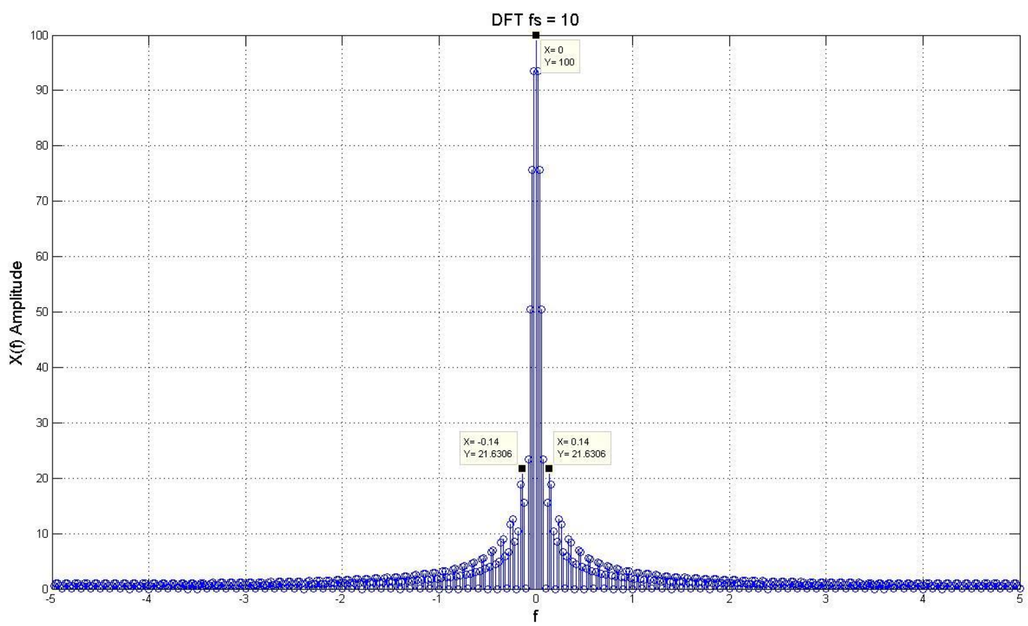

Modified DTFT fs = 75
X: 0
Y: 94.4
X: -1.125
Y: 19.84
X: 1.125
Y: 19.84
x[n] Amplitude
n

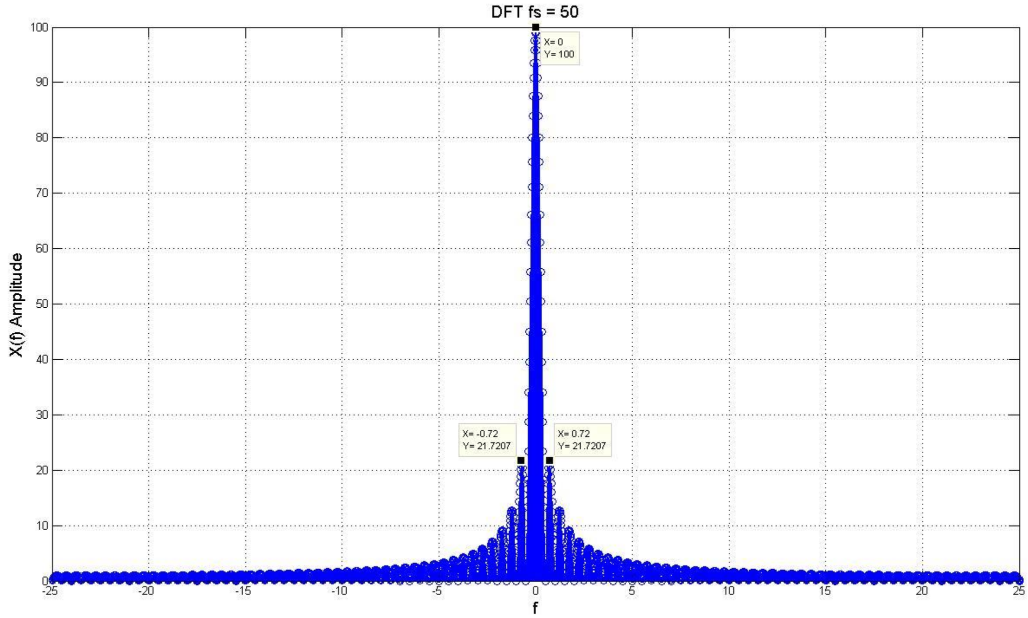

Modified DTFT fs = 100
X: 0
Y: 101
X: -1.4
Y: 21.92
X: 1.4
Y: 21.92
x[n] Amplitude
n

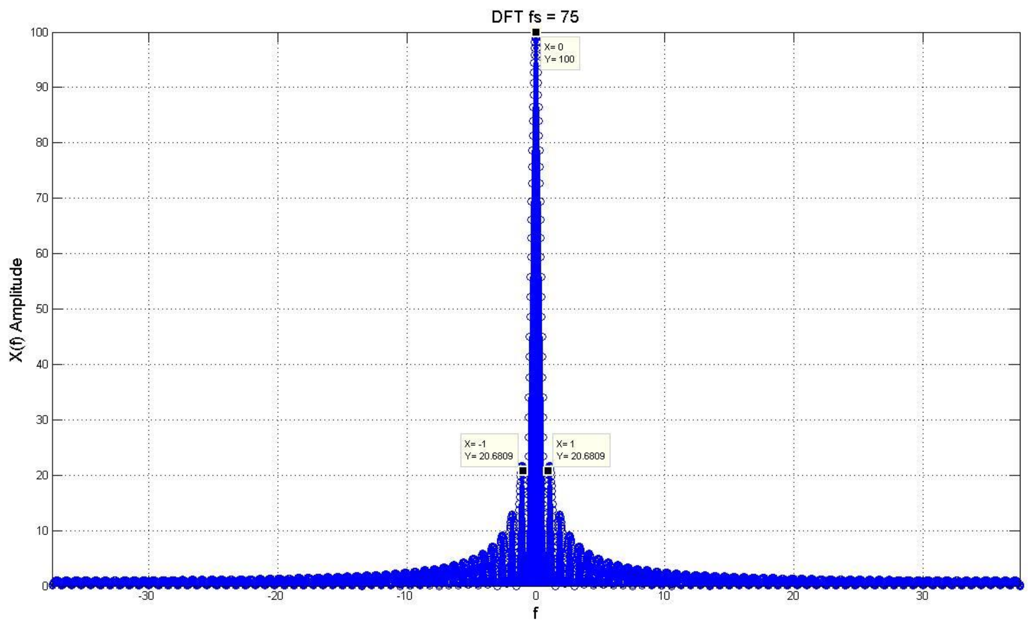

DFT fs = 1
X(f) Amplitude
f
X= 0
Y= 100
X= -0.04
Y= 1.5019e-14
X= 0.04
Y= 1.5019e-14

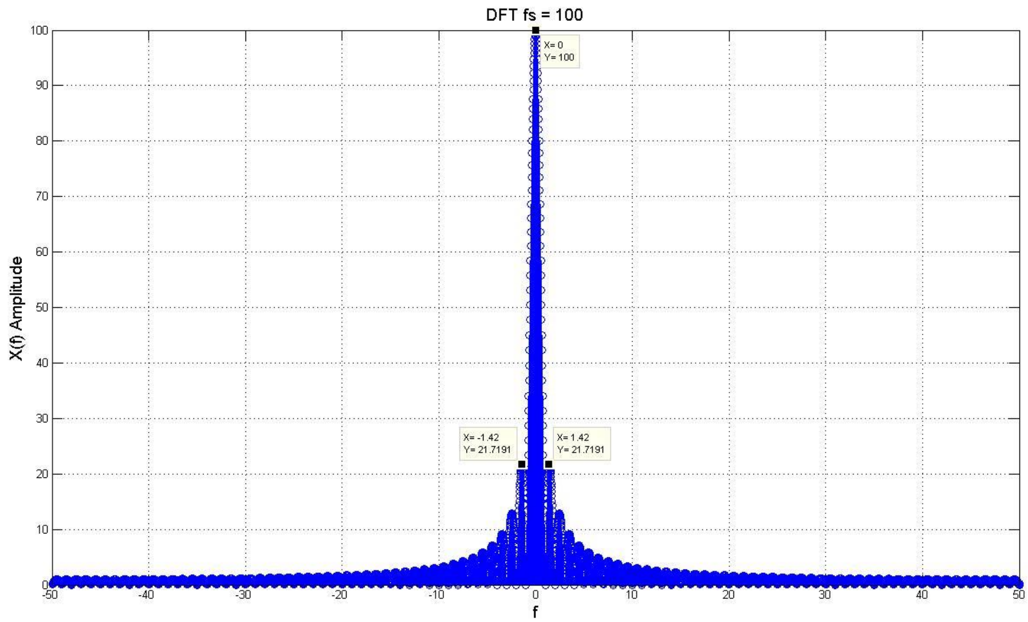

DFT fs = 2
X(f) Amplitude
f
X= 0
Y= 100
X= -0.08
Y= 1.5019e-14
X= 0.08
Y= 1.5019e-14

