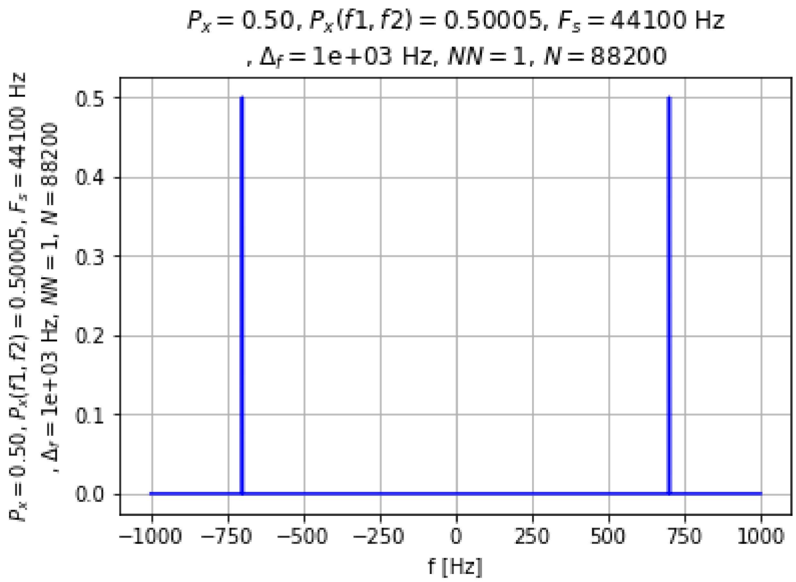
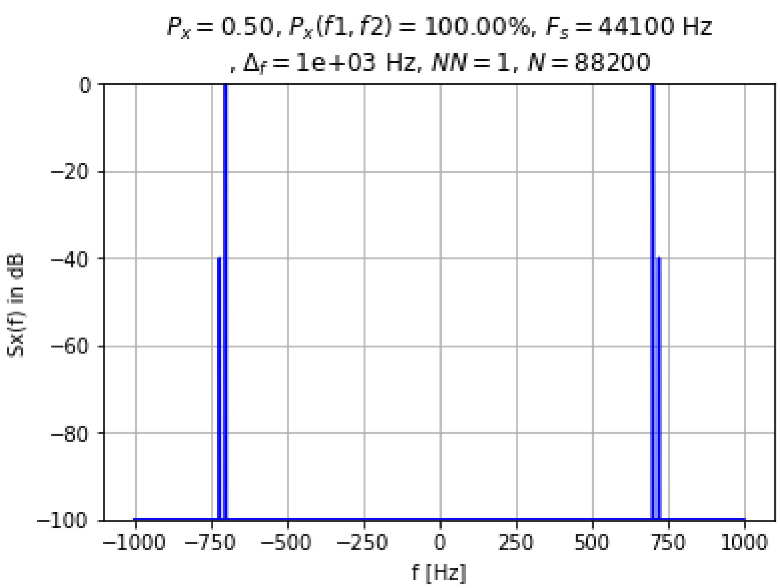
Expt 1 (a)

In [2]:
```
run Q1A
```



Both the sinusoids are not visible in this case. We enhance the code to display to display PSD in db so spectral lines of both waveforms are visible

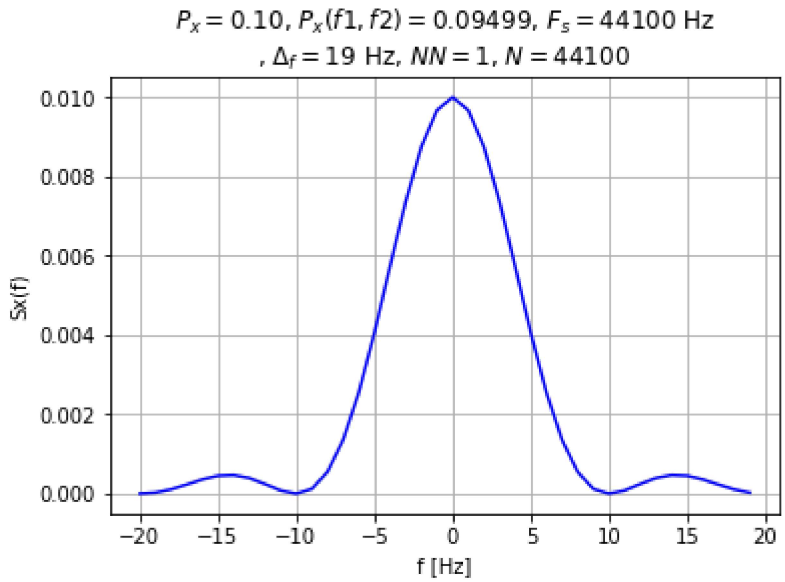
In [2]:
```
run Q1A
```



In the above figure both sinusoids are now visible

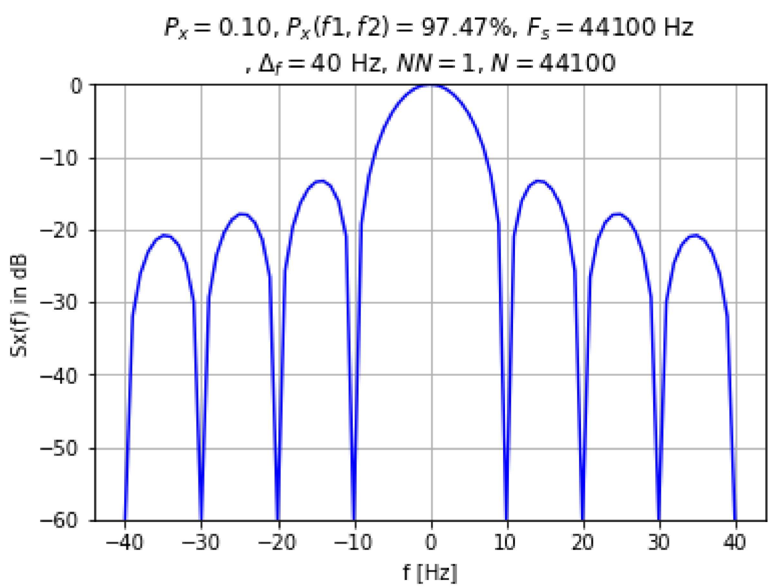
Expt 1 (b)

In [2]:
```
run Q1B
```



The above diagram shows total power as well as power over a range of frequencies for a rectangular pulse of width 100 ms and amplitude 1. This graph uses a linear and absolute vertical axis and displays Px(f1,f2) as an absolute value.

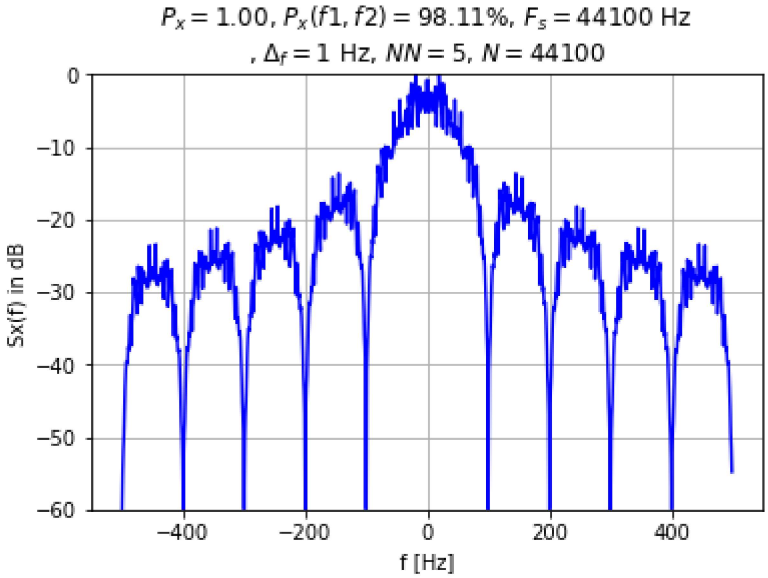
In [3]:
```
run Q1B
```



The second graph shows the Sx(f) of the same rectangular pulse normalized (i.e., maximum at 0 dB) and in dB and displays Px(f1,f2) as a percentage of the total power Px

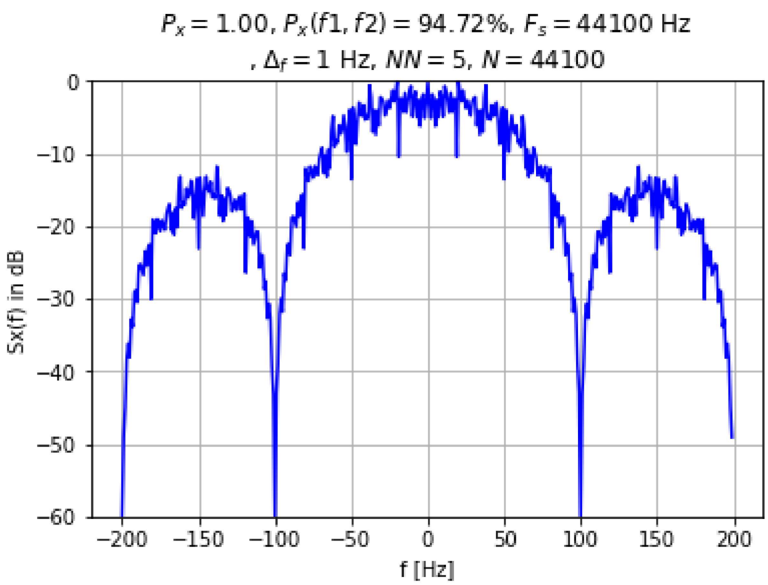
Expt 1 (c)

In [8]:
```
run Q1C
```



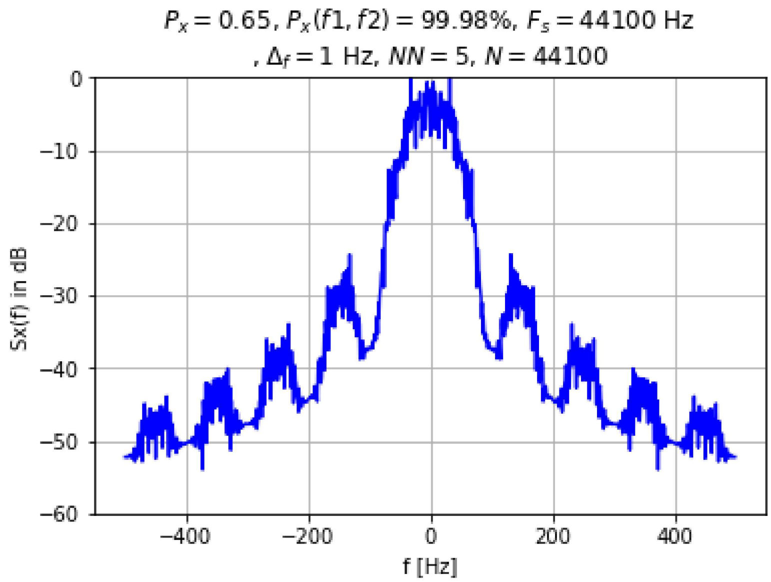
PSD of the rectangular PAM signal over a frequency range of -500, 500. It is equal to 98.11% of the total power.

In [9]:
```
run Q1C
```



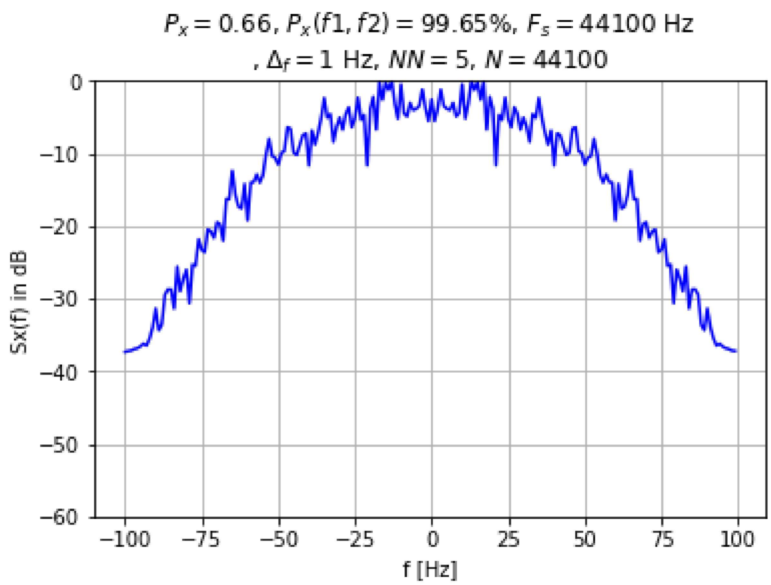
PSD of the rectangular PAM signal over a frequency range of -2Fb, 2Fb. It is equal to 94.72% of the total power.