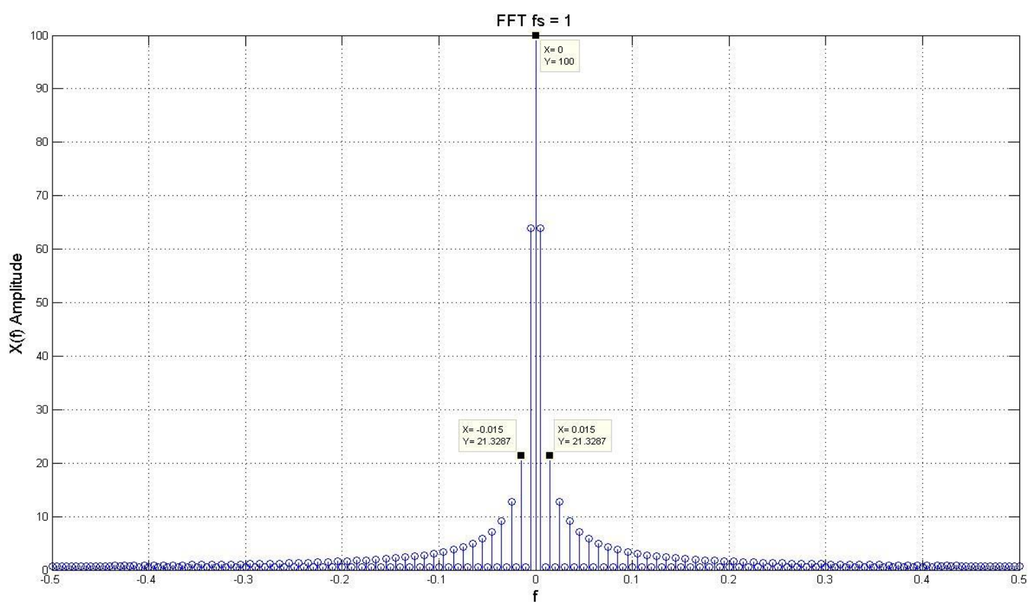

DFT fs = 3
X= 0
Y= 100
X= -0.04
Y= 20.6809
X= 0.04
Y= 20.6809
X(f) Amplitude
f

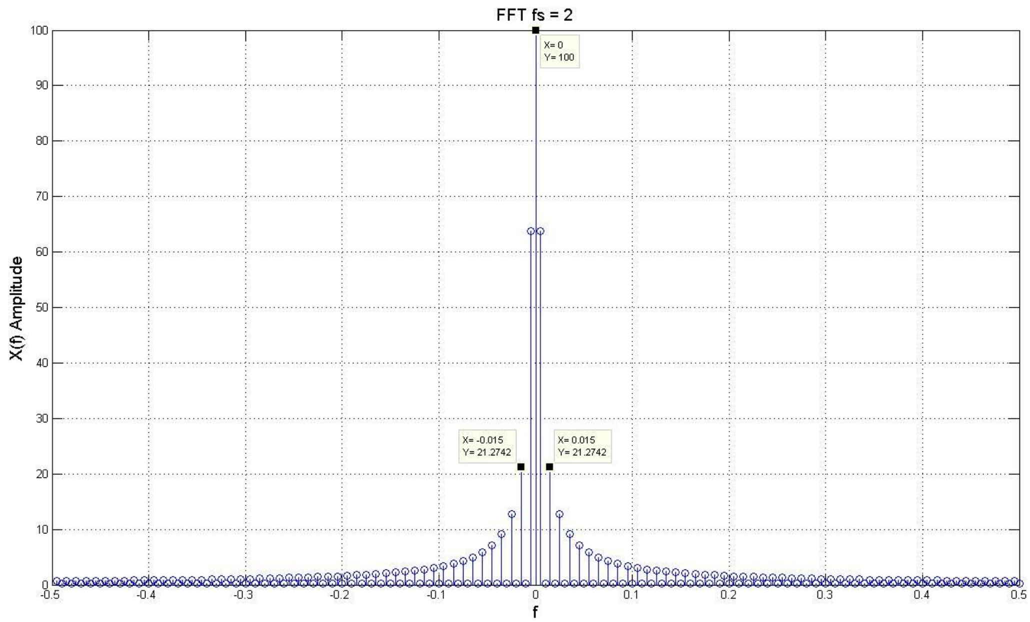

DFT fs = 10
X= 0
Y= 100
X= -0.14
Y= 21.6306
X= 0.14
Y= 21.6306
X(f) Amplitude
f

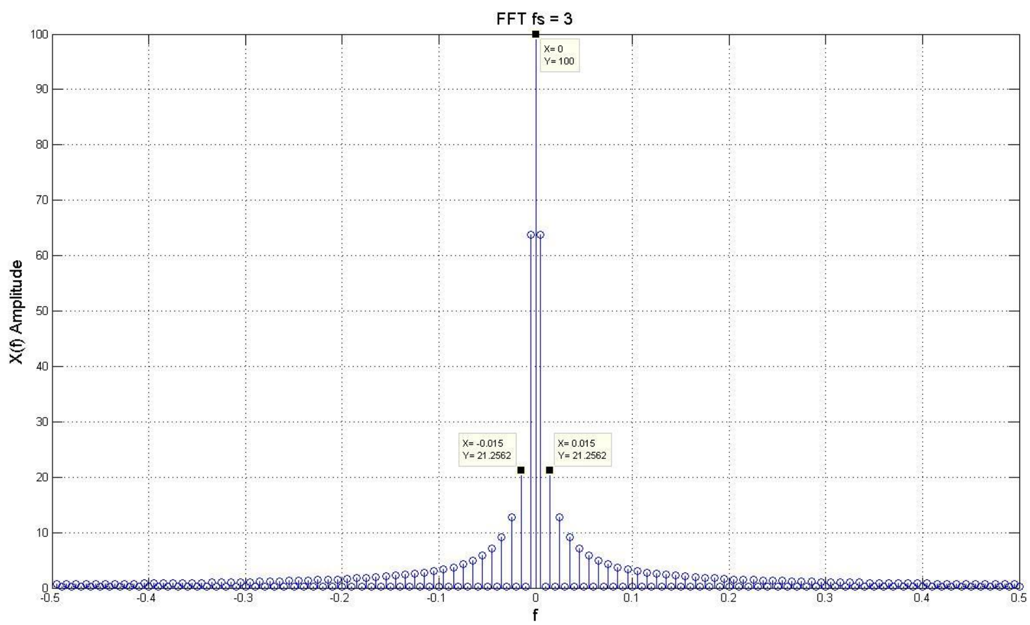

DFT fs = 50
X= 0
Y= 100
X= -0.72
Y= 21.7207
X= 0.72
Y= 21.7207
X(f) Amplitude
f

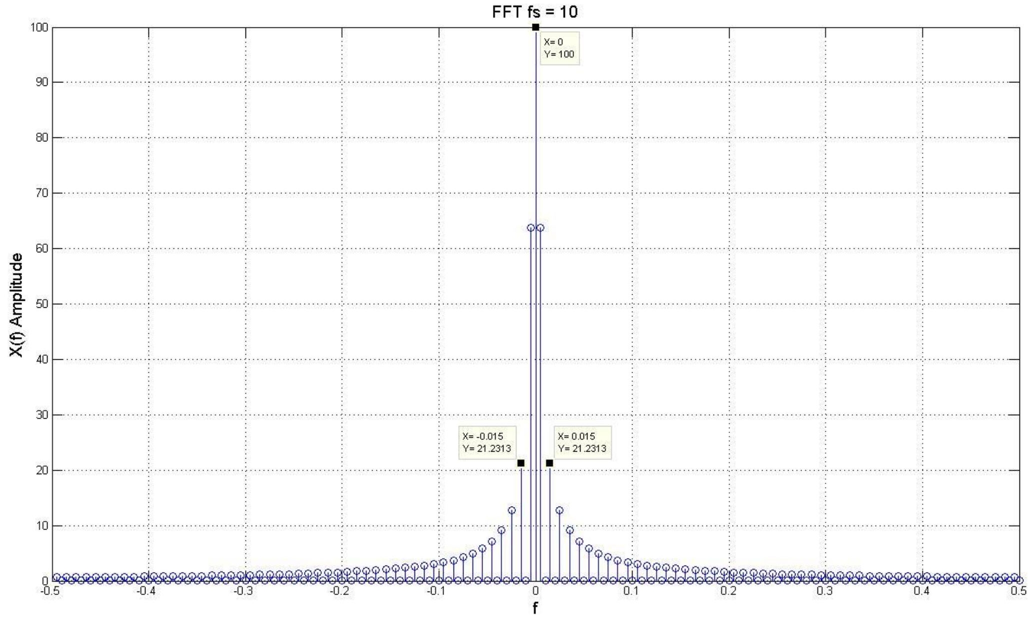

DFT fs = 75
X= 0
Y= 100
X= -1
Y= 20.6809
X= 1
Y= 20.6809
X(f) Amplitude
f

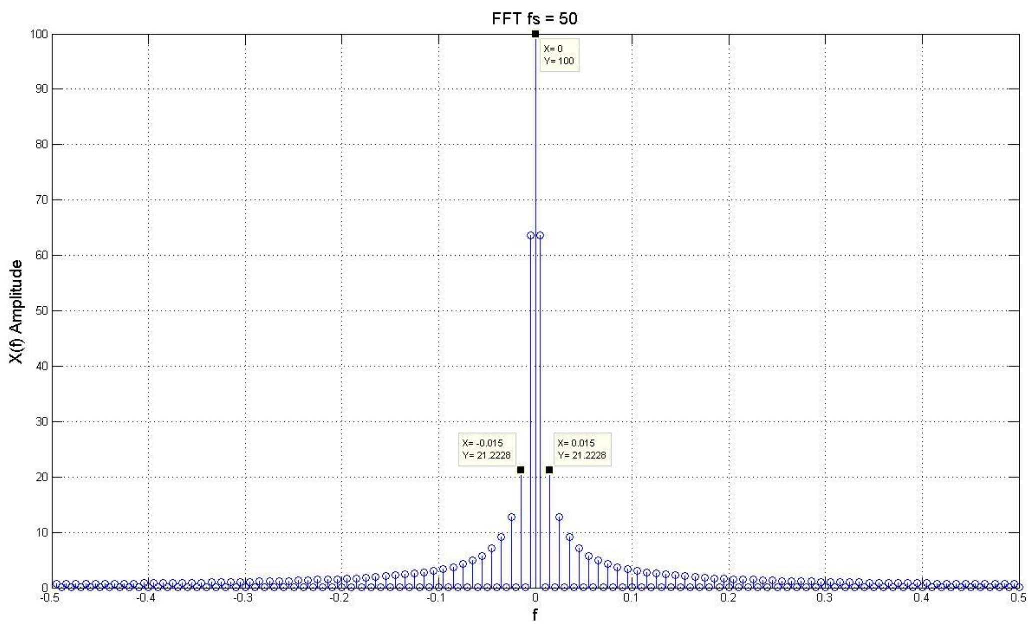

DFT fs = 100
X= 0
Y= 100
X= -1.42
Y= 21.7191
X= 1.42
Y= 21.7191
X(f) Amplitude
f

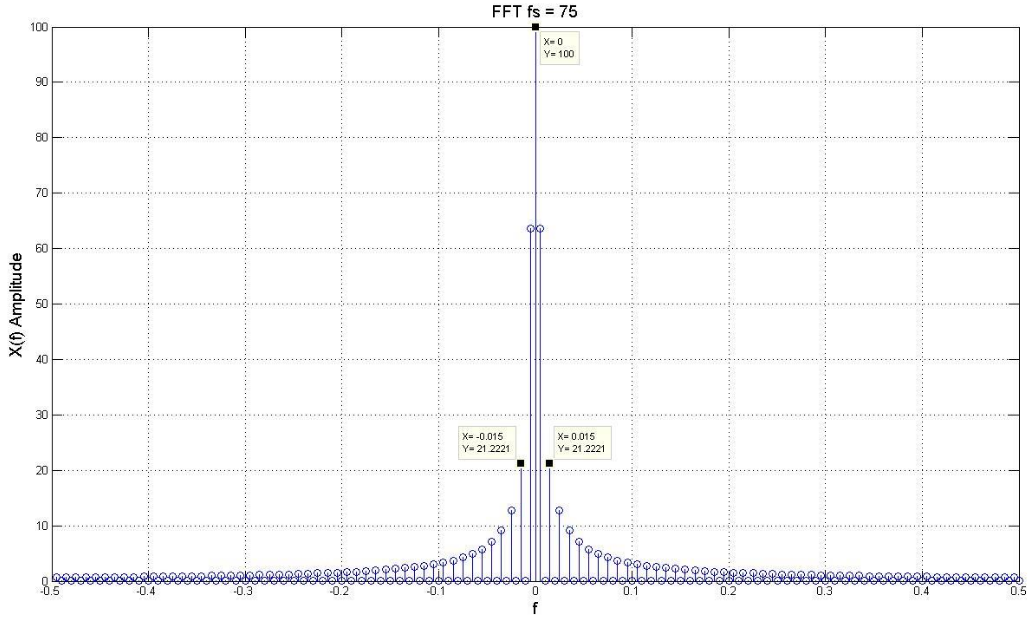

FFT fs = 1
X= 0
Y= 100
X= -0.015
Y= 21.3287
X= 0.015
Y= 21.3287
X(f) Amplitude
f

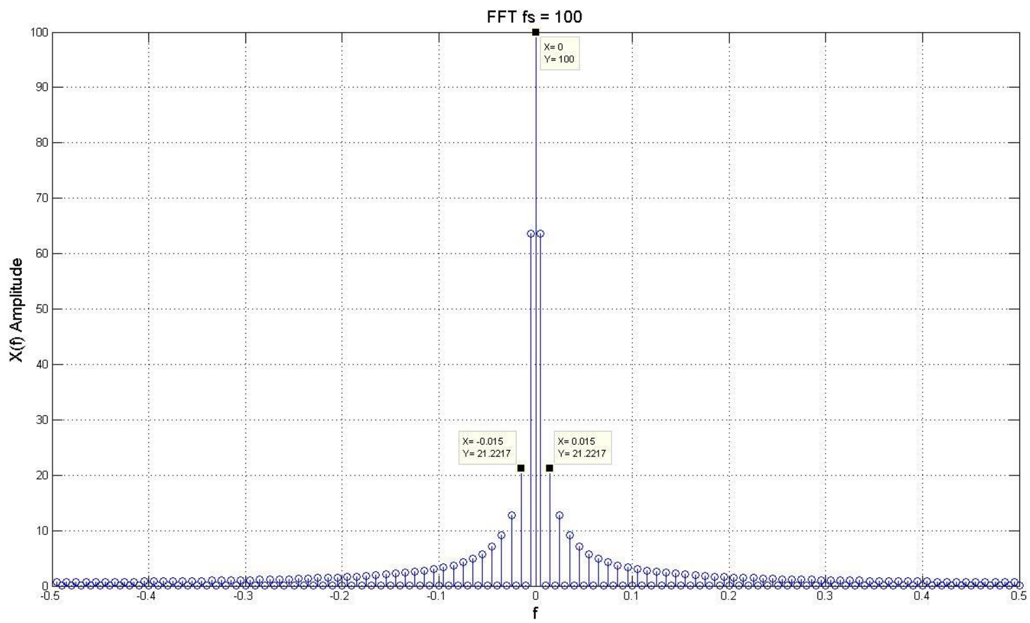

FFT fs = 2
X(f) Amplitude
f
X= 0
Y= 100
X= -0.015
Y= 21.2742
X= 0.015
Y= 21.2742


FFT fs = 3
X(f) Amplitude
f
X= 0
Y= 100
X= -0.015
Y= 21.2562
X= 0.015
Y= 21.2562

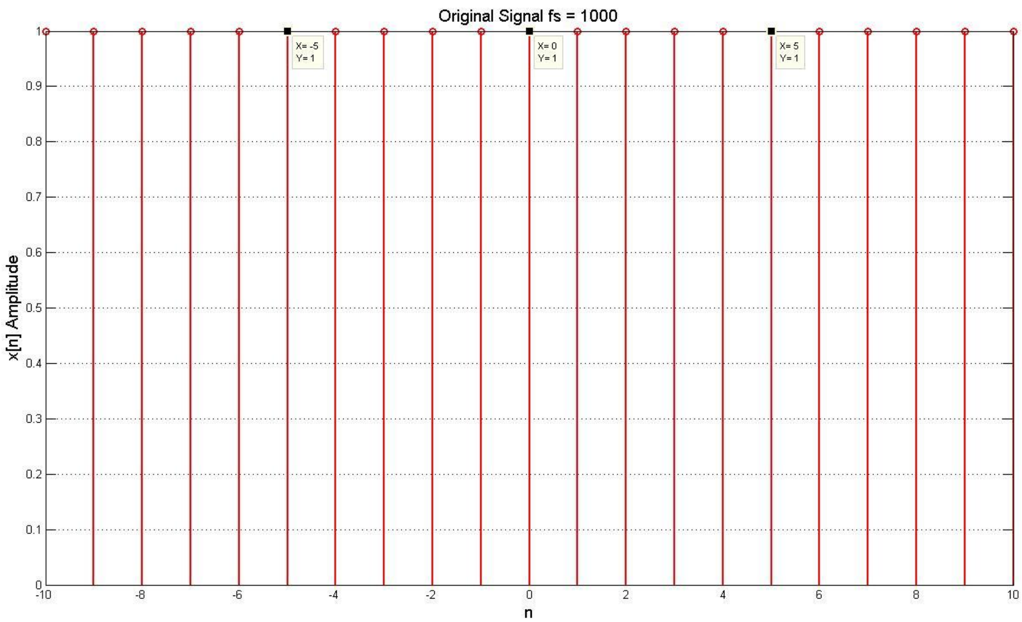

FFT fs = 10
X= 0
Y= 100
X= -0.015
Y= 21.2313
X= 0.015
Y= 21.2313
X(f) Amplitude
f

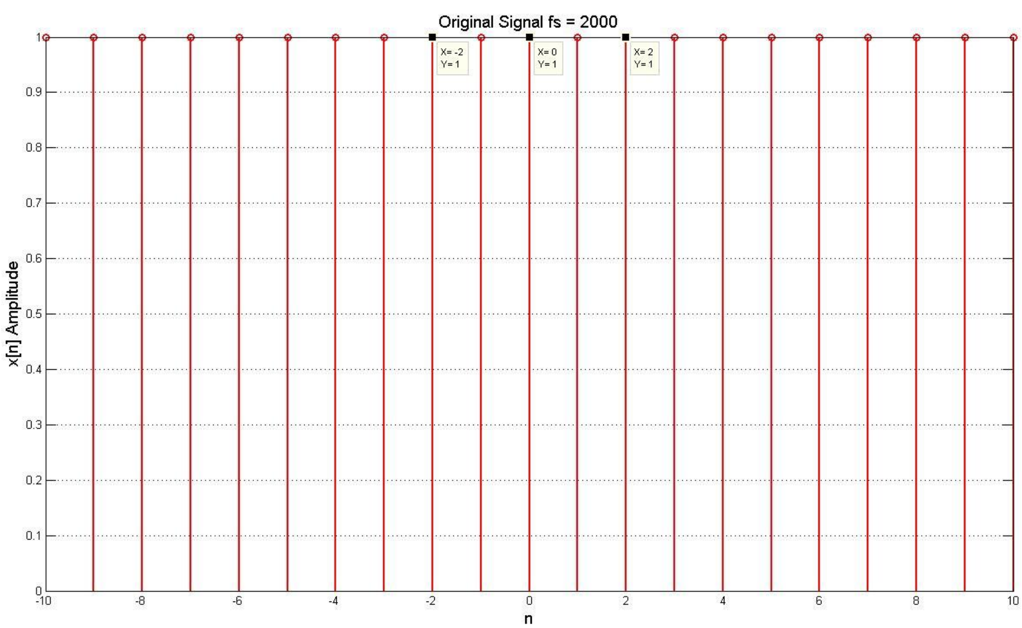

FFT fs = 50
X= 0
Y= 100
X= -0.015
Y= 21.2228
X= 0.015
Y= 21.2228
X(f) Amplitude
f