In [10]:
```
run Q1C
```



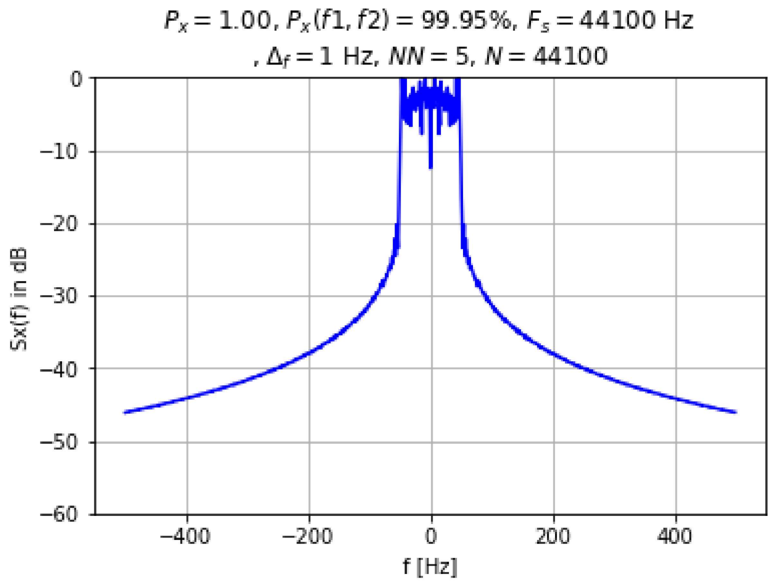
PSD of the triangular PAM signal over a frequency range of -500, 500. It is equal to 99.98% of the total power.

In [11]:
```
run Q1C
```



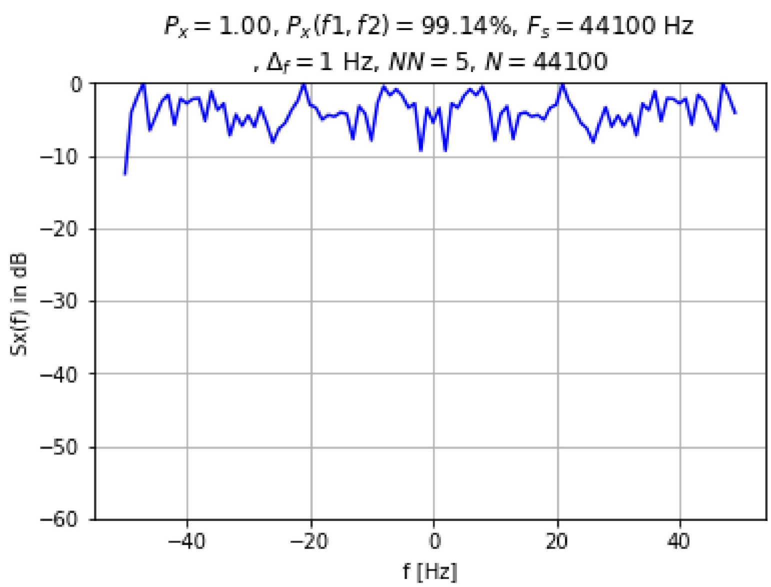
PSD of the triangular PAM signal over a frequency range of -Fb, Fb. It is equal to 99.65% of the total power.

In [12]:
```
run Q1C
```



PSD of the sinc PAM signal over a frequency range of -500, 500. It is equal to 99.9% of the total power.

In [13]:
```
run Q1C
```



PSD of the sinc PAM signal over a frequency range of -50, 50. It is equal to 99.14% of the total power.

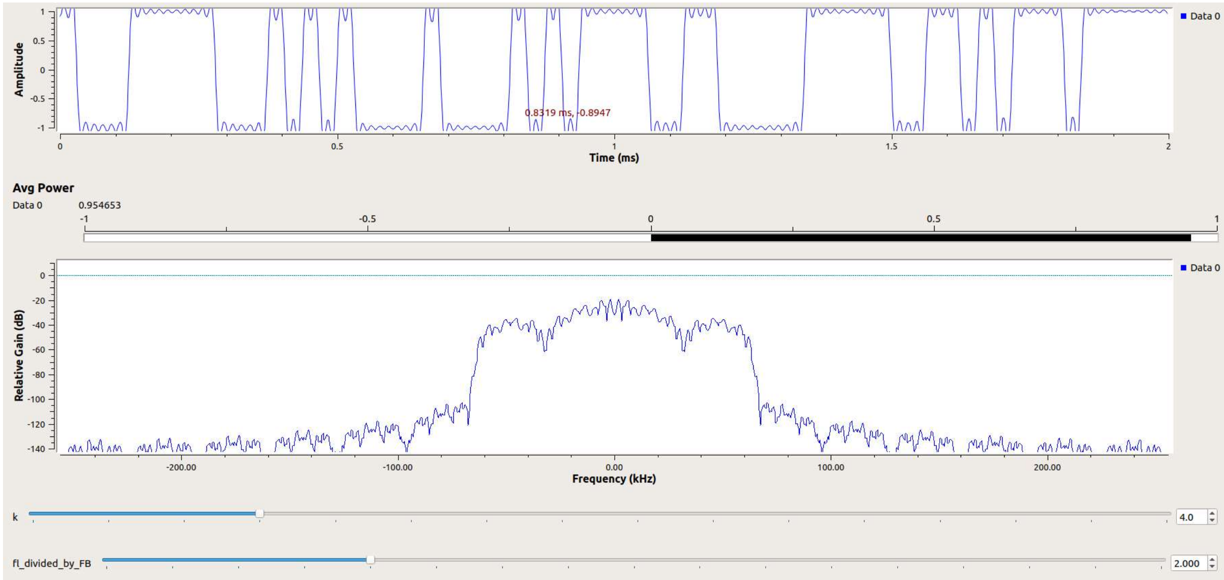
Thus for the rectangular, sinc and triangular signals we find that most of the power is contained within the corresponding bandwidth of 2Fb, Fb/2, Fb

Expt 1(d)

PAM signal based on a rectangular pulse p(t) after low pass filtering at Fl = 2Fb. We observe that 95% of the total average power is contained in the bandwidth of Fb

In [5]:
```python
from IPython.display import Image
Image(filename='capture.png')
```

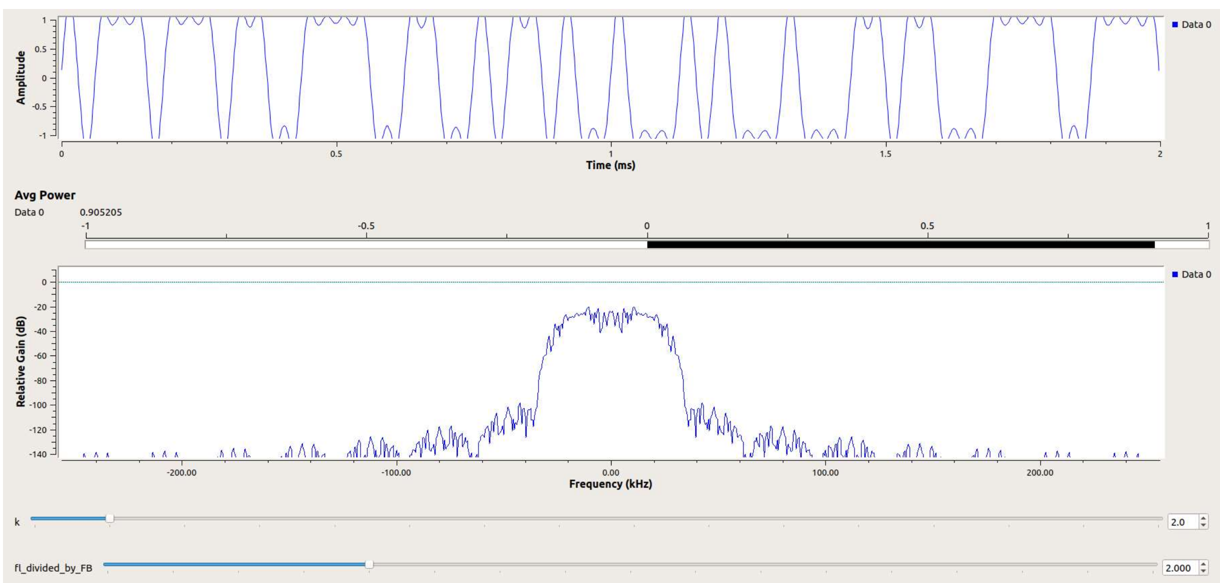
Out[5]:



PAM signal based on a triangular pulse p(t) after low pass filtering at Fl = Fb. The total power contained within the bandwidth of Fb is ~ 90%

In [6]:
```python
from IPython.display import Image
Image(filename='capture.png')
```

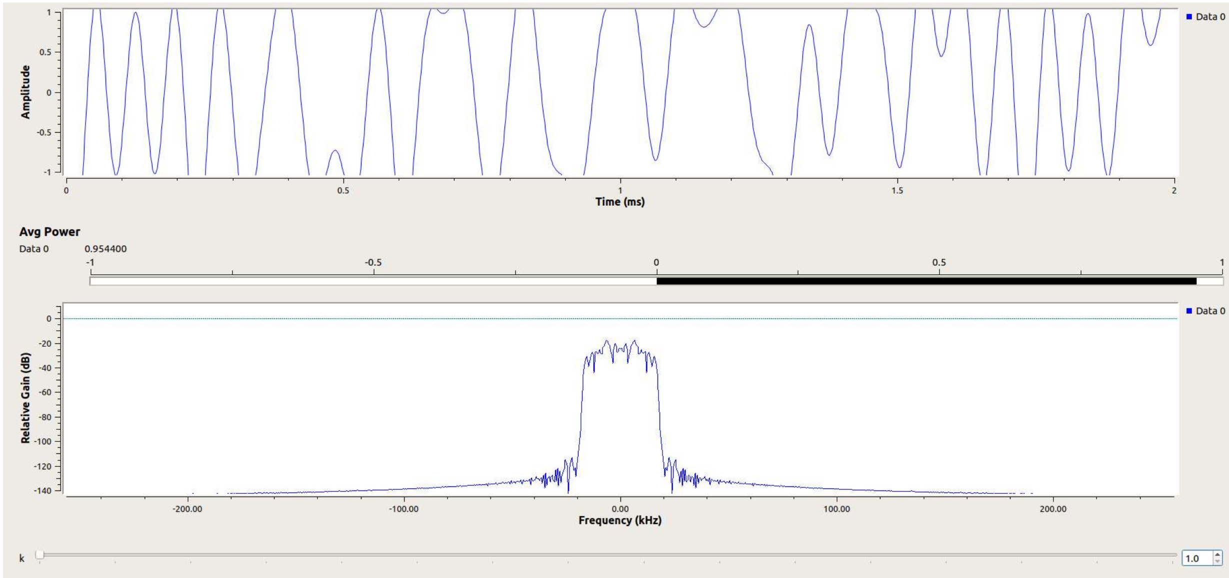
Out[6]:



PAM signal based on a sinc pulse after low pass filtering at Fl = Fb/2. ~ 95% of the total power is contained in this bandwidth

In [3]:
```python
from IPython.display import Image
Image(filename='capture.png')
```

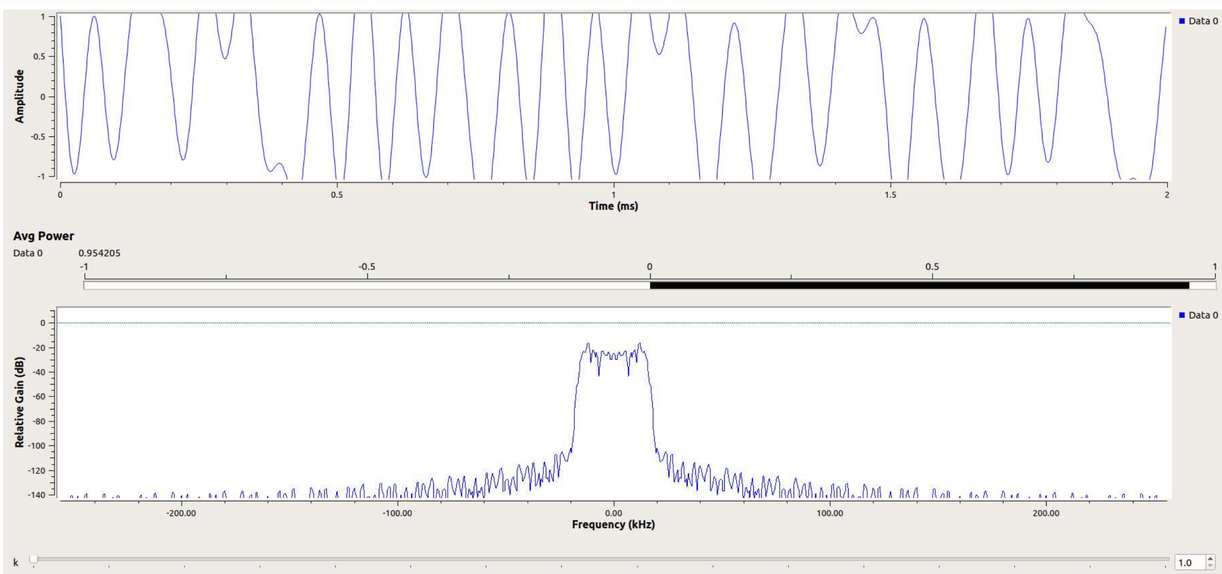
Out[3]:



PAM signal based on a RCf pulse after low pass filtering at Fl = Fb/2. The total power contained in this bandwidth is also 95%

In [4]:
```python
from IPython.display import Image
Image(filename='capture.png')
```

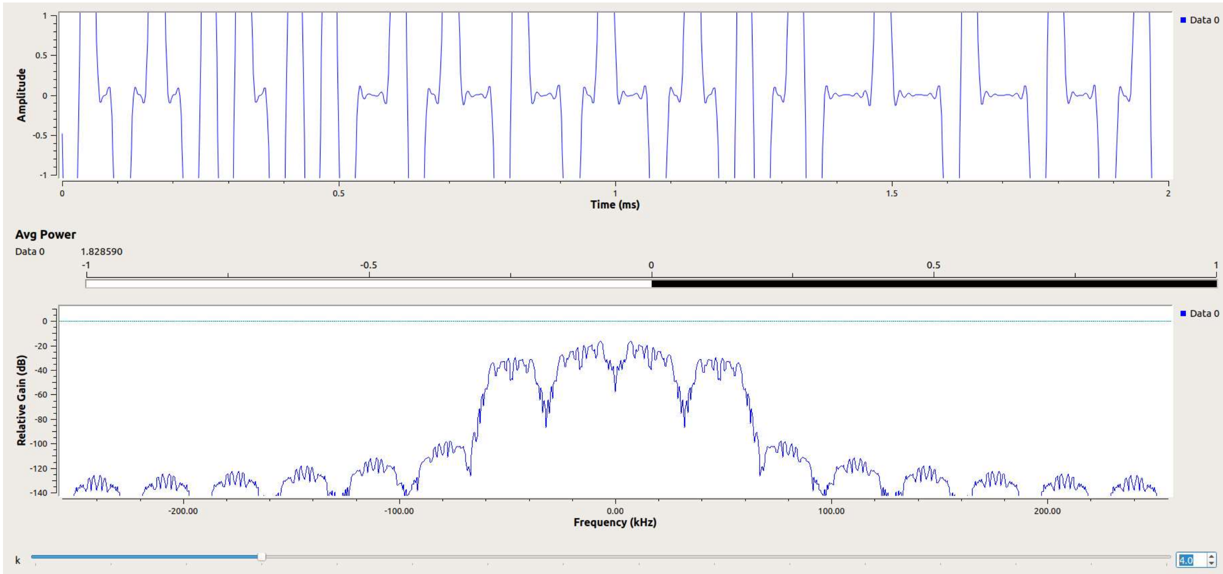
Out[4]:



PAM signal based on a manchester pulse after low pass filtering at Fl = 2Fb similar to rectangular signal, however We observe that the total average power is more than 100% which shouldnt be the case. But it can be justified by making a note of the fact that the amplitude is more than one in this case. This is causing the power to be more than 100%

In [6]:
```python
from IPython.display import Image
Image(filename='capture.png')
```

Out[6]:

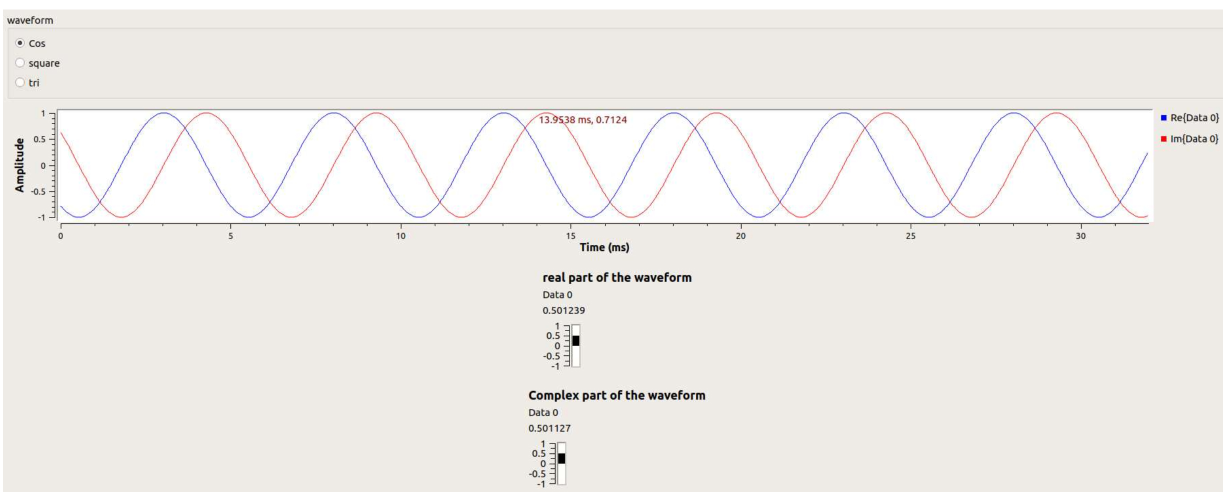


Expt 1(e)

Power for Complex Valued Cosine Waveform

In [10]:
```python
from IPython.display import Image
Image(filename='capture.png')
```

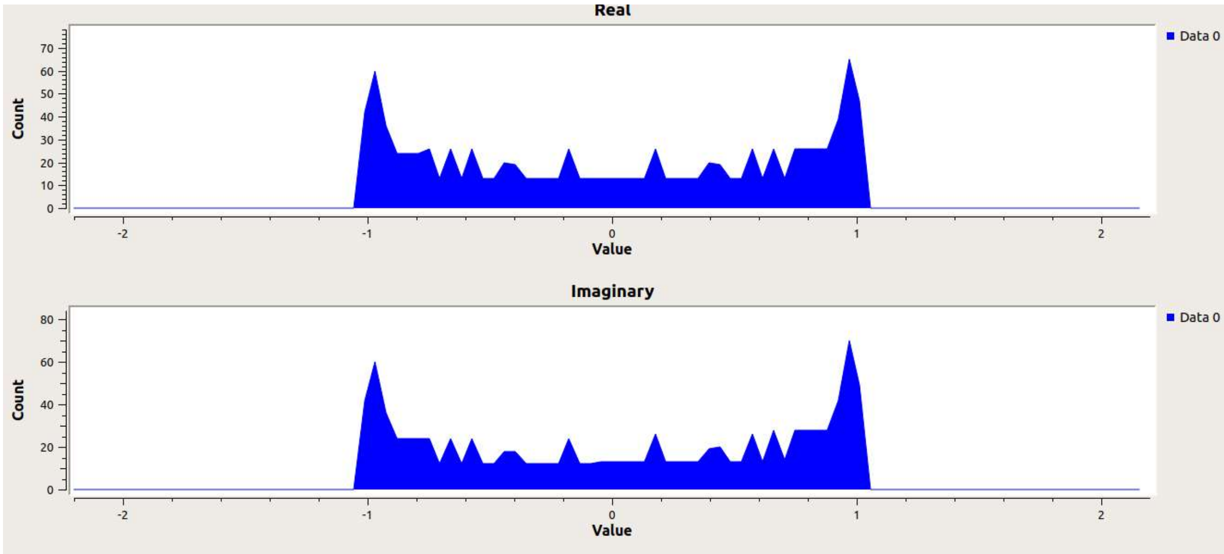
Out[10]:



Average power of the complex = Average part of the real part ~= 50% = 1/2(Total average of the waveform) Total average power of the complex valued waveform = Total average power for the real valued waveform ~= 100%

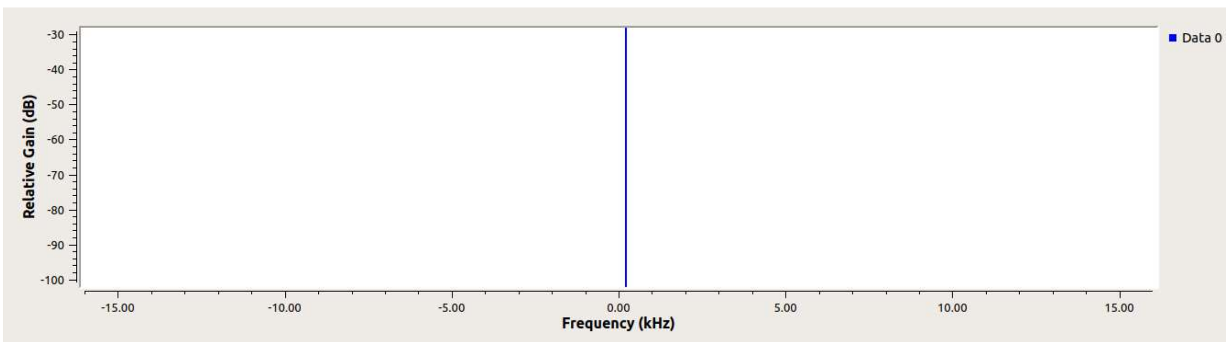
In [11]:
```
from IPython.display import Image
Image(filename='capture.png')
```

Out[11]:



In [12]:
```
from IPython.display import Image
Image(filename='capture.png')
```

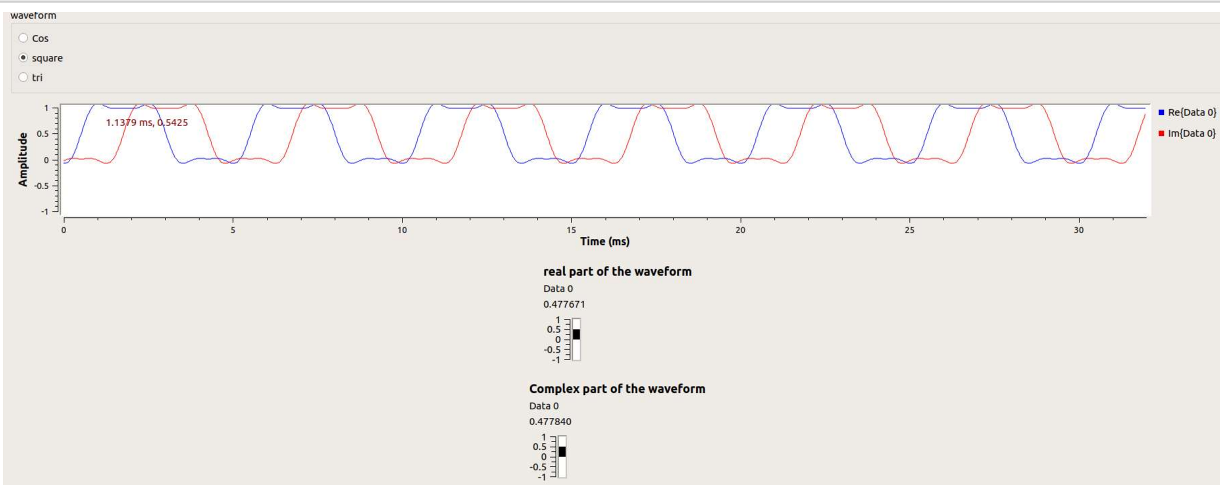
Out[12]:



Power for complex valued Square Waveform

In [13]:
```
from IPython.display import Image
Image(filename='capture.png')
```

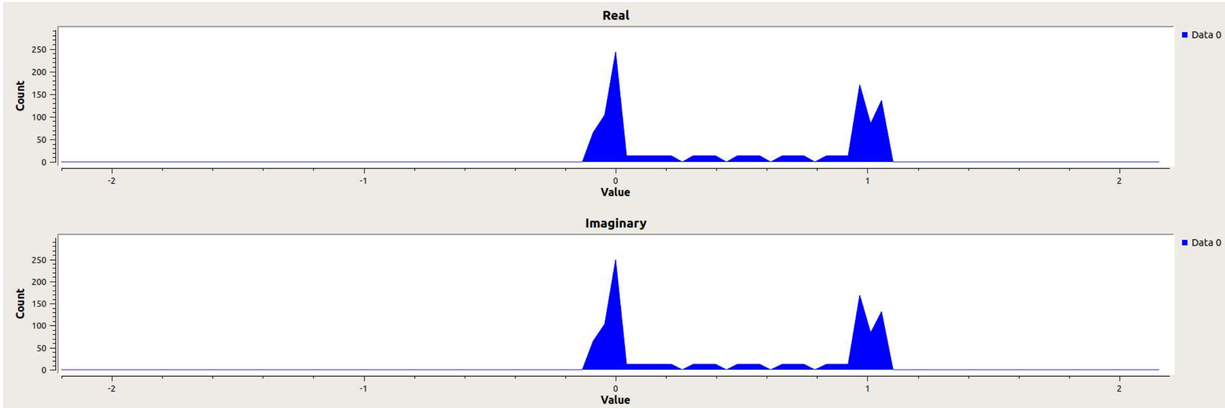
Out[13]:



Average power of the complex = Average part of the real part ~= 48% = 1/2(Total average of the waveform) Total average power of the complex valued waveform = Total average power for the real valued waveform ~= 97%

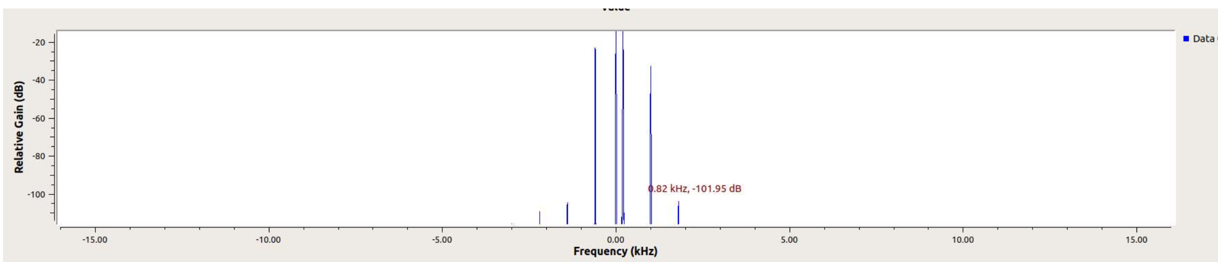
In [14]:
```python
from IPython.display import Image
Image(filename='capture.png')
```

Out[14]:



In [15]:
```python
from IPython.display import Image
Image(filename='capture.png')
```

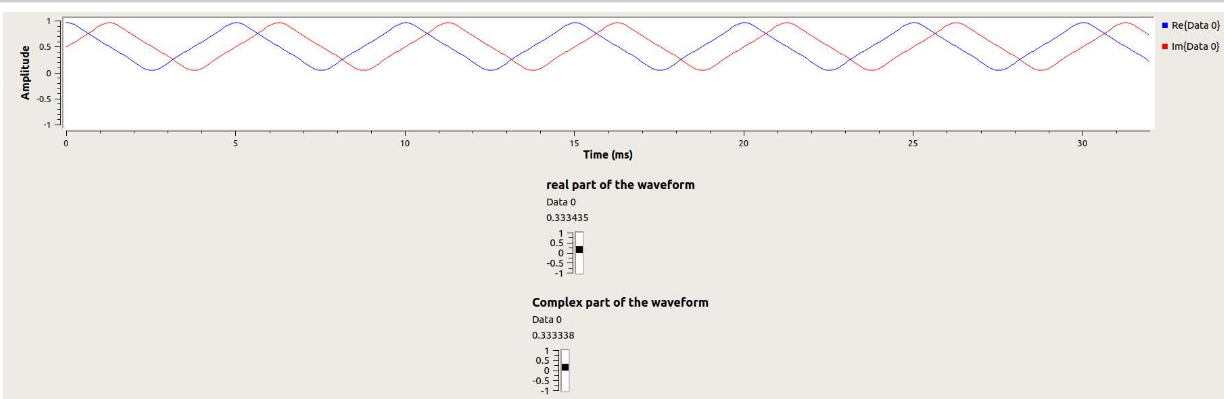
Out[15]:



Power for complex valued Triangular waveform

In [16]:
```python
from IPython.display import Image
Image(filename='capture.png')
```

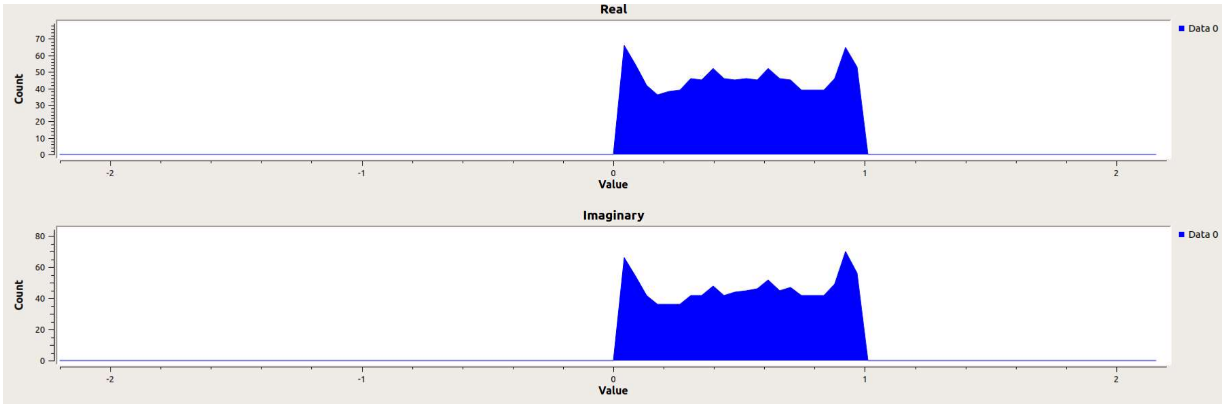
Out[16]:



Average power of the complex = Average part of the real part ~= 33% = 1/2(Total average of the waveform) Total average power of the complex valued waveform = 67% is not equal to Total average power for the real valued waveform ~= 99%. The difference can be explaine by looking at the triangular wave. We observe that the Peak to Peak amplitude is only 0.5 instead of 1.

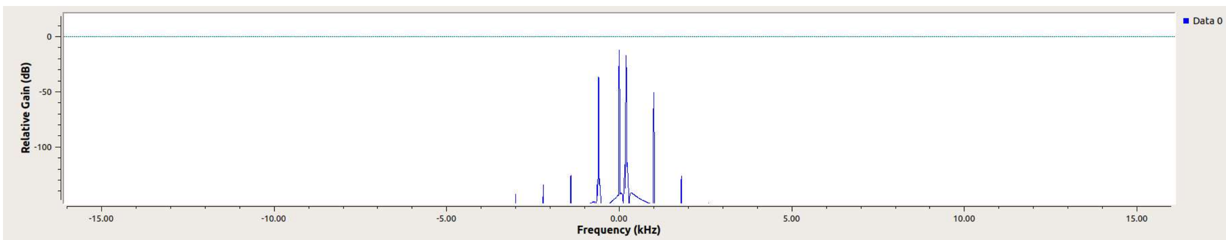
In [17]:
```
from IPython.display import Image
Image(filename='capture.png')
```

Out[17]:



In [18]:
```
from IPython.display import Image
Image(filename='capture.png')
```

Out[18]:

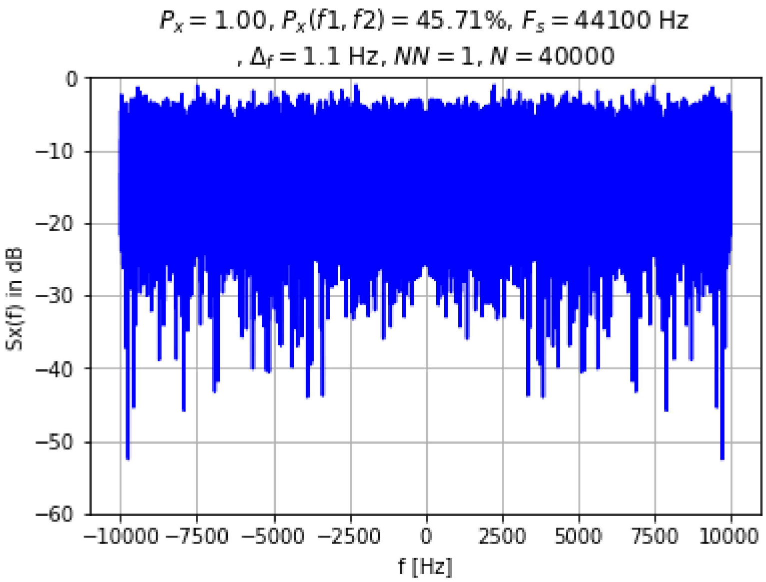


For all three complex valued waveform we observe that the power contained in the real part is the same as the imaginery part. Both are approximately half of the total average power of the signals.

The total average power of the complex valued waveform is nearly the same and comparable with the real valued waveform. We compare with the real valued waveform Power values form Q1(a),(b),(c)

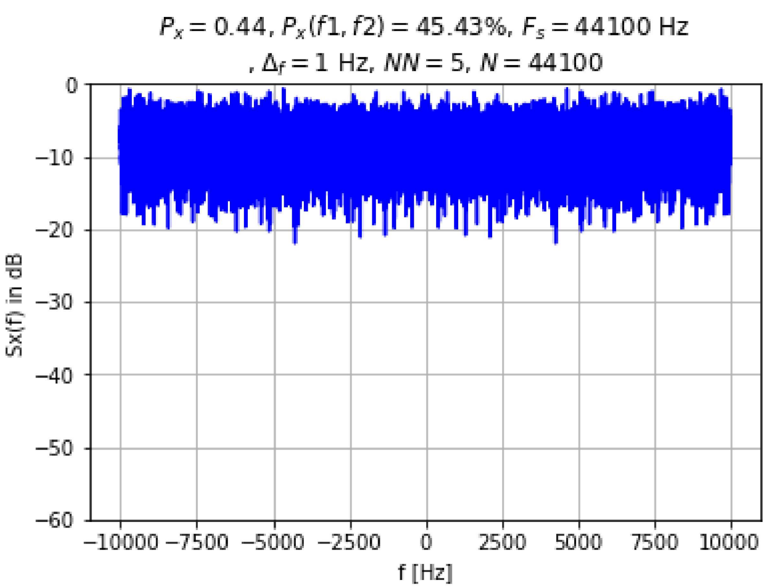
Expt 2(a)

In [3]:
```
run Q2A
```



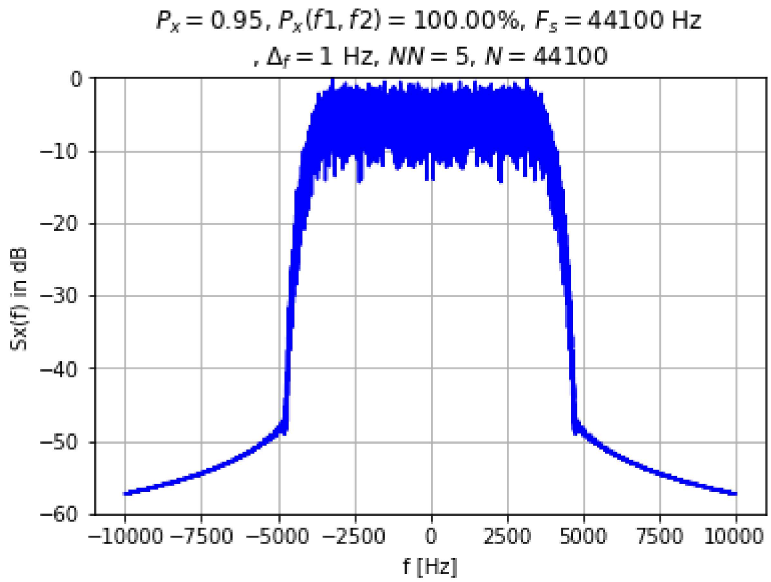
The above plot is the PSD of white gaussian noise