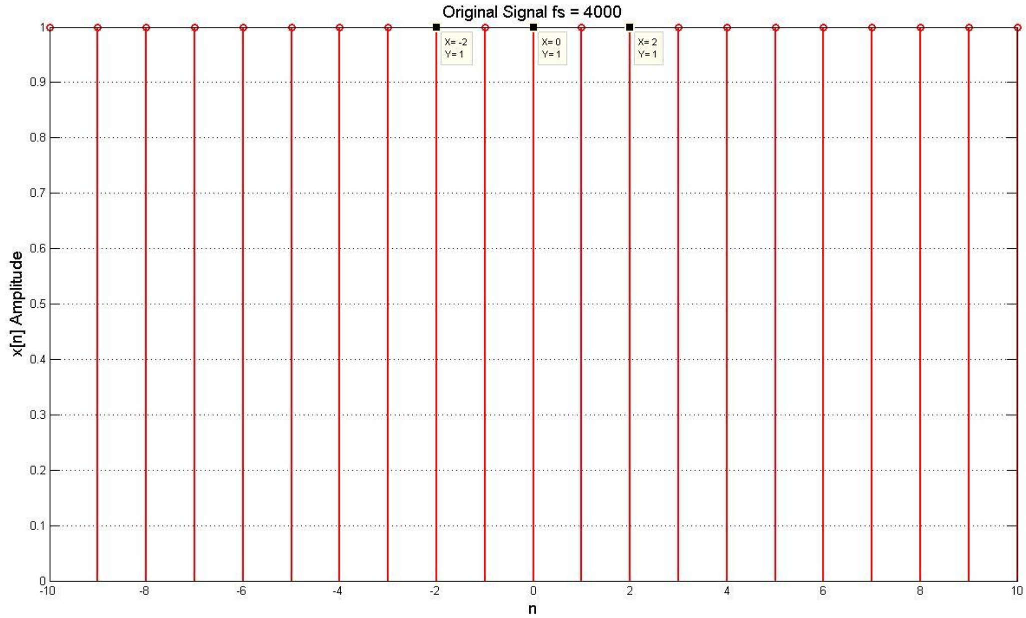

FFT fs = 75
X= 0
Y= 100
X= -0.015
Y= 21.2221
X= 0.015
Y= 21.2221
X(f) Amplitude
f

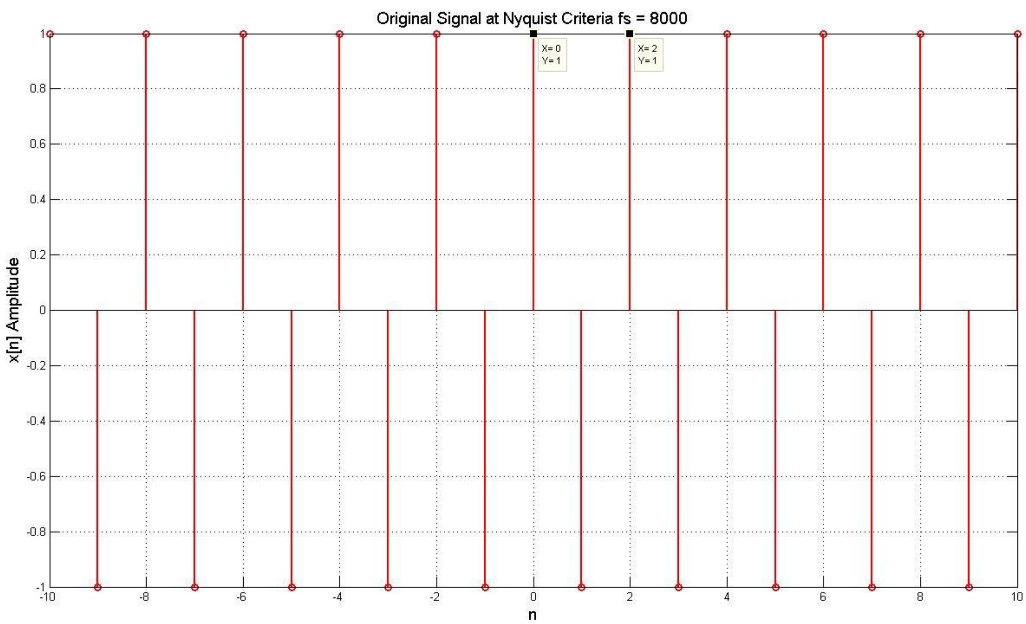

FFT fs = 100
X= 0
Y= 100
X= -0.015
Y= 21.2217
X= 0.015
Y= 21.2217
X(f) Amplitude
f

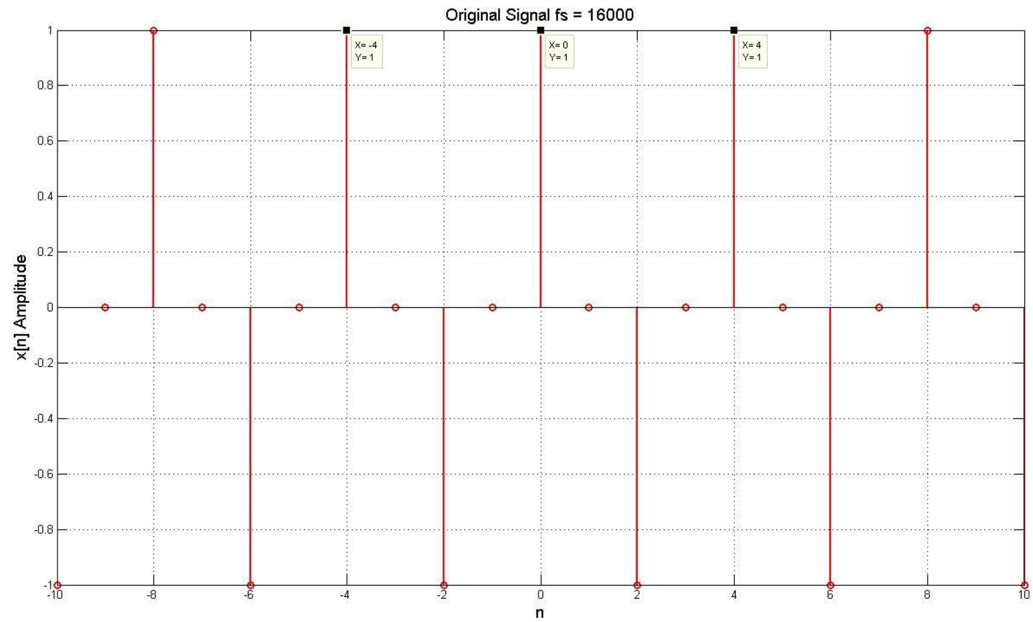
Cosine Function Graphs:

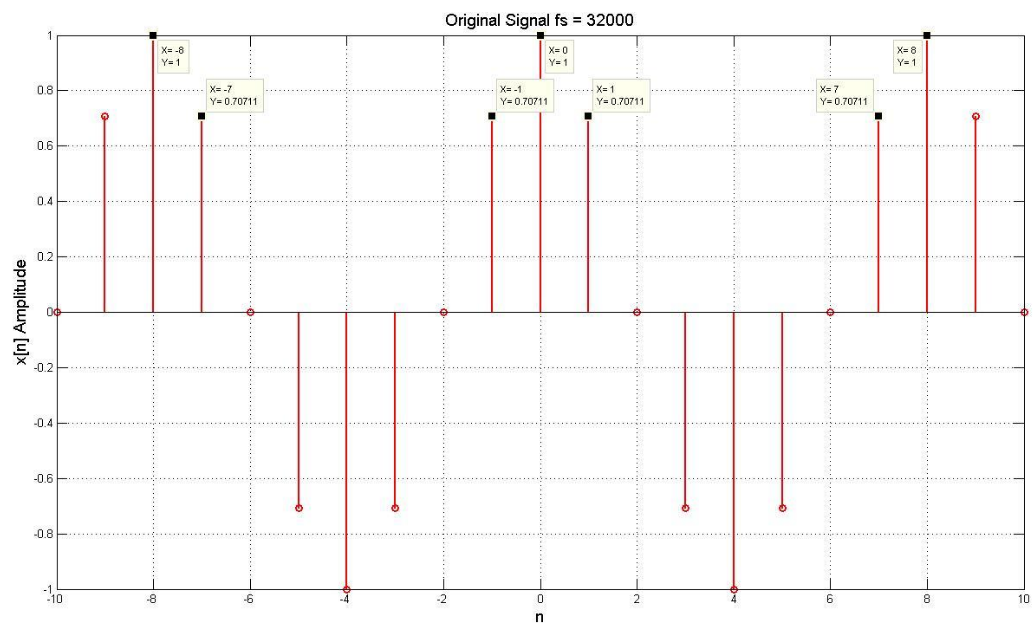


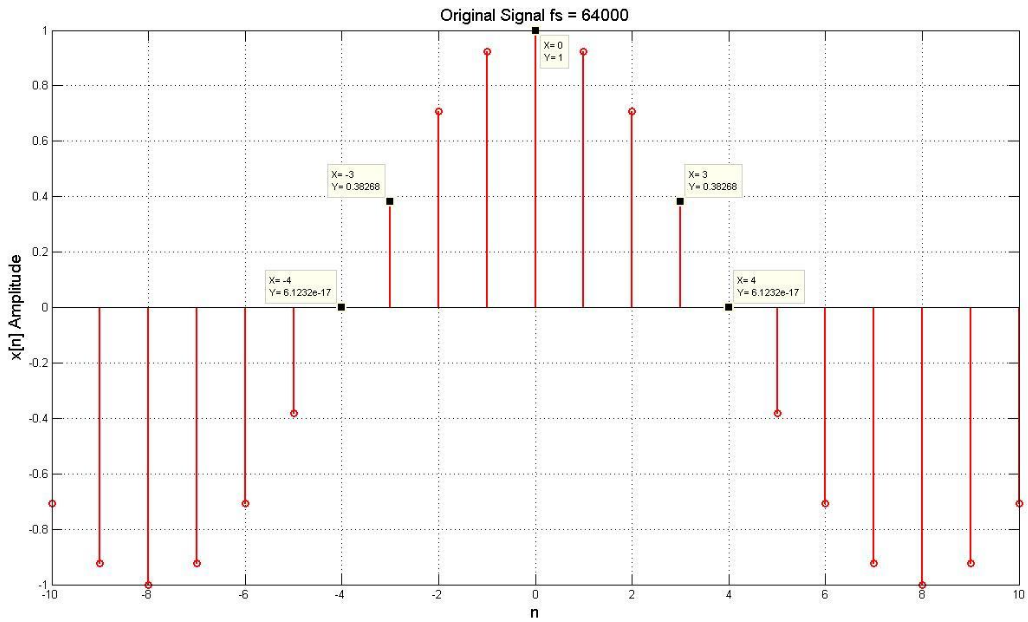



Original Signal fs = 4000
X= -2
Y= 1
X= 0
Y= 1
X= 2
Y= 1
x[n] Amplitude
n

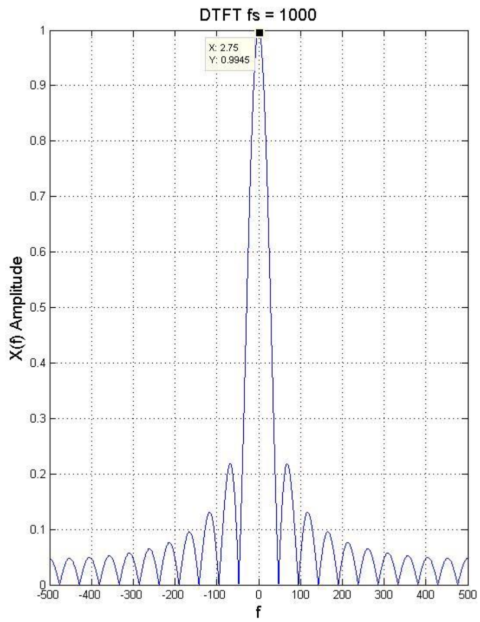

Original Signal at Nyquist Criteria fs = 8000
X= 0
Y= 1
X= 2
Y= 1
x[n] Amplitude
n

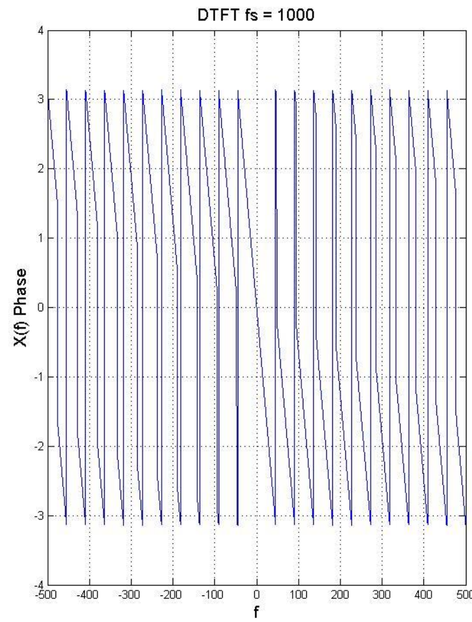

Original Signal fs = 16000
x[n] Amplitude
n
X= -4
Y= 1
X= 0
Y= 1
X= 4
Y= 1

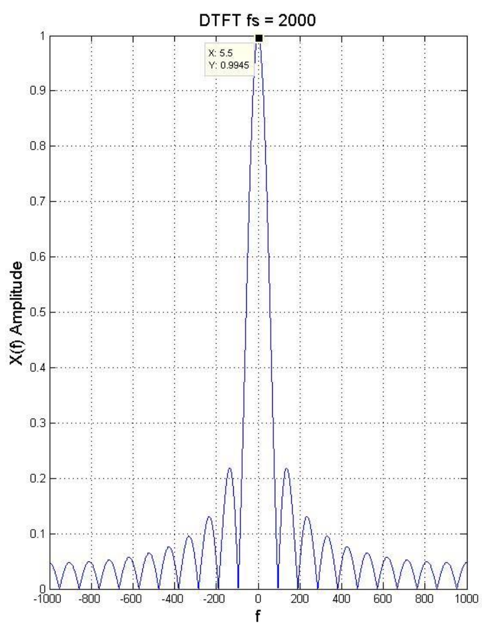

Original Signal fs = 32000
x[n] Amplitude
n
X= -8
Y= 1
X= -7
Y= 0.70711
X= 0
Y= 1
X= -1
Y= 0.70711
X= 1
Y= 0.70711
X= 7
Y= 0.70711
X= 8
Y= 1