In [2]:
```
run Q2A
```



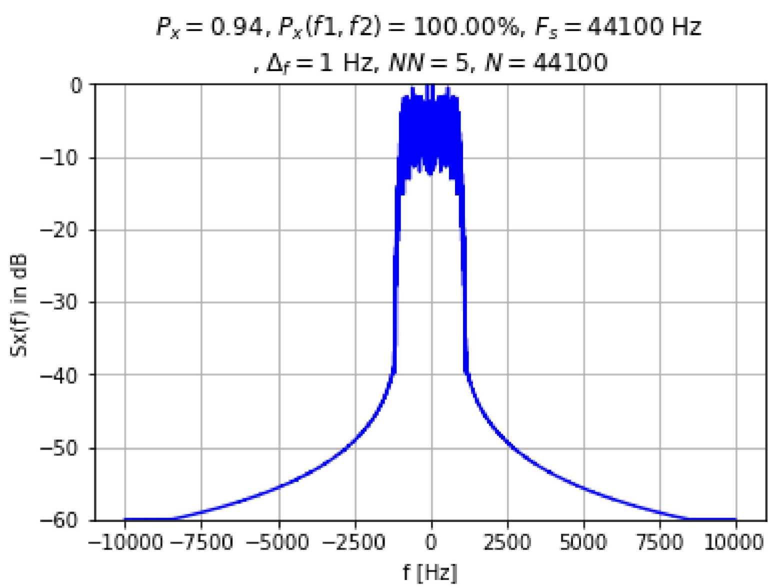
The above plot is the PSD for full bandwidth Gaussian noise

In [3]:
```
run Q2A
```



The above plot is the PSD of bandlimited Gaussian noise with fL = 4000 Hz

In [4]:
```
run Q2A
```



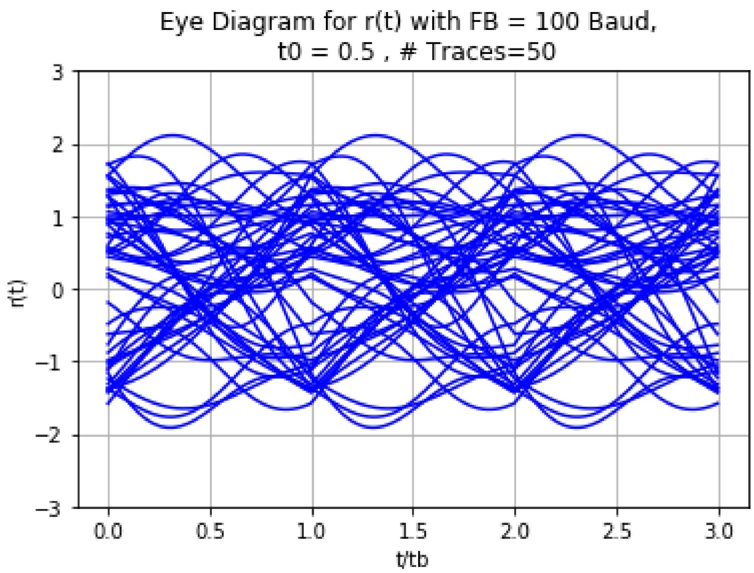
The above plot is the PSD of bandlimited Gaussian noise with fl =1000 Hz

The wav file containing the full bandwidth noise signal had the highest pitched white noise. The bandlimited signal with Fl = 4000Hz was slightly lower pitched than that and the one with Fl = 1000 Hz had the lowest pitched white noise.

Expt 2(b)

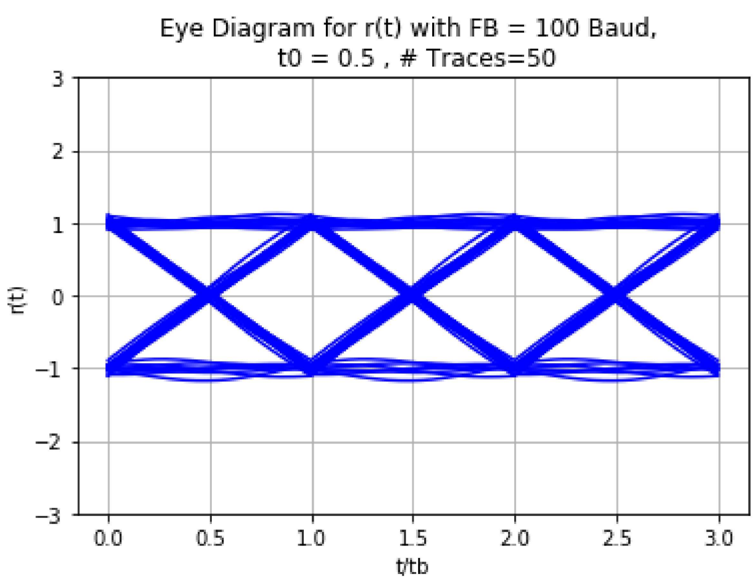
Generated a random polar binary PAM signal s(t) of length 2 sec with triangular p(t). Used FB = 100 and Fs = 44100. Then generated bandlimited Gaussian noise n(t) with cutoff frequency fL = FB (the rule of thumb bandwidth for PAM with triangular p(t)). Used the showeye function to plot the eye diagram of r(t), where r(t) = s(t) + An(t) be the received PAM signal.

In [11]:
```
run Q2B
```



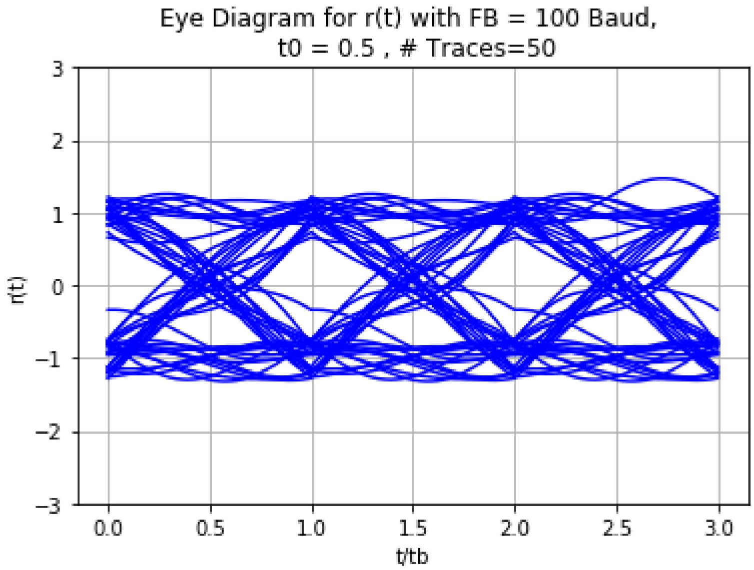
This eye diagram is for A = 0.5

In [12]:
```
run Q2B
```



This eye diagram is for A =0.05

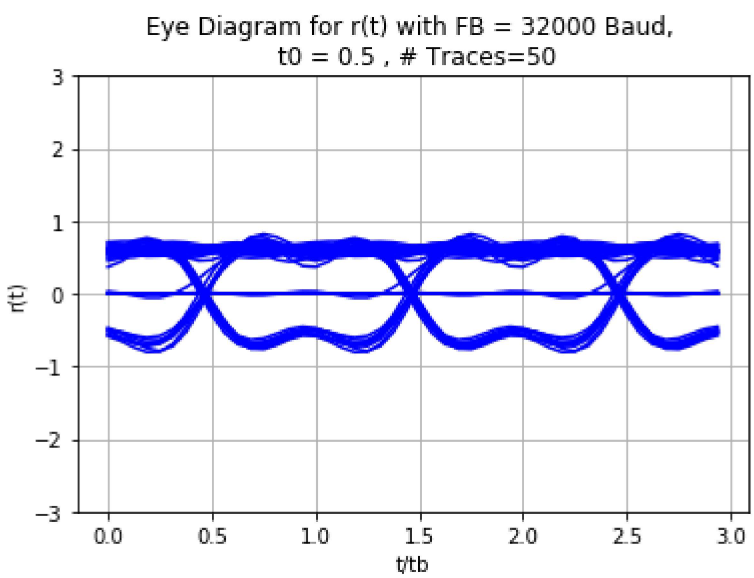
In [20]:
```
run Q2B
```



The first two eye diagrams are examples of respectively 100% and 0% closure. The third eye diagram is somewhere between the two thus we find that 50% closure happens for A = 0.185

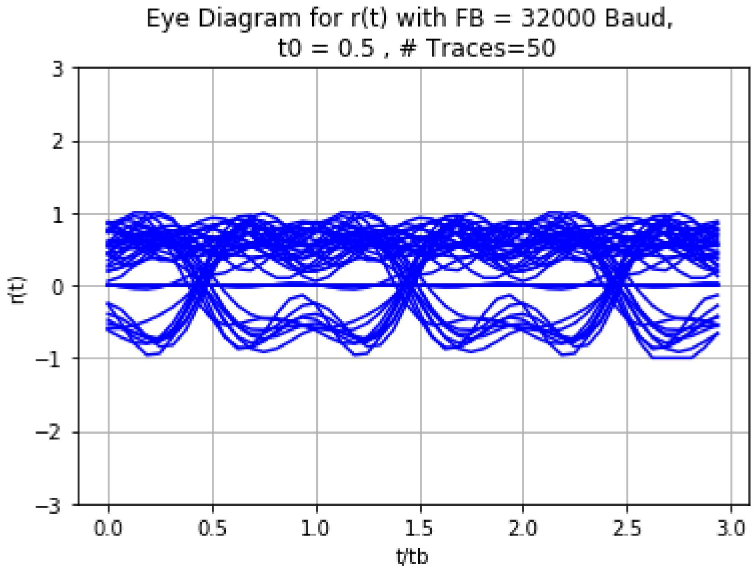
Expt 2 (c)

In [3]:
```
run Q2C
```



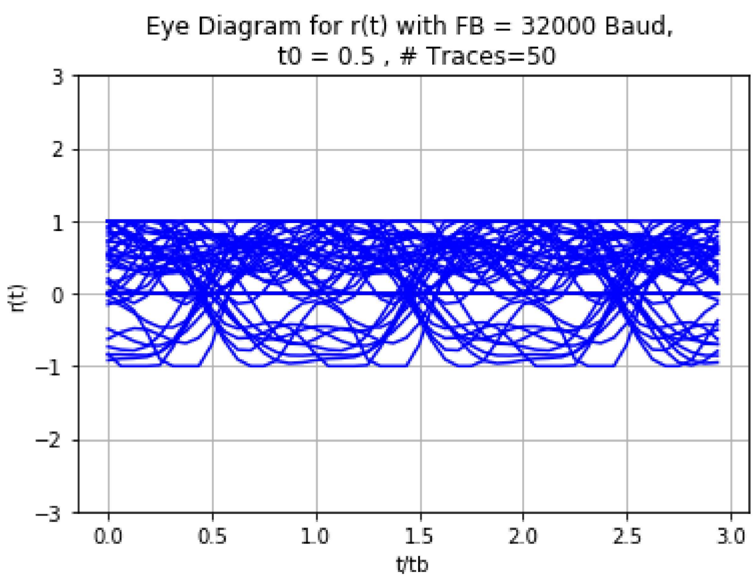
The above eye diagram is for a rectangular PAM with bandwidth (both noise and PAM signal) 2FB and SNR of 20.