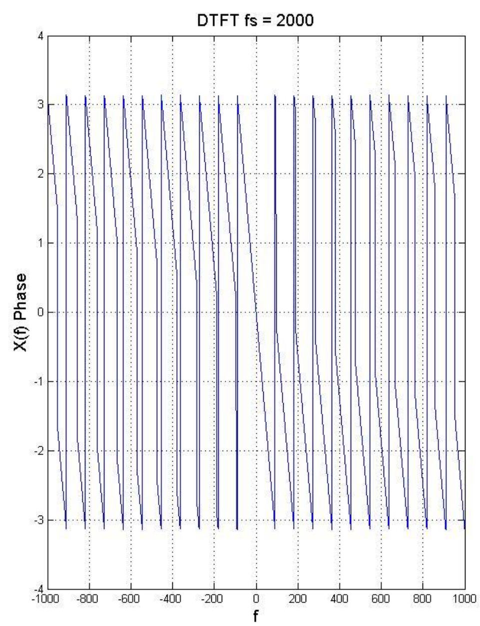


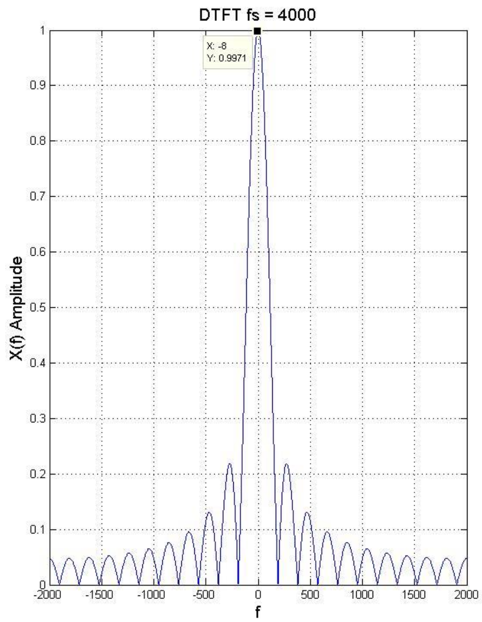
EE215
5/6/2014

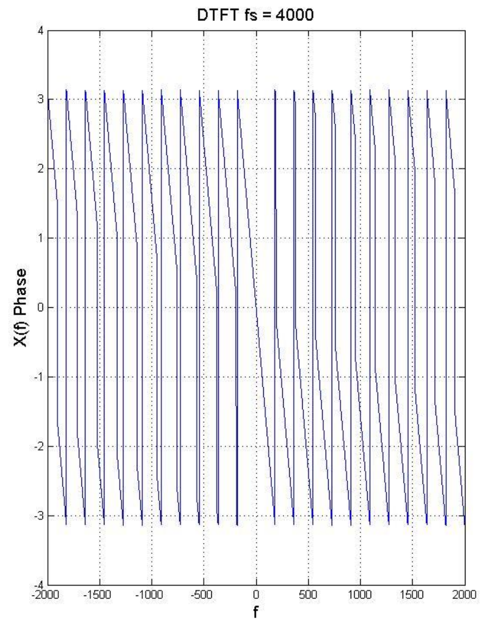

DTFT fs = 1000
X: 2.75
Y: 0.9945
X(f) Amplitude
f

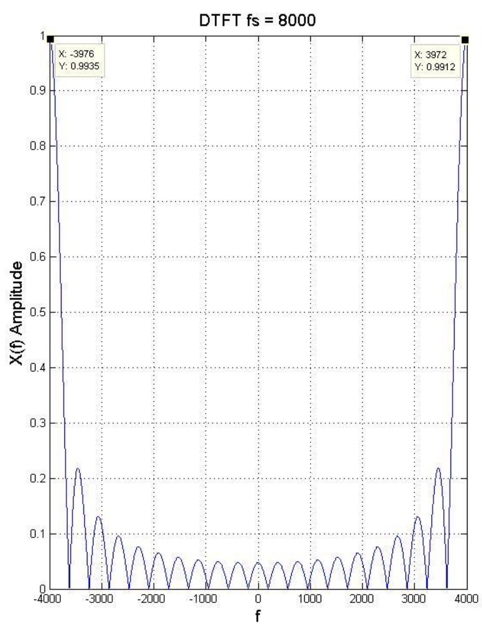

DTFT fs = 1000
X(f) Phase
f

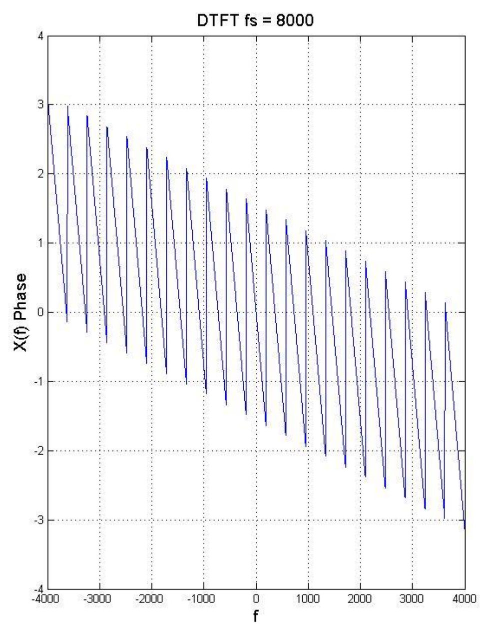

DTFT fs = 2000
X: 5.5
Y: 0.9945
X(f) Amplitude
f

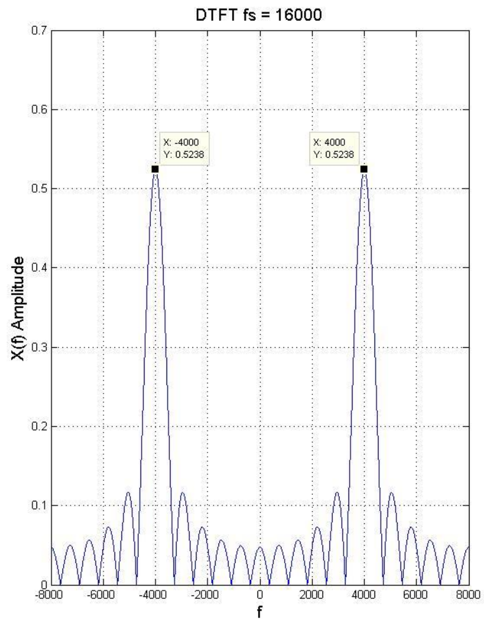

DTFT fs = 2000
X(f) Phase
f

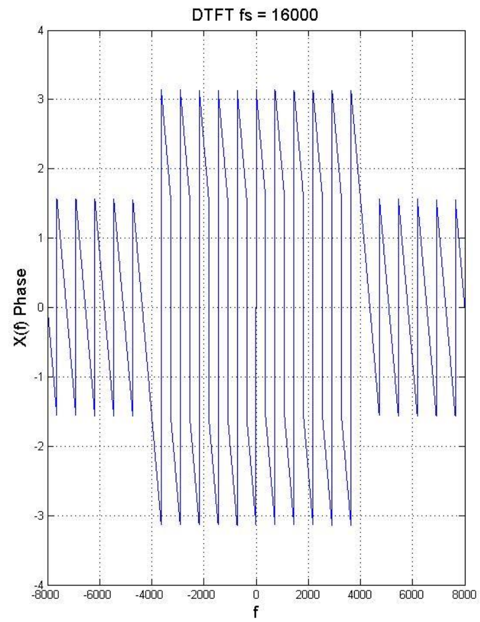

DTFT fs = 4000
X: -8
Y: 0.9971
X(f) Amplitude
f

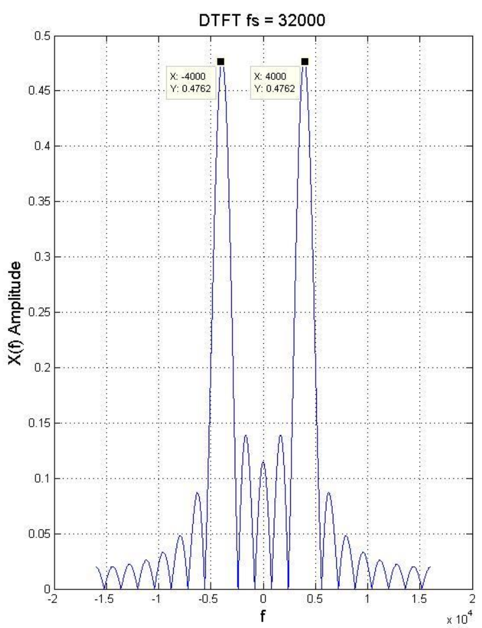

DTFT fs = 4000
X(f) Phase
f

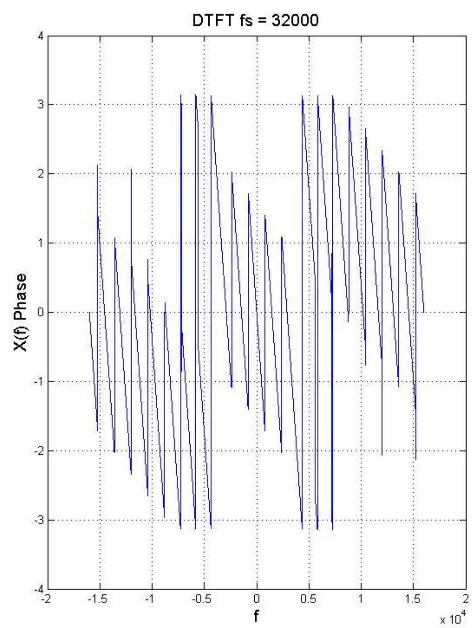

DTFT fs = 8000
X: -3976
Y: 0.9935
X: 3972
Y: 0.9912
X(f) Amplitude
f

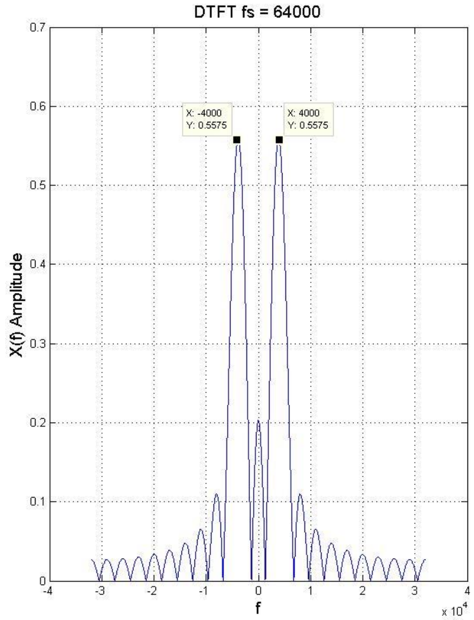

DTFT fs = 8000
X(f) Phase
f

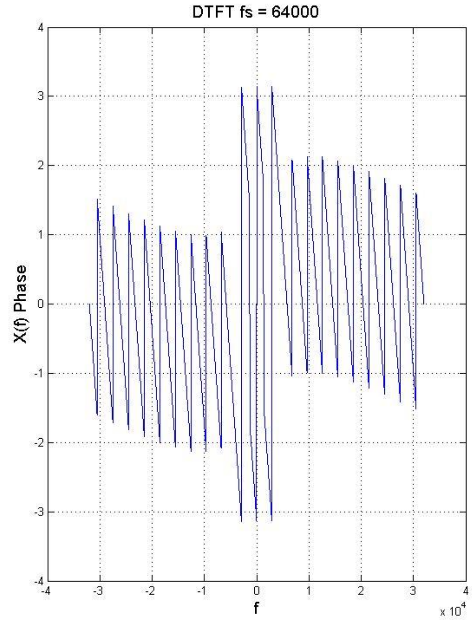

DTFT fs = 16000
X(f) Amplitude
f
X: -4000
Y: 0.5238
X: 4000
Y: 0.5238

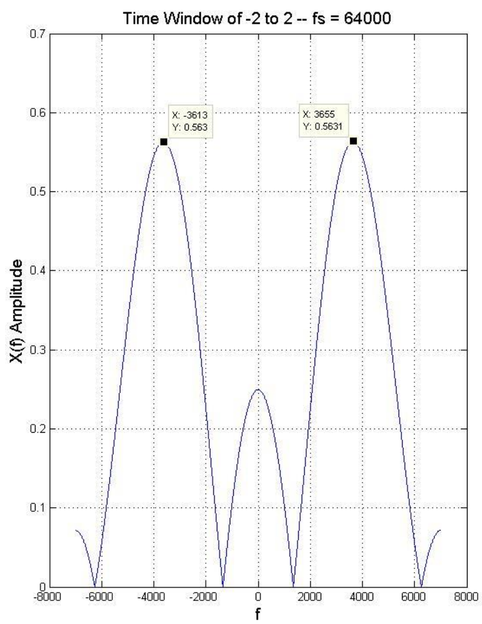

DTFT fs = 16000
X(f) Phase
f

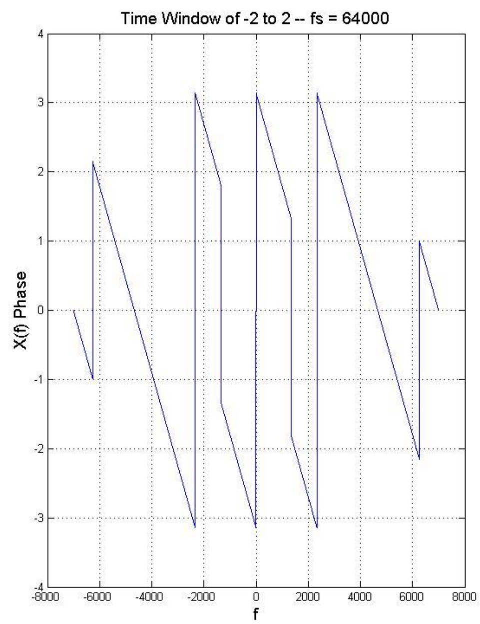

DTFT fs = 32000
X(f) Amplitude
f
X: -4000
Y: 0.4762
X: 4000
Y: 0.4762

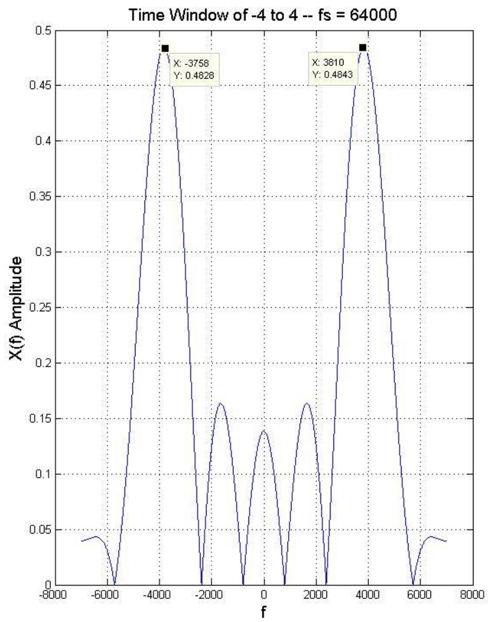

DTFT fs = 32000
X(f) Phase
f

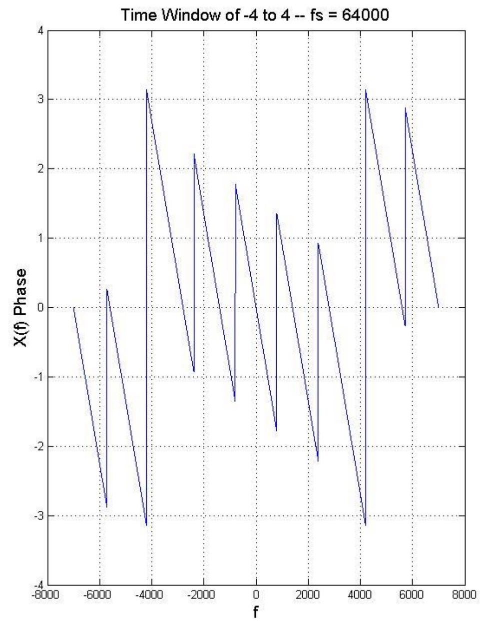

DTFT fs = 64000
X(f) Amplitude
f
x 10^4
X: -4000
Y: 0.5575
X: 4000
Y: 0.5575

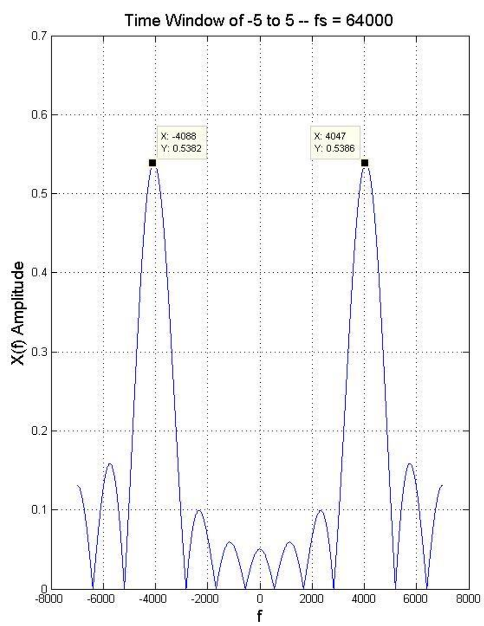

DTFT fs = 64000
X(f) Phase
f
x 10^4

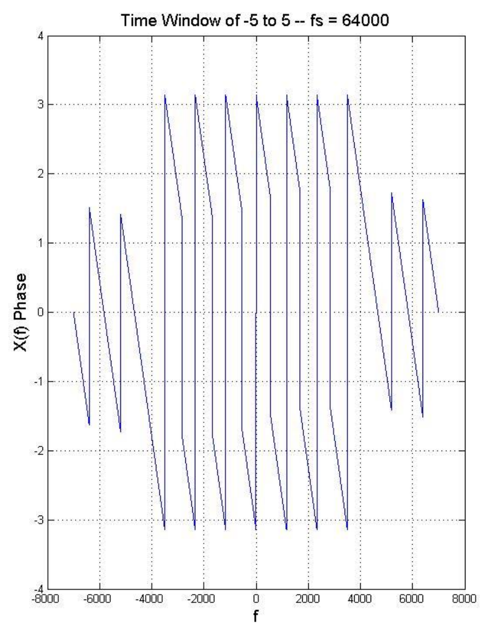

Time Window of -2 to 2 -- fs = 64000
X(f) Amplitude
f
X: -3613
Y: 0.563
X: 3655
Y: 0.5631

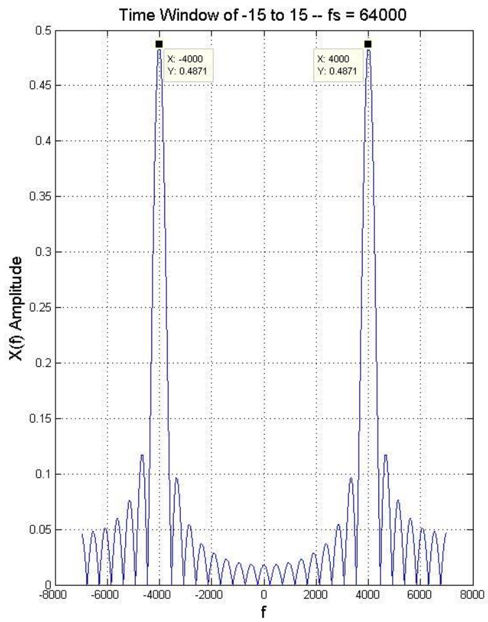

Time Window of -2 to 2 -- fs = 64000
X(f) Phase
f

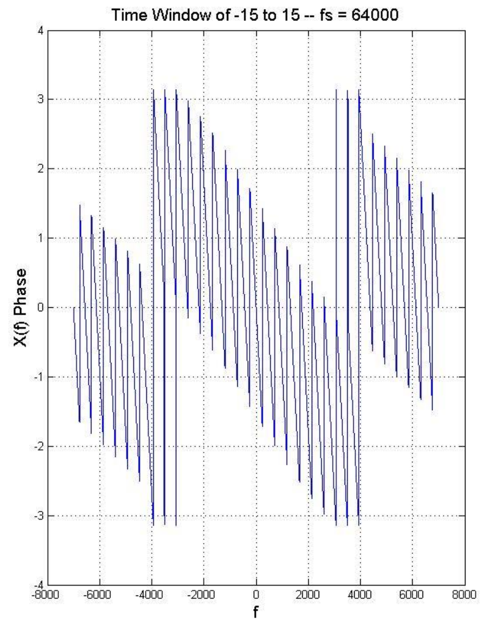

Time Window of -4 to 4 -- fs = 64000
X: -3758
Y: 0.4828
X: 3810
Y: 0.4843
X(f) Amplitude
f

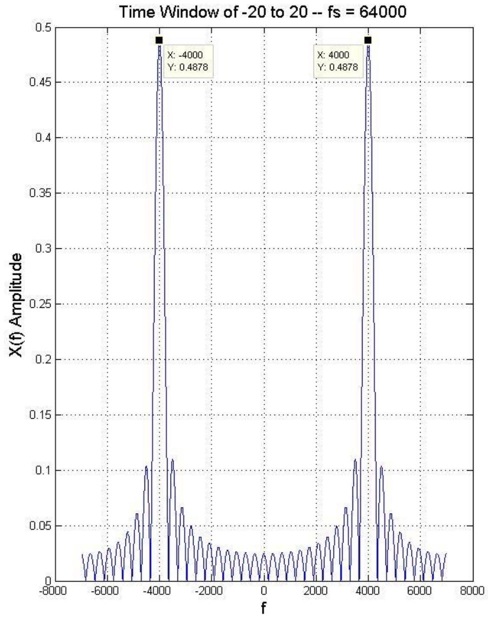

Time Window of -4 to 4 -- fs = 64000
X(f) Phase
f

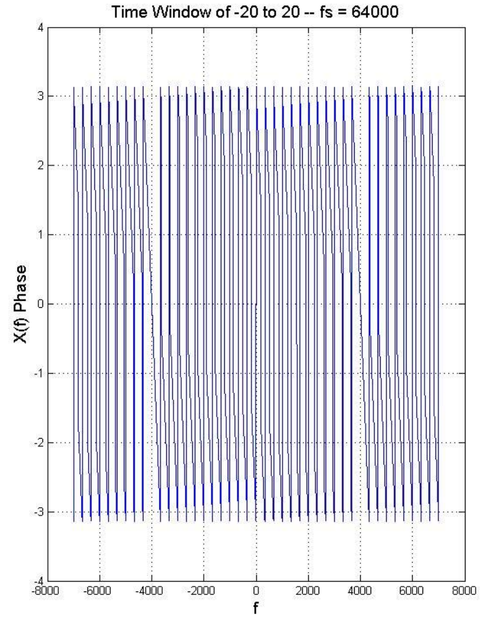

Time Window of -5 to 5 -- fs = 64000
X: -4088
Y: 0.5382
X: 4047
Y: 0.5386
X(f) Amplitude
f

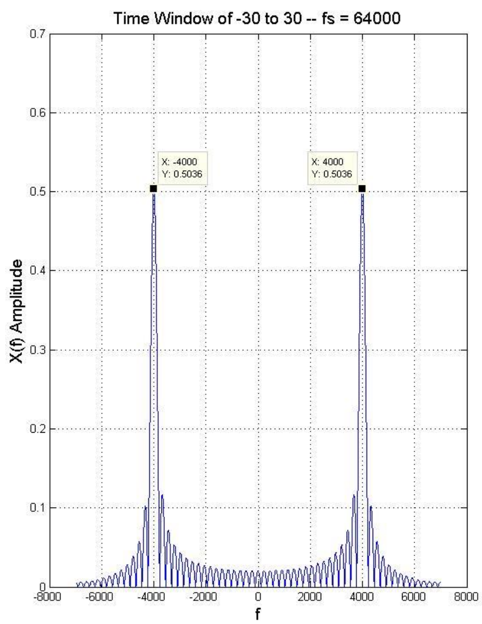

Time Window of -5 to 5 -- fs = 64000
X(f) Phase
f

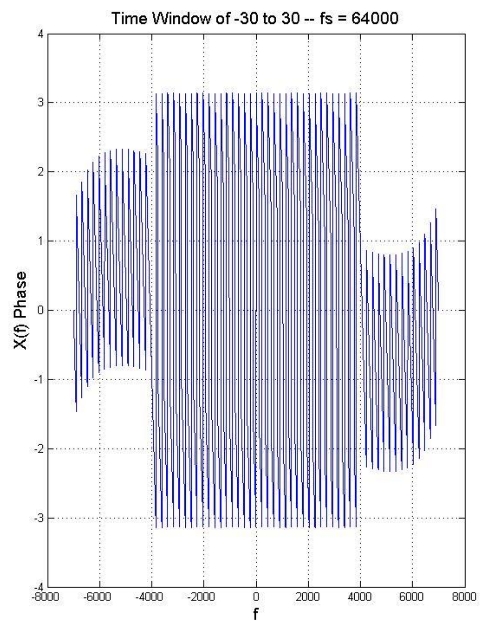

Time Window of -15 to 15 -- fs = 64000
X(f) Amplitude
f
X: -4000
Y: 0.4871
X: 4000
Y: 0.4871

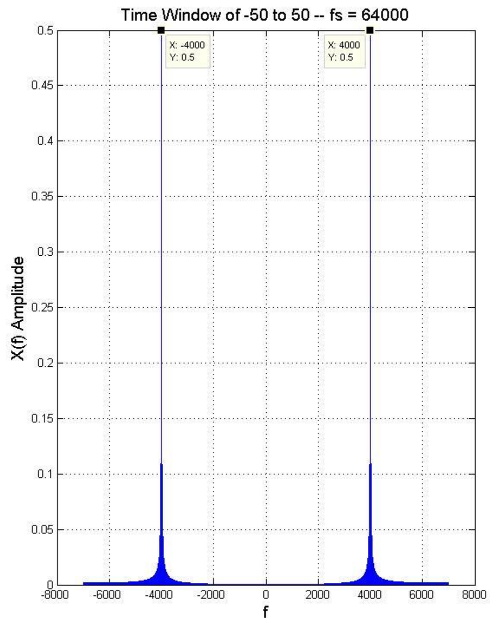

Time Window of -15 to 15 -- fs = 64000
X(f) Phase
f


Time Window of -20 to 20 -- fs = 64000
X: -4000
Y: 0.4878
X: 4000
Y: 0.4878
X(f) Amplitude
f

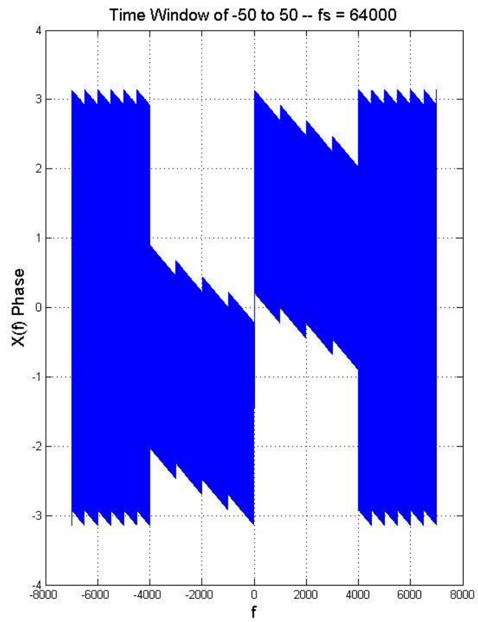

Time Window of -20 to 20 -- fs = 64000
X(f) Phase
f


Time Window of -30 to 30 -- fs = 64000
X: -4000
Y: 0.5036
X: 4000
Y: 0.5036
X(f) Amplitude
f