In [4]:
```
run Q2C
```



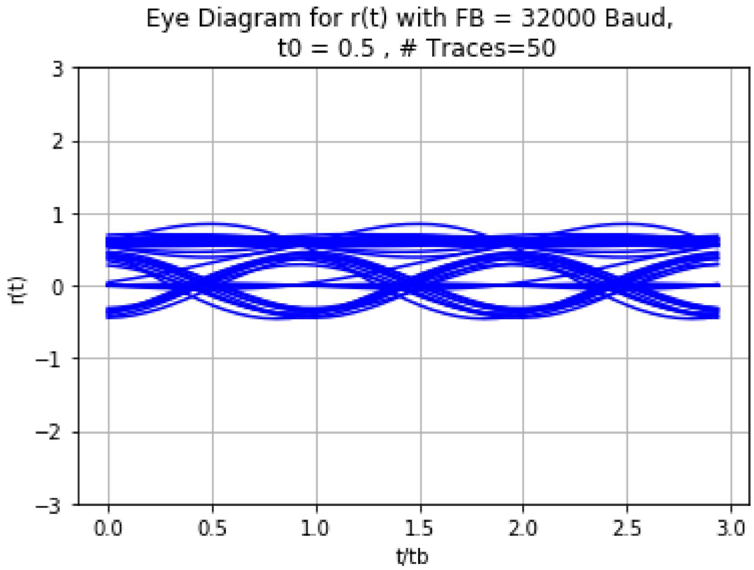
The above eye diagram is for a rectangular PAM with bandwidth (both noise and PAM signal) 2FB and SNR of 10

In [5]:
```
run Q2C
```



The above eye diagram is for a rectangular PAM with bandwidth (both noise and PAM signal) 2FB and SNR of 5

In [7]:
```
run Q2C
```



The above eye diagram is for a rectangular PAM with bandwidth (both noise and PAM signal) FB/2 and SNR of 20. This most closely resembles the eye diagram of the rectangular PAM with bandwidth = 2Fb and SNR of 10

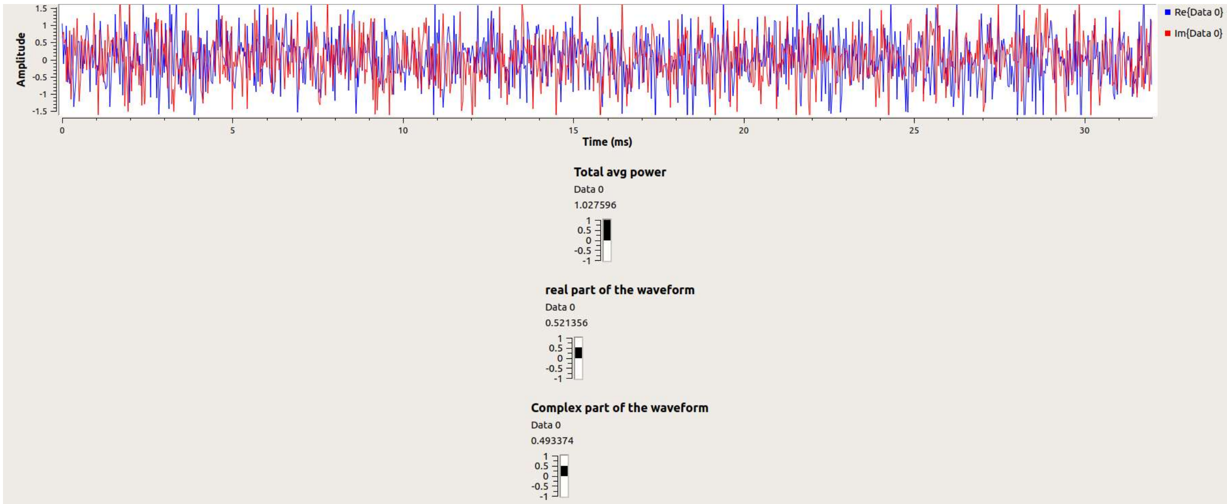
The PAM signal with SNR = 20 db is the one that's the least closed. The closure of the eye diagram keeps increasing as the SNR decreases Which makes sense analytically since the noise added to the signal is increasing as the SNR decreases.

Expt 2(d)

Power of a complex valued gaussian noise in GRC with A = 1, Fl = 16000Hz. The theoretical shape for the amplitude histogram should be a gaussian distribution

In [22]:
```
from IPython.display import Image
Image(filename='capture.png')
```

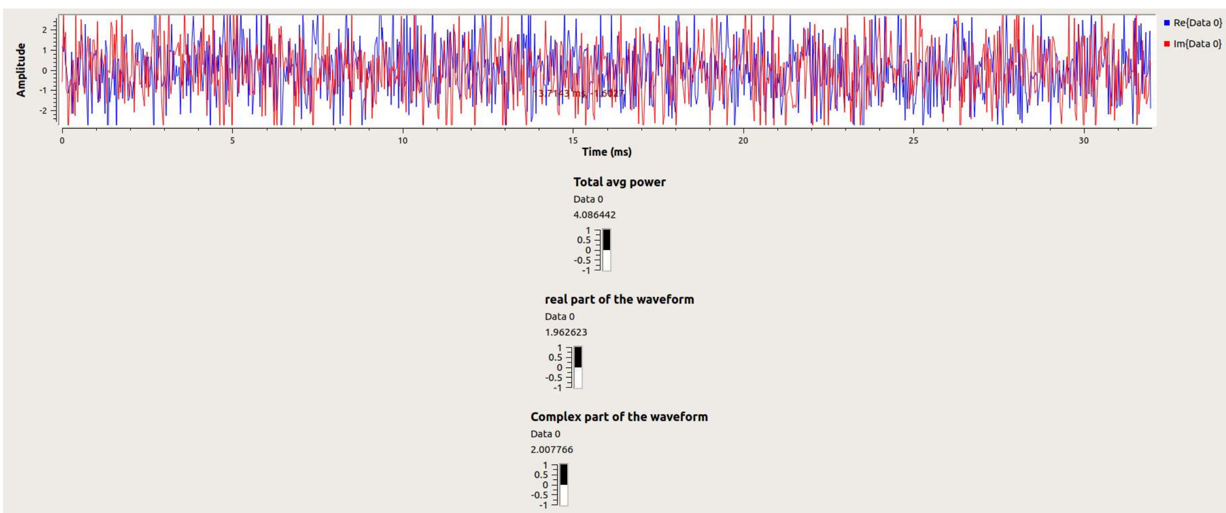
Out[22]:



We observe that the power of the real part is equal to the complex part of the waveform. Both of which are approximately = 50% which is half of the total average power = 100%

In [23]:
```
from IPython.display import Image
Image(filename='capture.png')
```

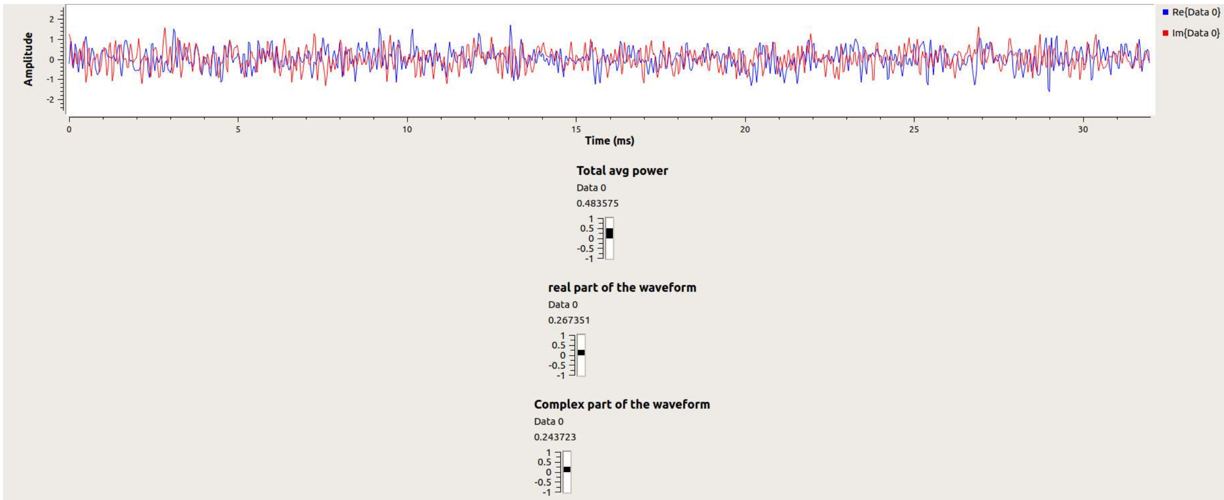
Out[23]:



When we increase the amplitude from 1 to 2 we see that each power measurement has approximately become = 4 (the original measurement).

In [25]:
```
from IPython.display import Image
Image(filename='capture.png')
```

Out[25]:

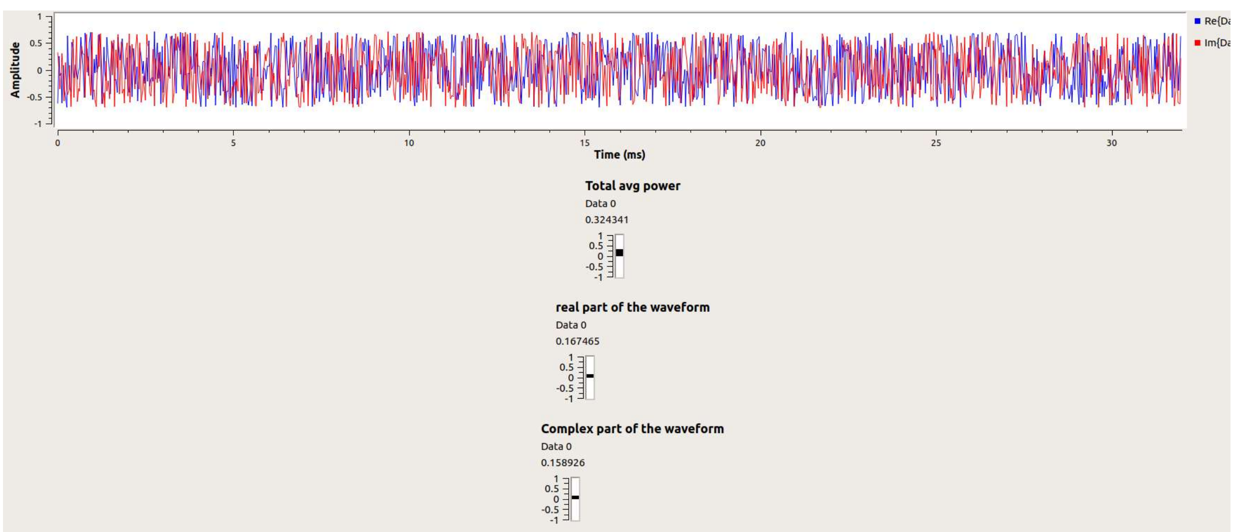


When we decrease the cut off frequency to 8000 from 16000 at amplitude = 1, the power meaurements become = 0.5(the power measurements at fl = 16000)

For uniform noise of A = 1 and Fl = 16000 the power measurements are shown above. The power in the real part = the power in the imaginery part = (1/2) of the total average power. The theoretical shape of the amplitude histogram should be a uniform distribution

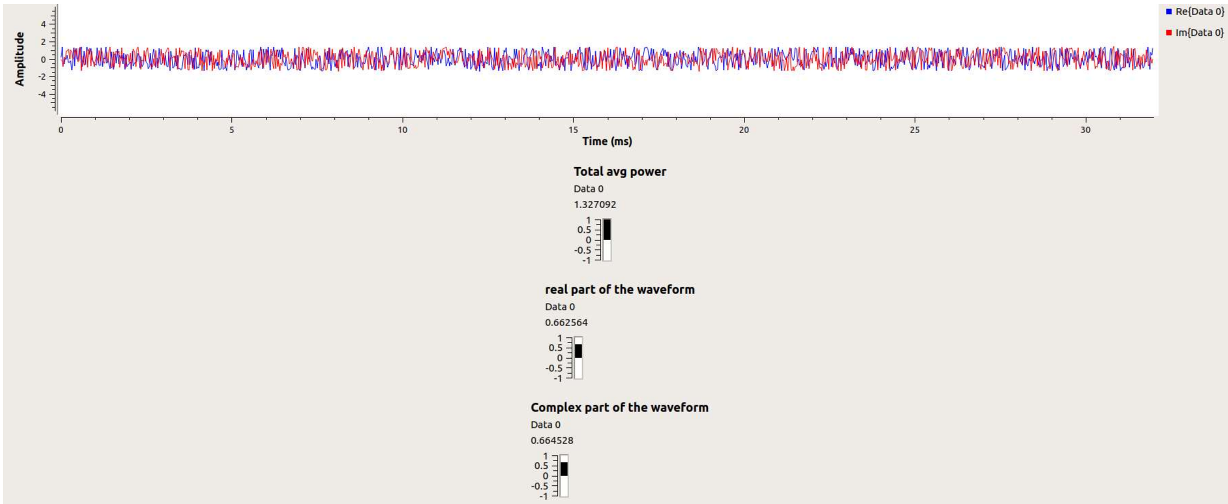
In [26]:
```
from IPython.display import Image
Image(filename='capture.png')
```

Out[26]:



In [27]:
```
from IPython.display import Image
Image(filename='capture.png')
```

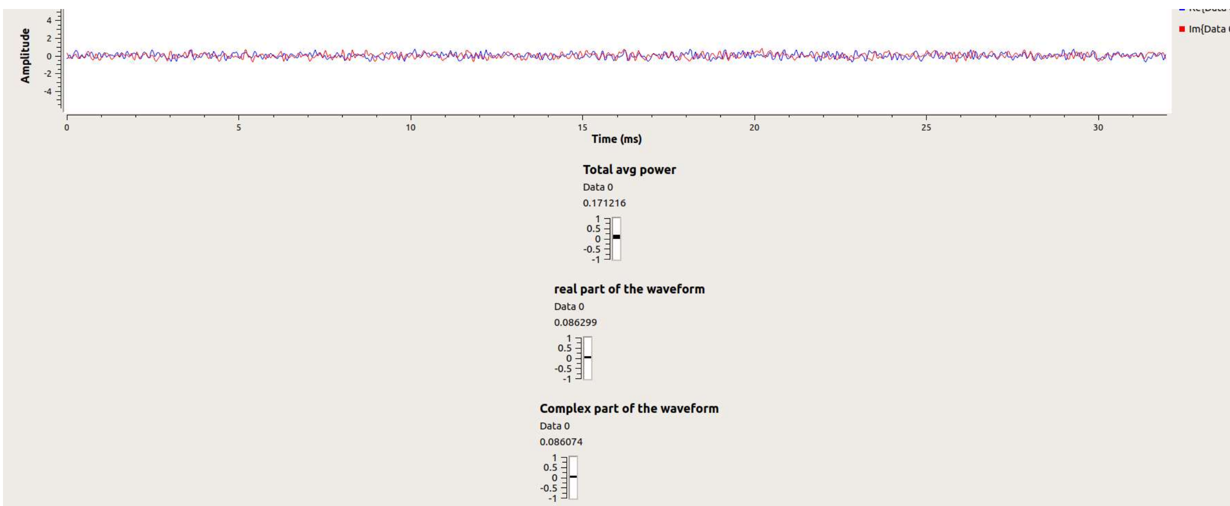
Out[27]:



If the ampitude is increased to A = 2, the power measurements becomes = 4 (times the original)

In [28]:
```
from IPython.display import Image
Image(filename='capture.png')
```

Out[28]:



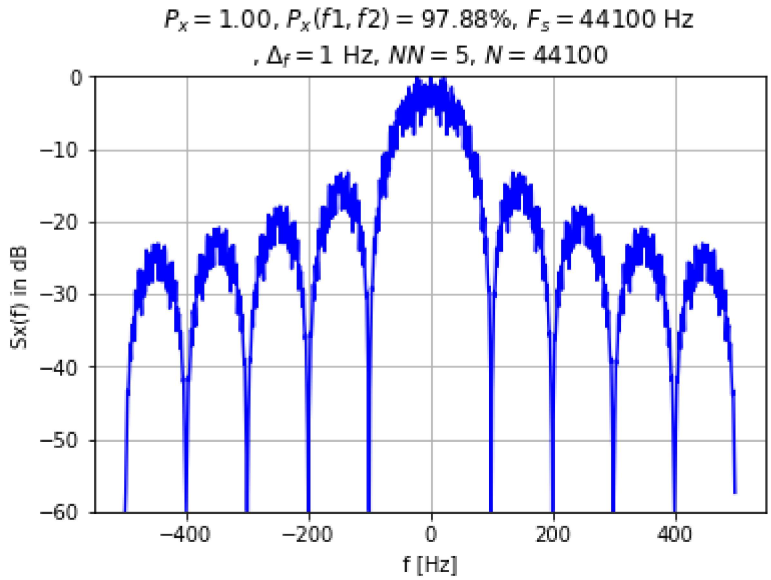
The above measurements are for A = 1 and Fl = 8000. We observe that they are half of the measurements for Fl = 16000

We observe that the the power measurements for uniform noise is less than the power measurements for Gaussian noise for the same cutoff frequency and same amplitude. This can be explained by the fact that while uniform noise is uniformly distributed, Gaussian noise is concentrated around a single frequency. Thus for the same bandwidth the total average power contained in that bandwidth is Less for uniform noise.

Expt 3(a)

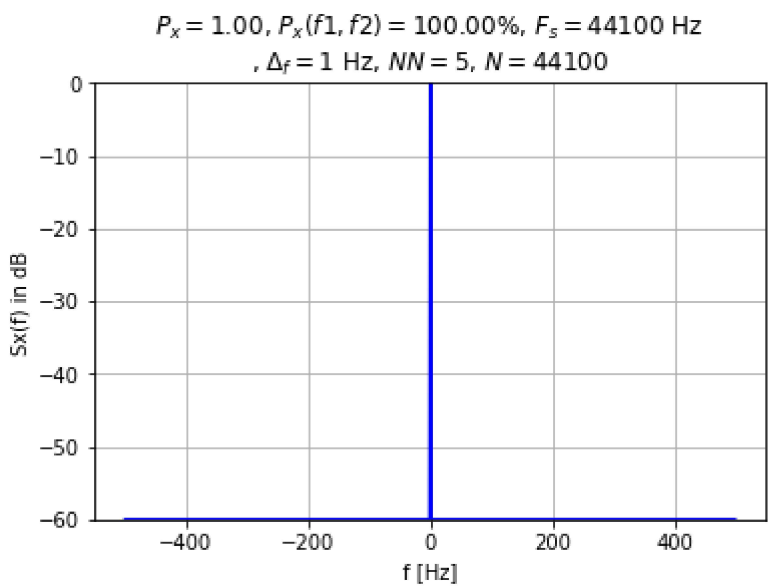
For PAM signal with Rectangular pulse.

In [3]:
```
run Q1C
```



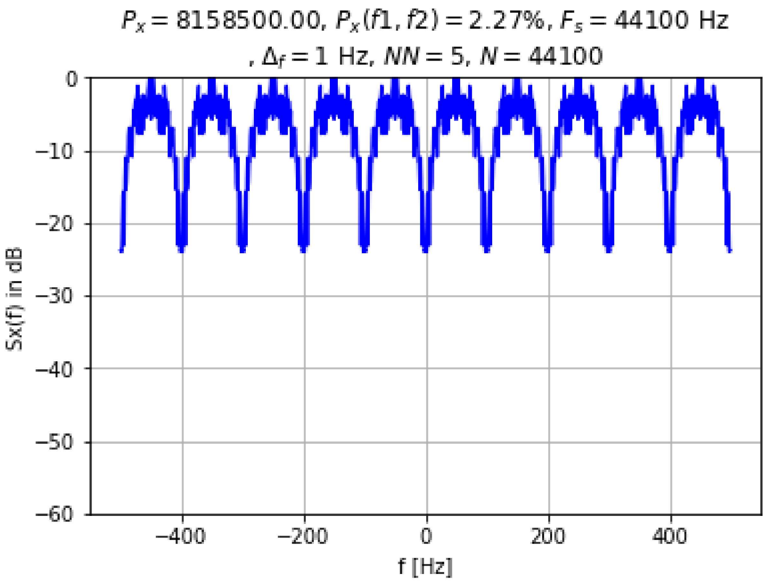
xt = st^2 for rectangular pulse PAM signal

In [2]:
```
run Q1C
```



Derivative of the signal

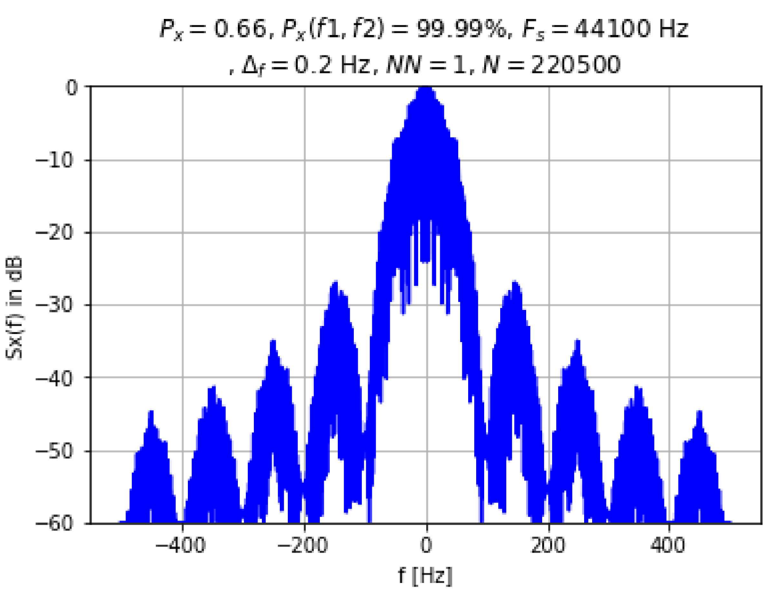
In [2]:
```
run Q1C
```



By observing the plot for the derivative of the rectangular pulse, We see that it has distinct spectral lines at Fb which can be used to synchronize the receivers. By looking at the rectangular signal we see that they are not present for xt but present for the derivative of xt

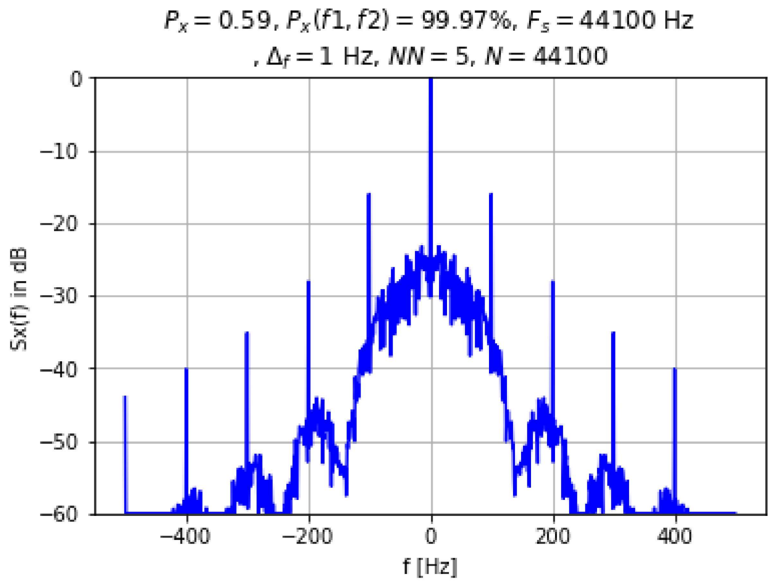
PAM signal with triangular pulse

In [19]:
```
run Q1C
```



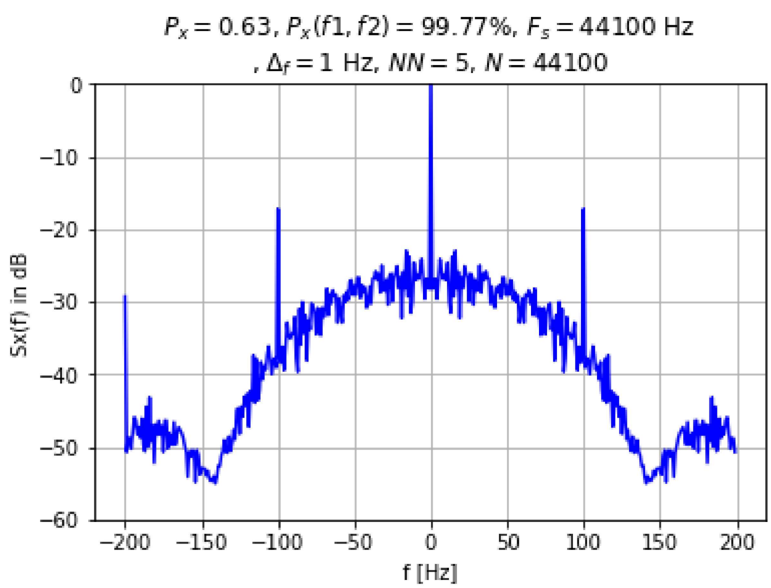
Triangular pulse PAM squared

In [3]:
```
run Q1C
```



Derivative of the PAM triangular signal

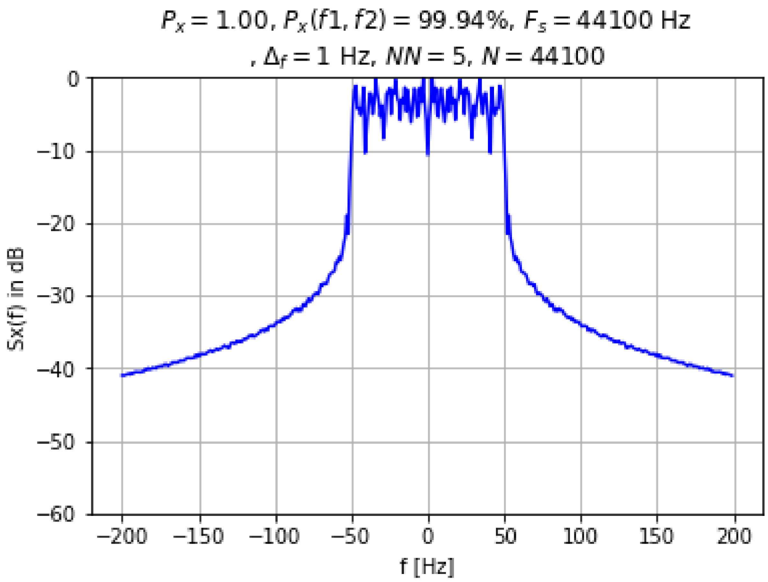
In [4]:
```
run Q1C
```



For the triangular signal we observe distinct spectral lines for bot xt^2 and the derivative for Xt. Hence both can be used for synchronising the receiver