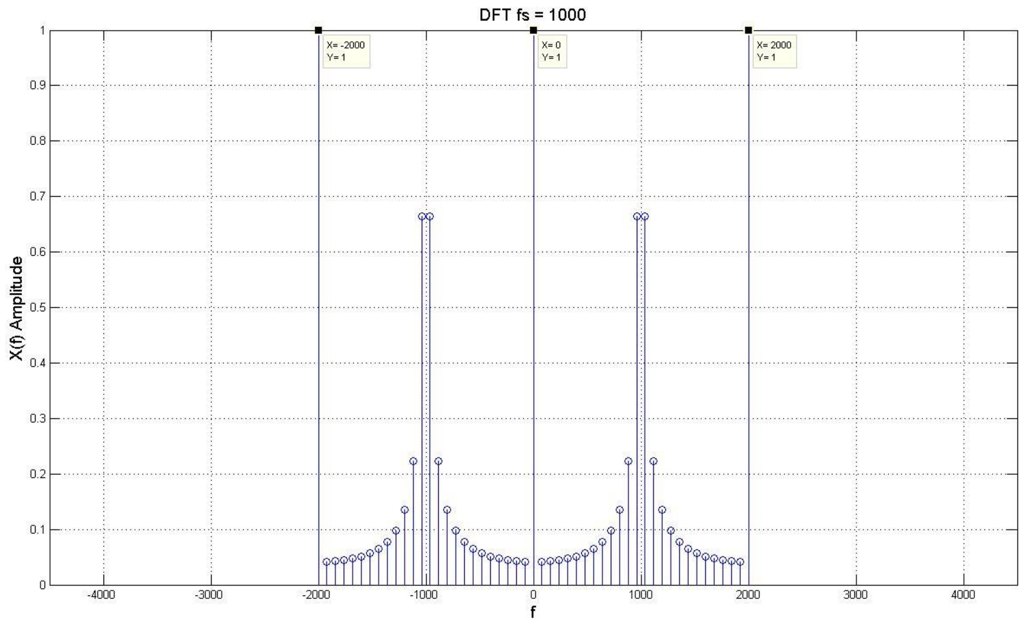

Time Window of -30 to 30 -- fs = 64000
X(f) Phase
f

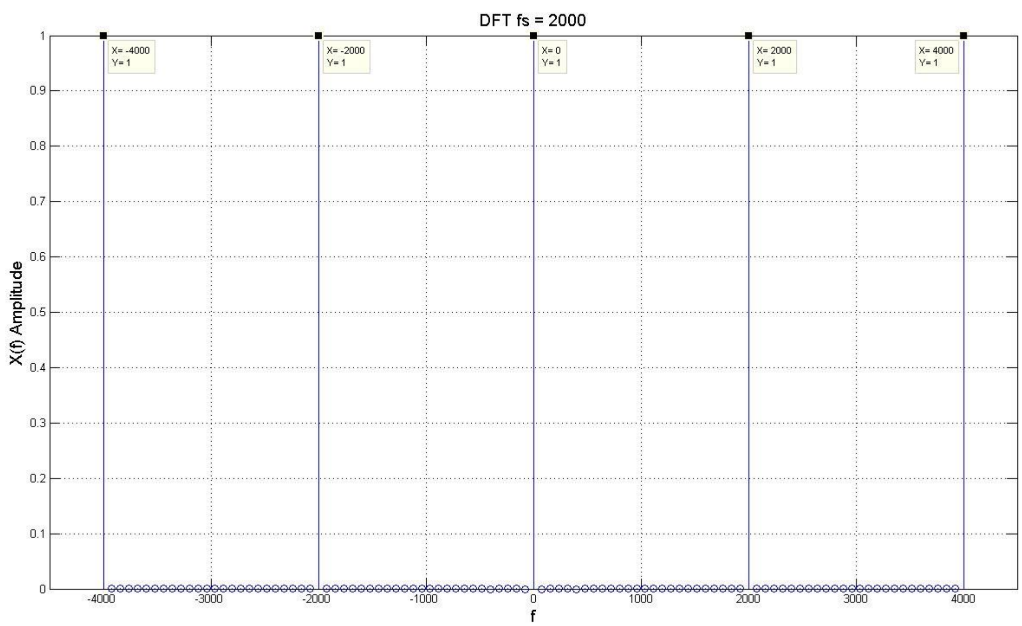


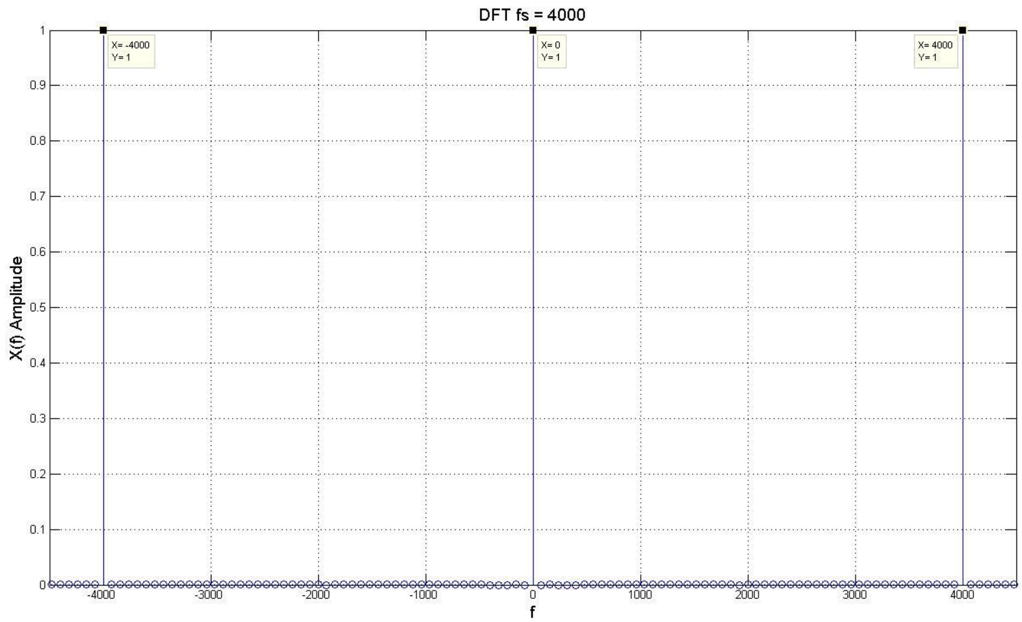
Time Window of -50 to 50 -- fs = 64000
X: -4000
Y: 0.5
X: 4000
Y: 0.5
X(f) Amplitude
f

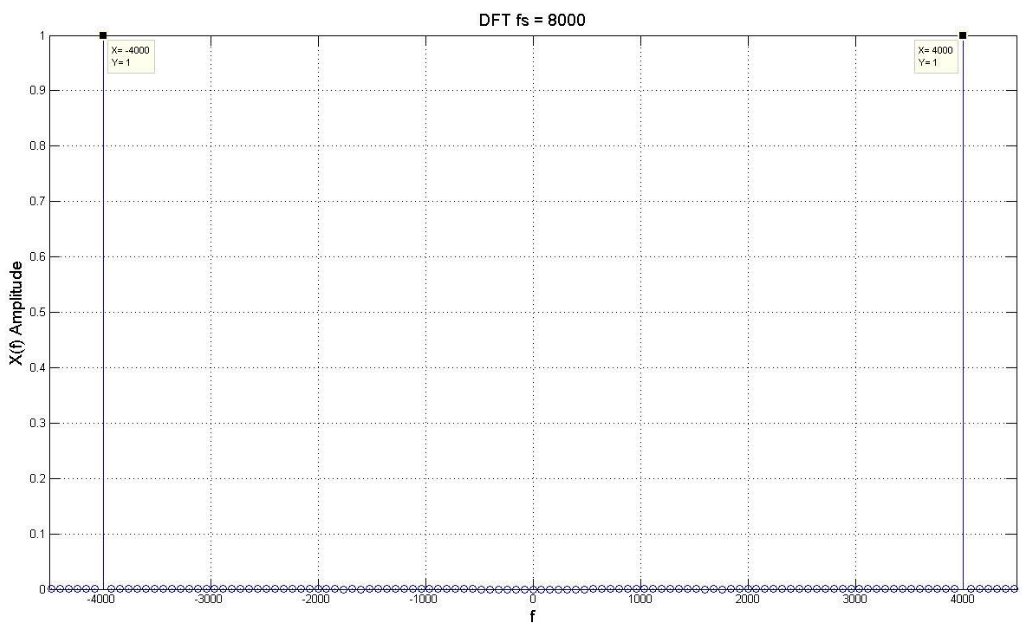


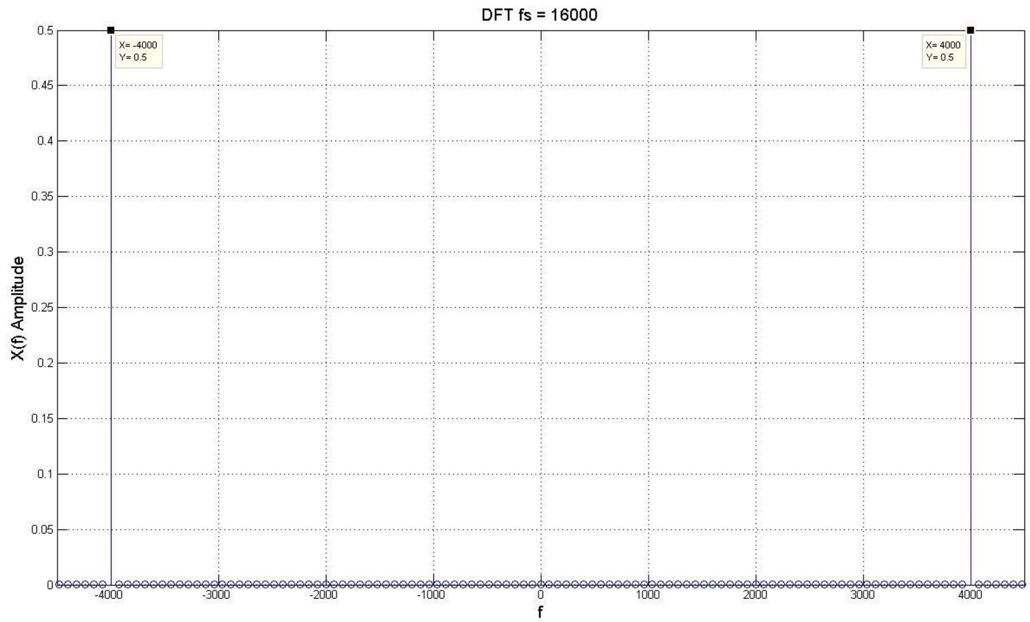
Time Window of -50 to 50 -- fs = 64000
X(f) Phase
f

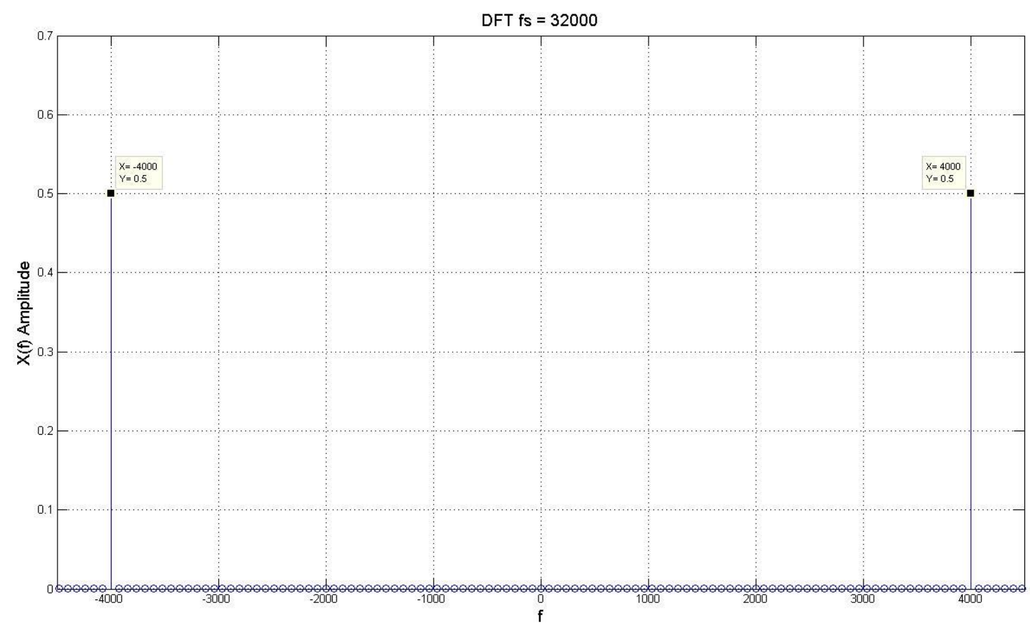

DFT fs = 1000
X= -2000
Y= 1
X= 0
Y= 1
X= 2000
Y= 1
X(f) Amplitude
f

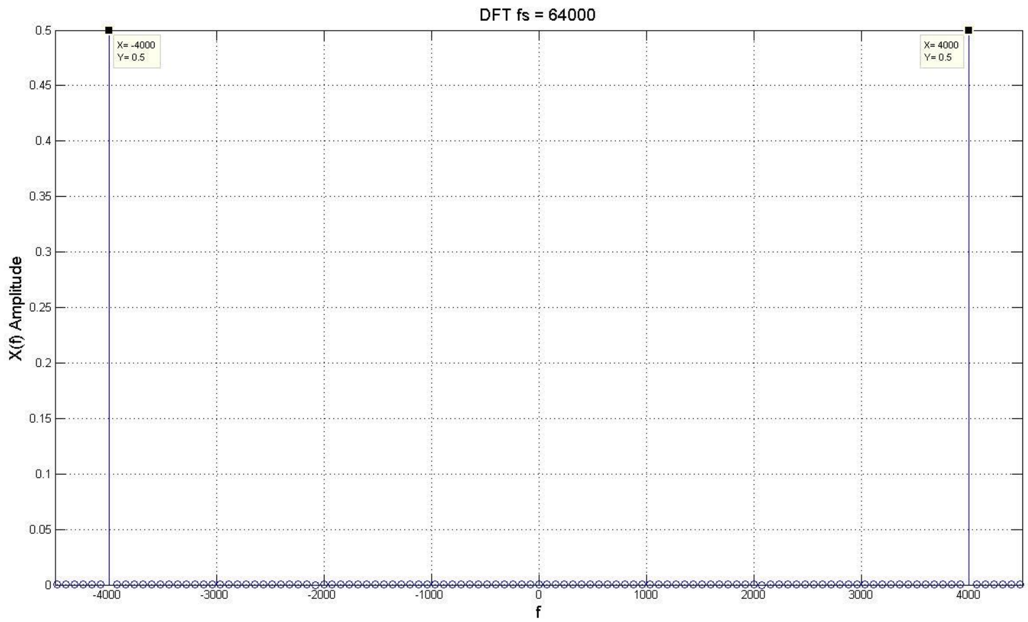

DFT fs = 2000
X= -4000
Y= 1
X= -2000
Y= 1
X= 0
Y= 1
X= 2000
Y= 1
X= 4000
Y= 1
X(f) Amplitude
f

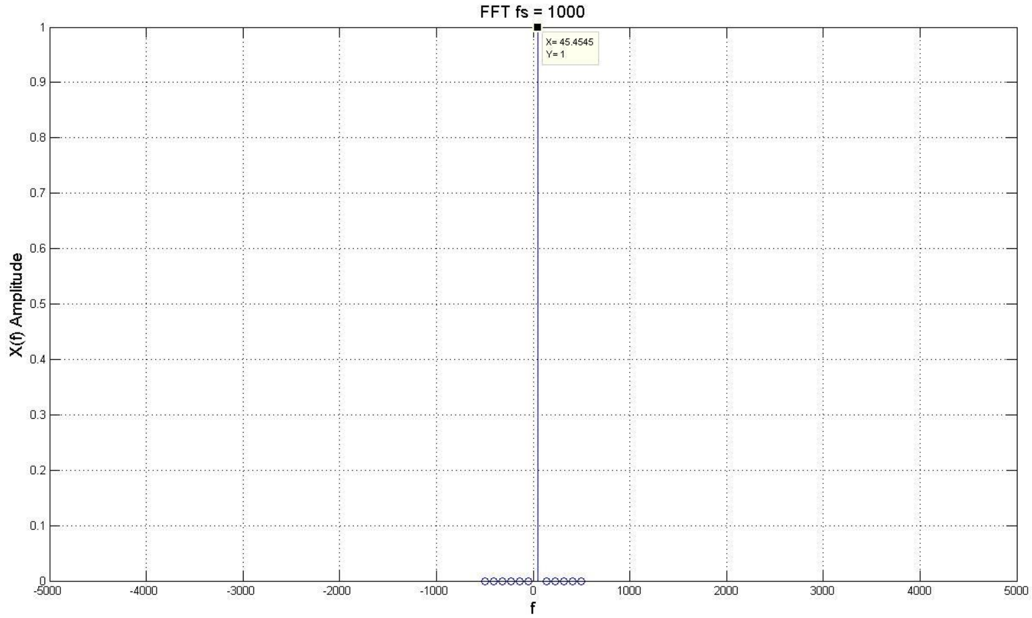

DFT fs = 4000
X= -4000
Y= 1
X= 0
Y= 1
X= 4000
Y= 1
X(f) Amplitude
f
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000

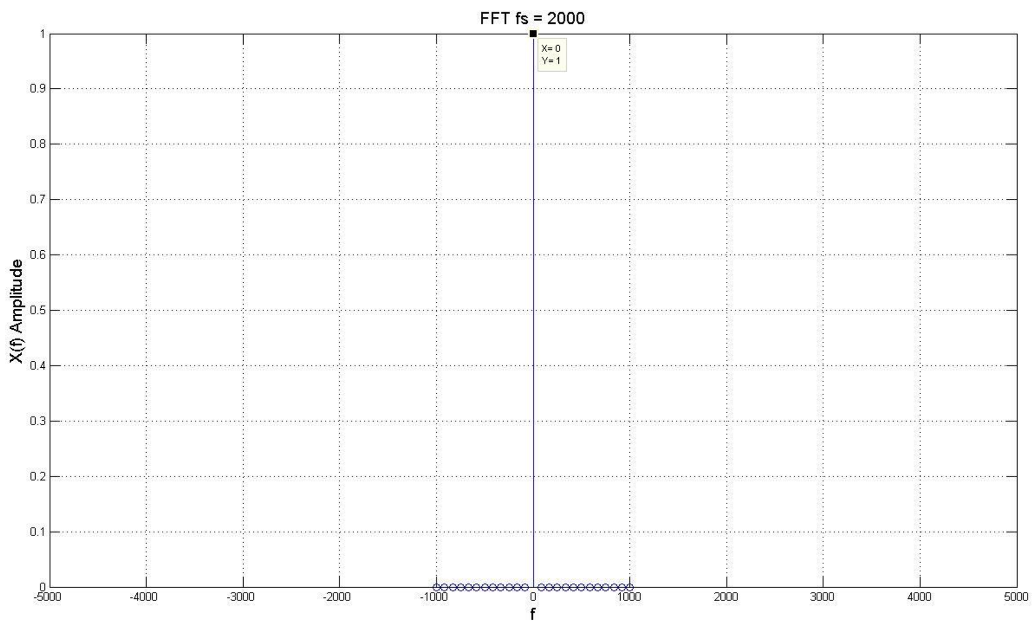

DFT fs = 8000
X= -4000
Y= 1
X= 4000
Y= 1
X(f) Amplitude
f
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000

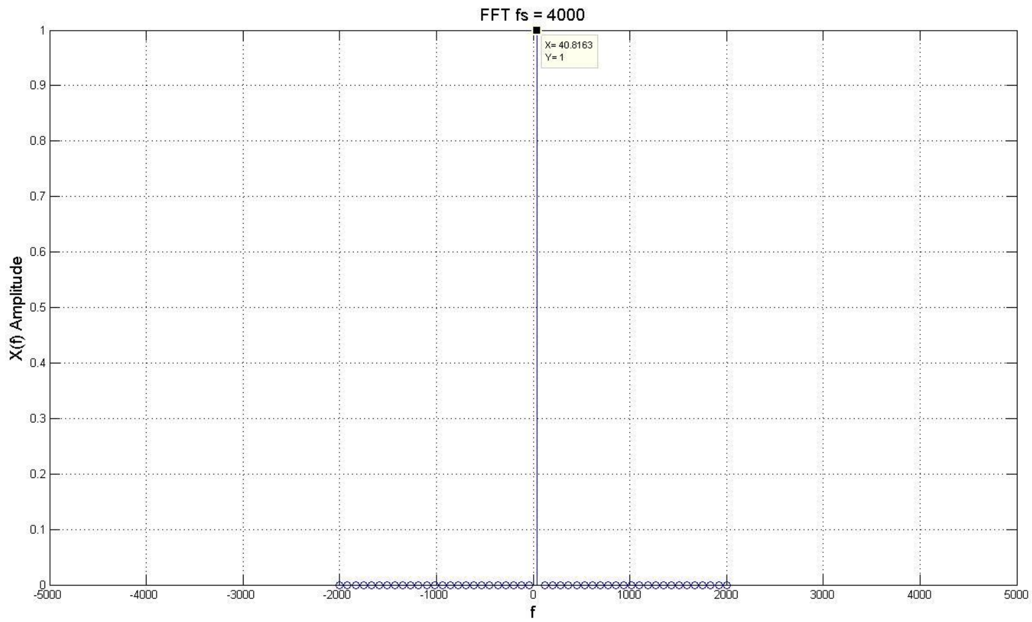

DFT fs = 16000
X(f) Amplitude
f
X= -4000
Y= 0.5
X= 4000
Y= 0.5

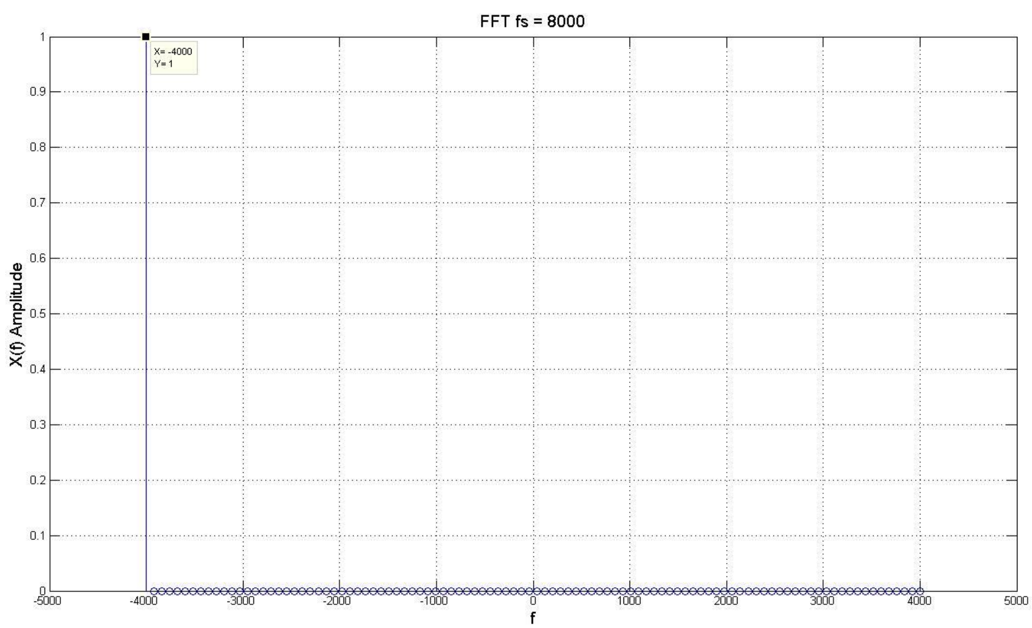

DFT fs = 32000
X(f) Amplitude
f
X= -4000
Y= 0.5
X= 4000
Y= 0.5

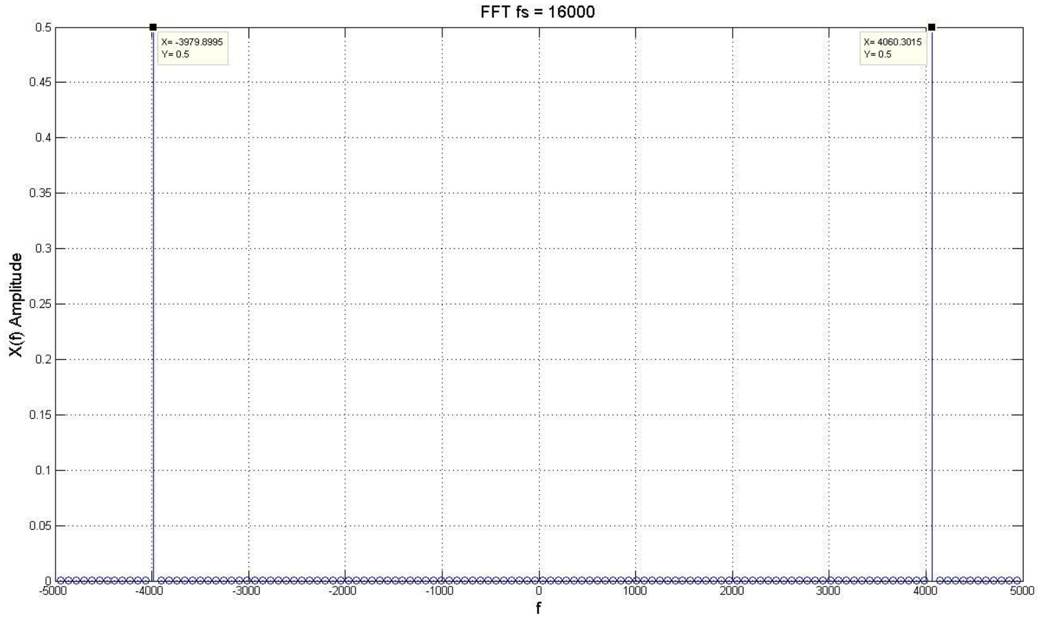

DFT fs = 64000
X= -4000
Y= 0.5
X= 4000
Y= 0.5
X(f) Amplitude
f
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000

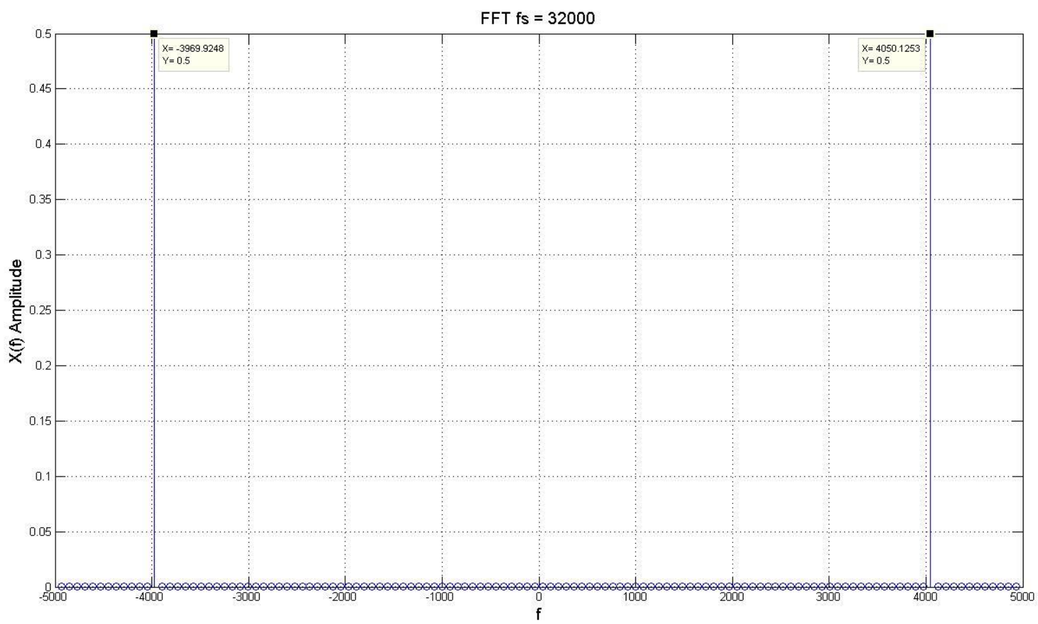

FFT fs = 1000
X= 45.4545
Y= 1
X(f) Amplitude
f

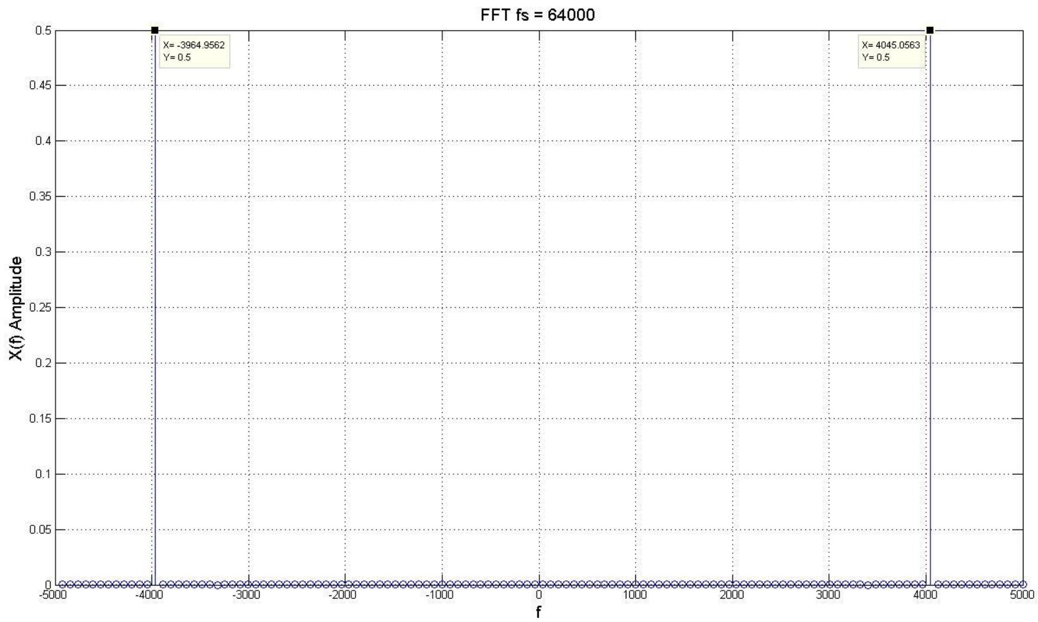

FFT fs = 2000
X= 0
Y= 1
X(f) Amplitude
f

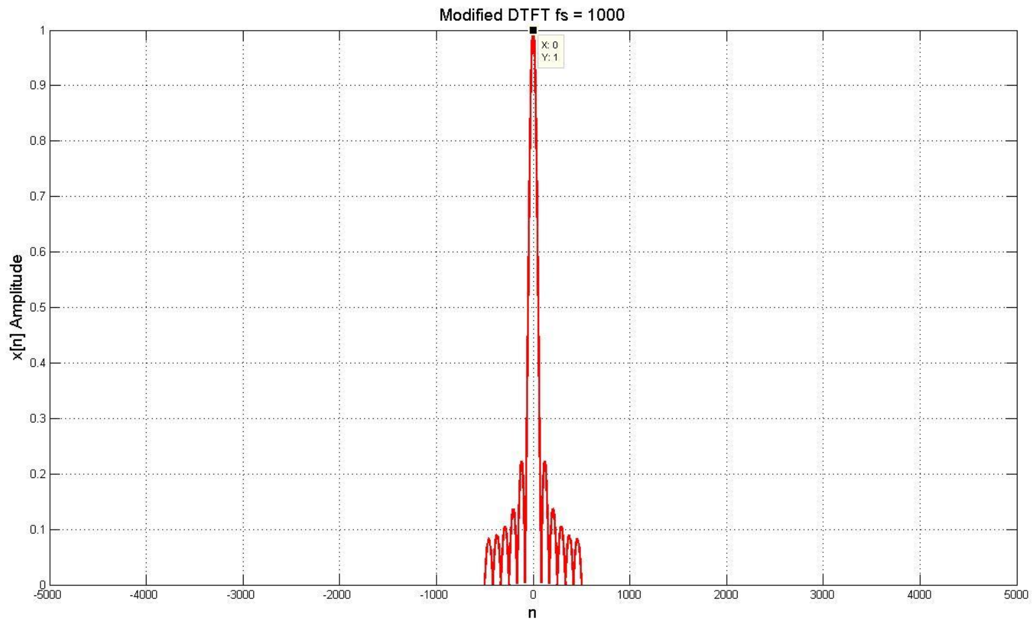

FFT fs = 4000
X= 40.8163
Y= 1
X(f) Amplitude
f

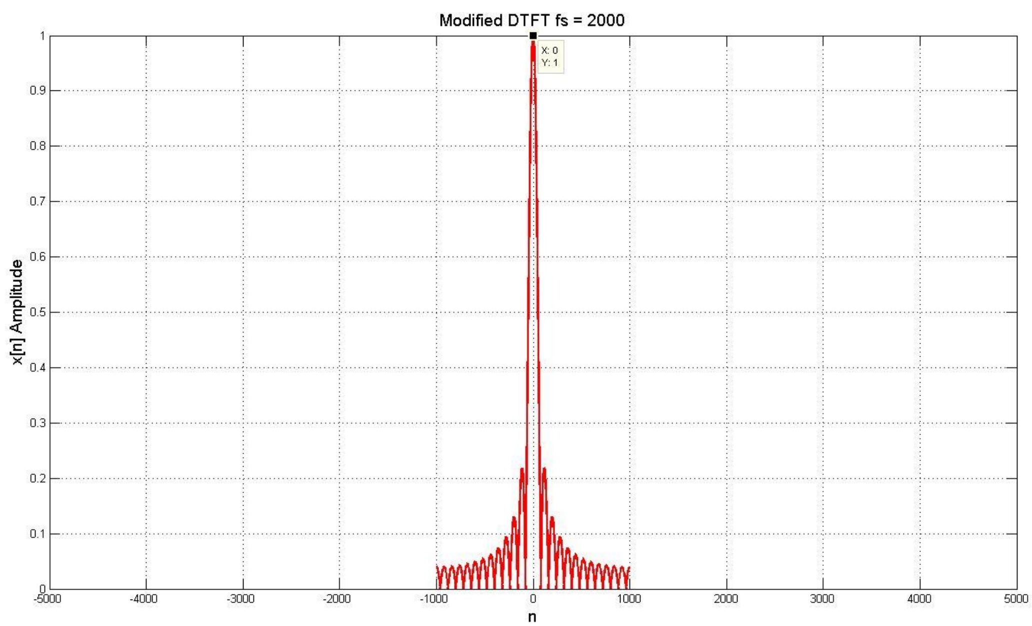

FFT fs = 8000
X= -4000
Y= 1
X(f) Amplitude
f

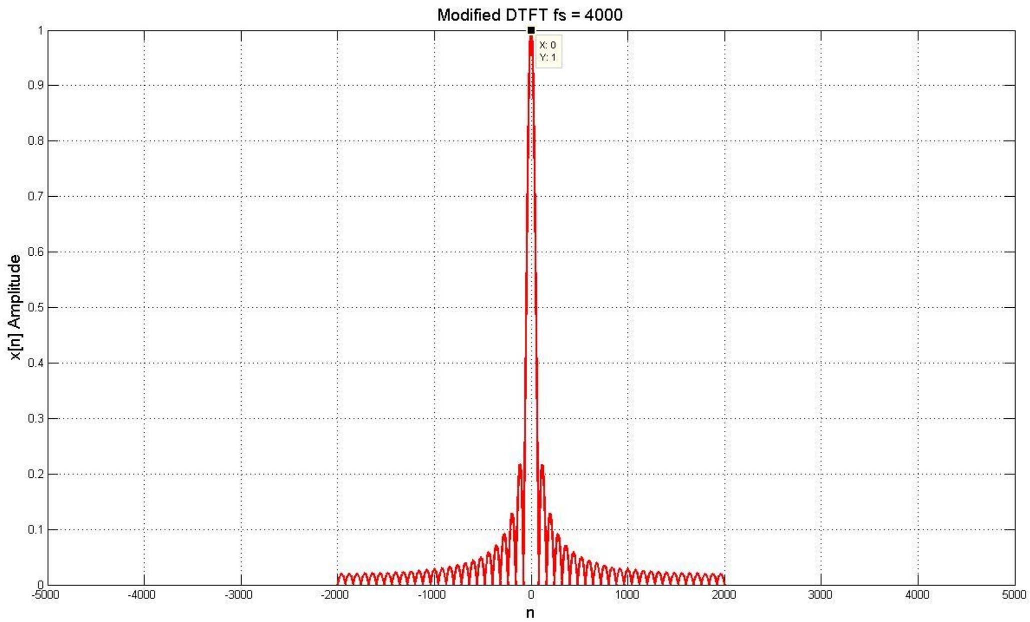

FFT fs = 16000
X= -3979.8995
Y= 0.5
X= 4060.3015
Y= 0.5
X(f) Amplitude
f

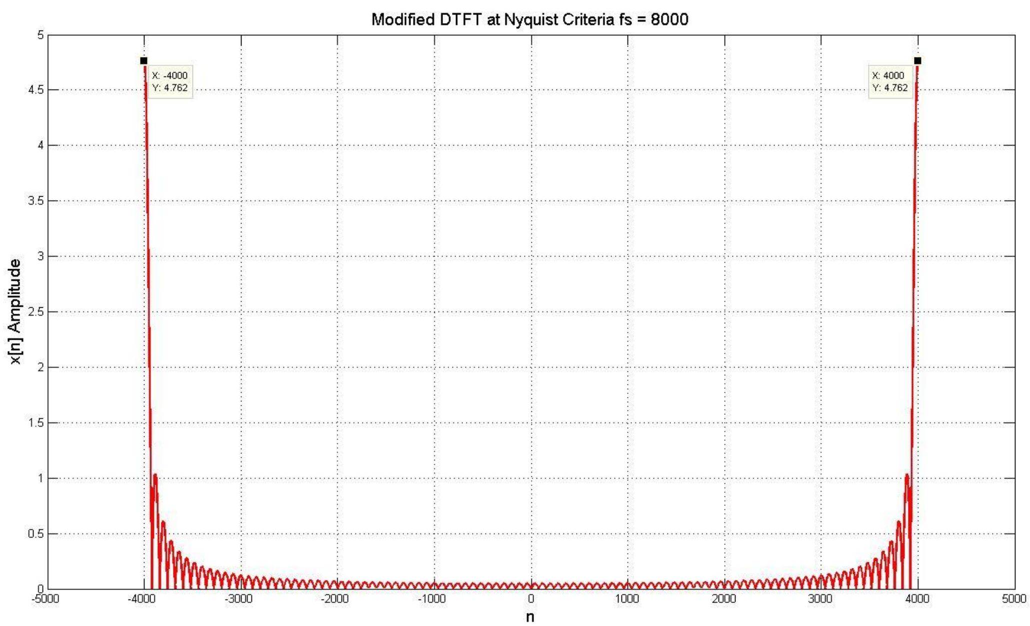

FFT fs = 32000
X= -3969.9248
Y= 0.5
X= 4050.1253
Y= 0.5
X(f) Amplitude
f

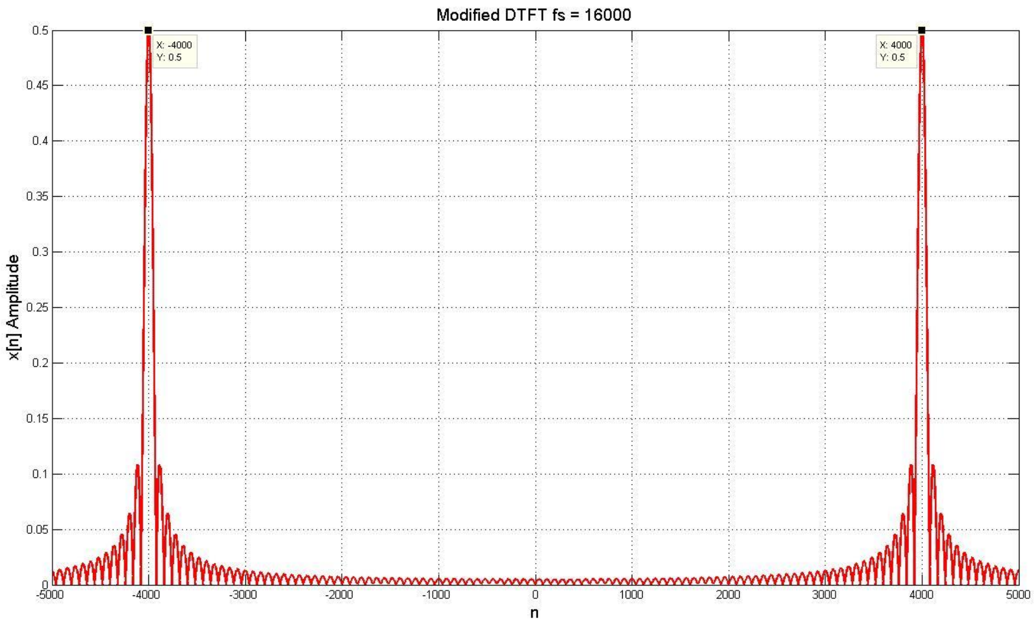

FFT fs = 64000
X= -3964.9562
Y= 0.5
X= 4045.0563
Y= 0.5
X(f) Amplitude
f
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-5000
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000
5000

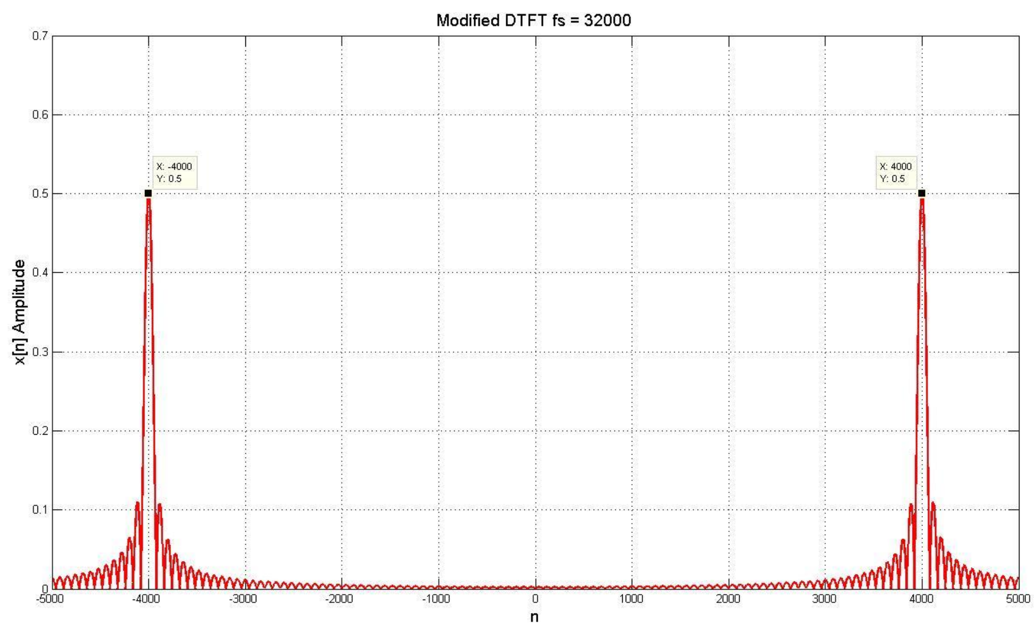

Modified DTFT fs = 1000
X: 0
Y: 1
x[n] Amplitude
n
-5000
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000
5000

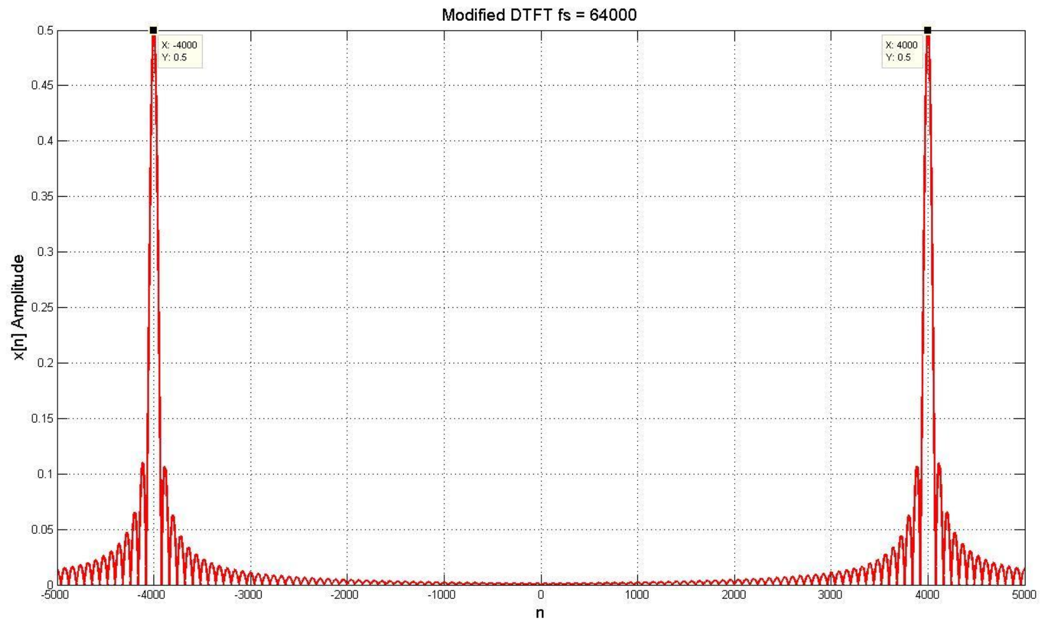

Modified DTFT fs = 2000
X: 0
Y: 1
x[n] Amplitude
n
-5000
-4000
-3000
-2000
-1000
0
1000
2000
3000
4000
5000


Modified DTFT fs = 4000
x[n] Amplitude
n
X: 0
Y: 1


Modified DTFT at Nyquist Criteria fs = 8000
x[n] Amplitude
n
X: -4000
Y: 4.762
X: 4000
Y: 4.762


Modified DTFT fs = 16000
X: -4000
Y: 0.5
X: 4000
Y: 0.5
x[n] Amplitude
n


Modified DTFT fs = 32000
X: -4000
Y: 0.5
X: 4000
Y: 0.5
x[n] Amplitude
n



EE215
5/6/2014