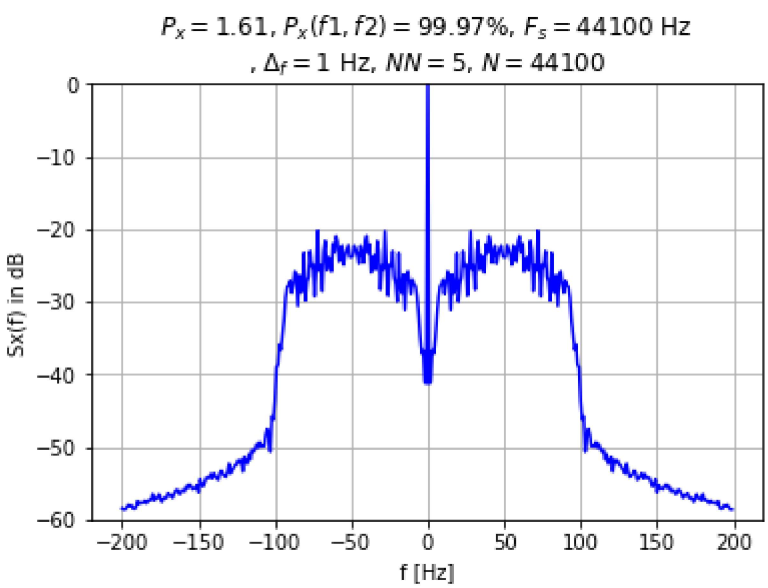
For PAM signal with sinc pulse

In [2]:
```
run Q1C
```



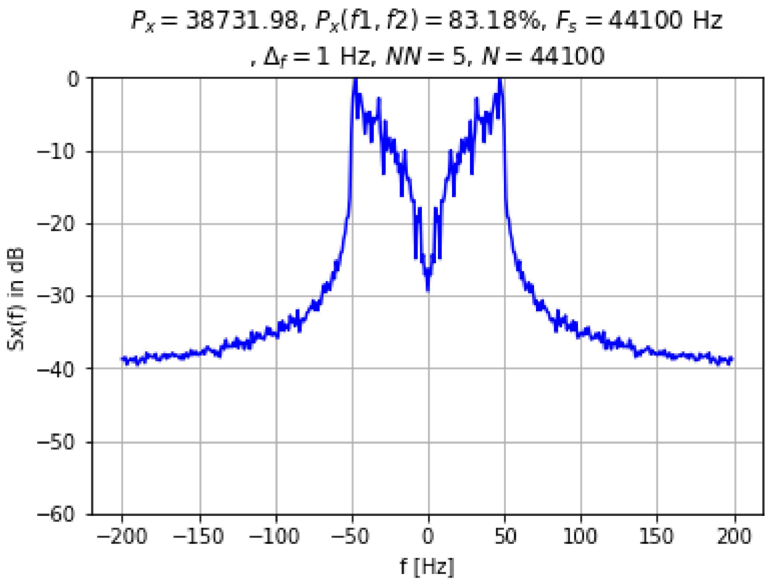
For sinc pulse squared

In [2]:
```
run Q1C
```



Derivative of sinc pulse

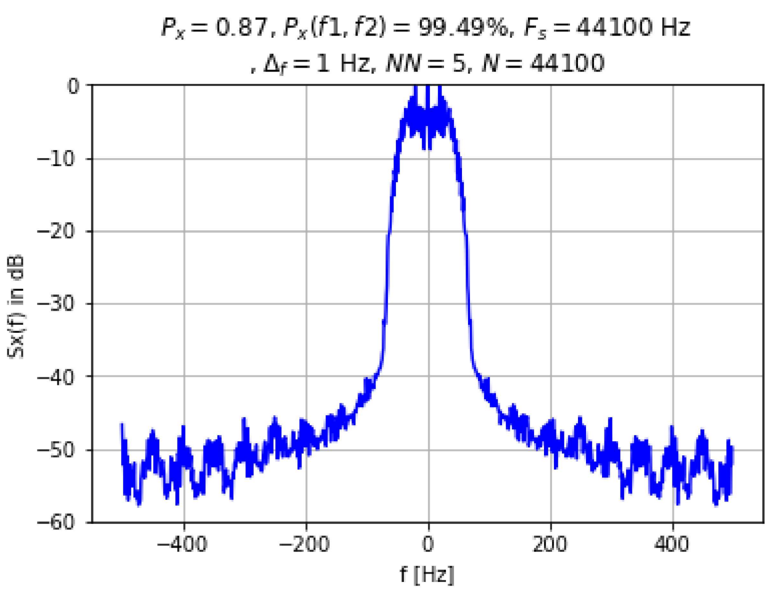
In [2]:
```
run Q1C
```



From observing the derivative for the sinc pulse we can tell that there are distinctive spectral lines at Fb/2. We can use this plot to synchronize the receiver

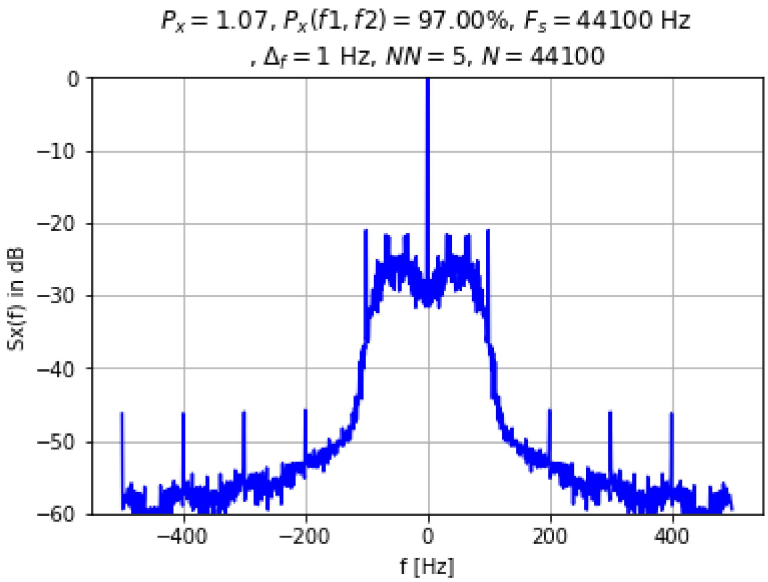
For RCf PAM signal

In [2]:
```
run Q1C
```



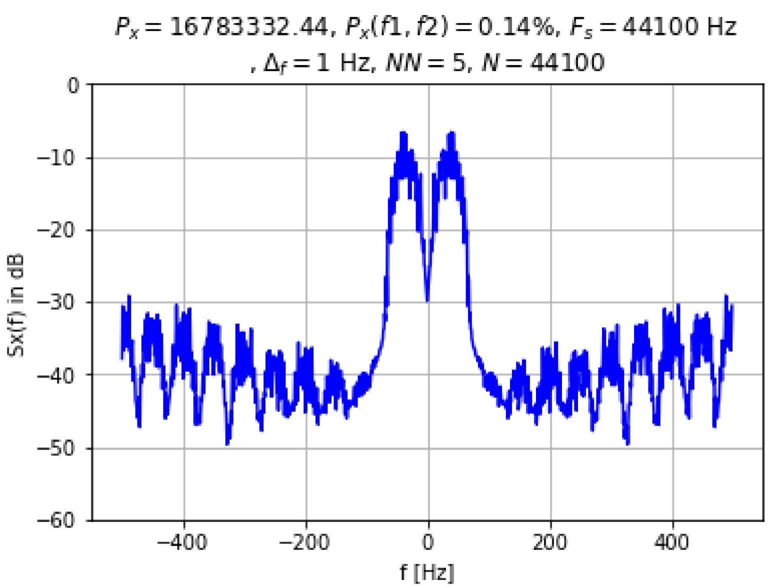
For RCf PAM signal squared

In [2]:
```
run Q1C
```



For the derivative of the RCf PAM signal

In [2]:
```
run Q1C
```

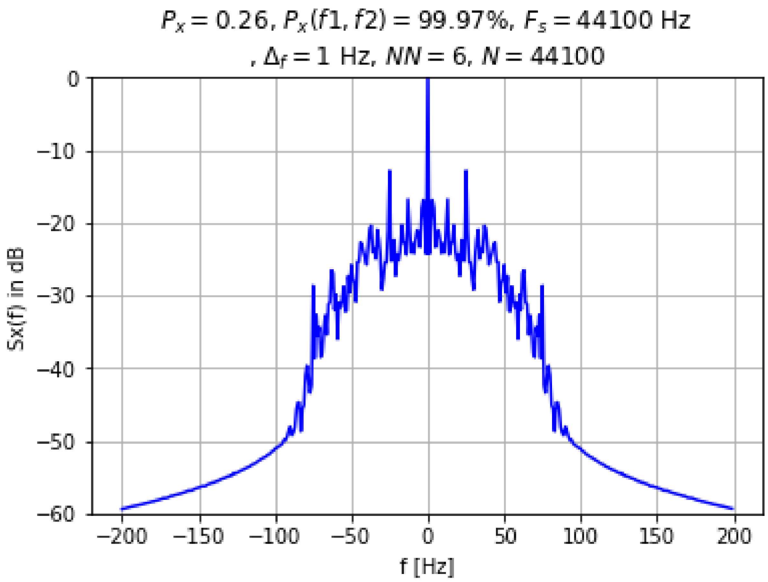


For the RCf signal we observe distinct spectral lines at Fb for the plot for xt^2. This plot can be used to synchronize the receiver

Expt 3(b)

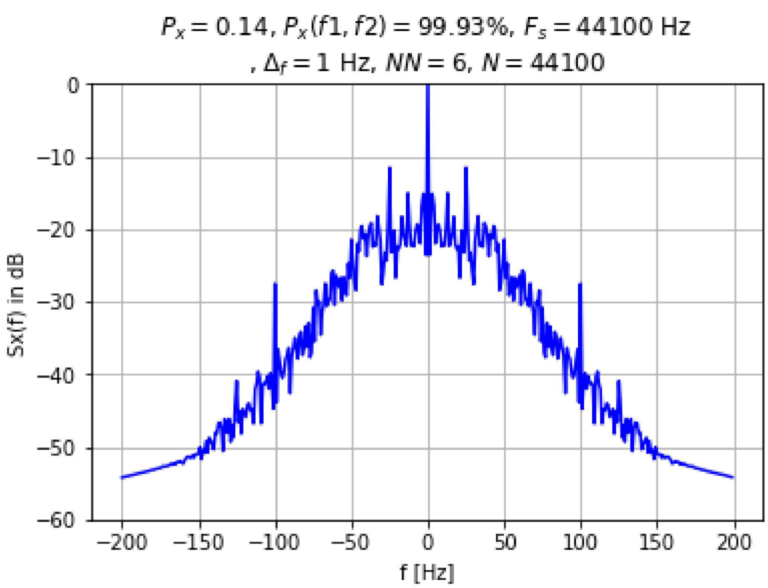
PSD plot for the square of signal "pamsig401"

In [2]:
```
run Q3B
```



PSD plot for the 4th power of signal "pamsig401"

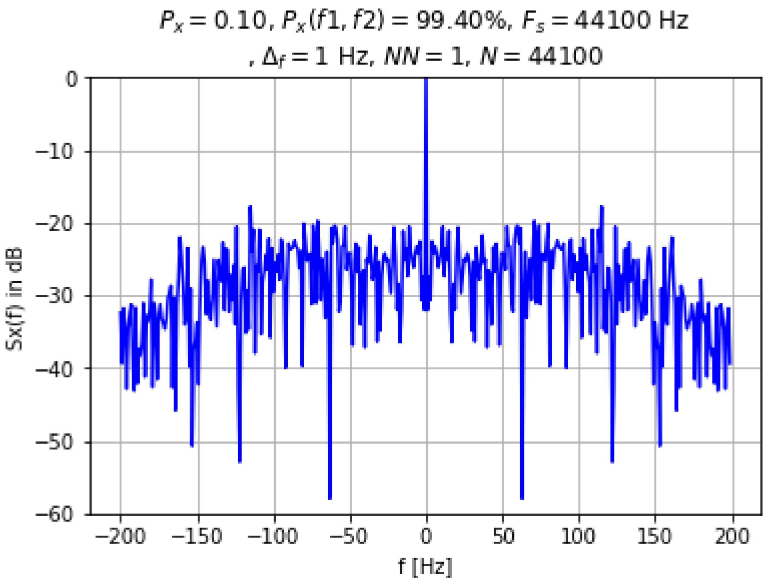
In [2]:
```
run Q3B
```



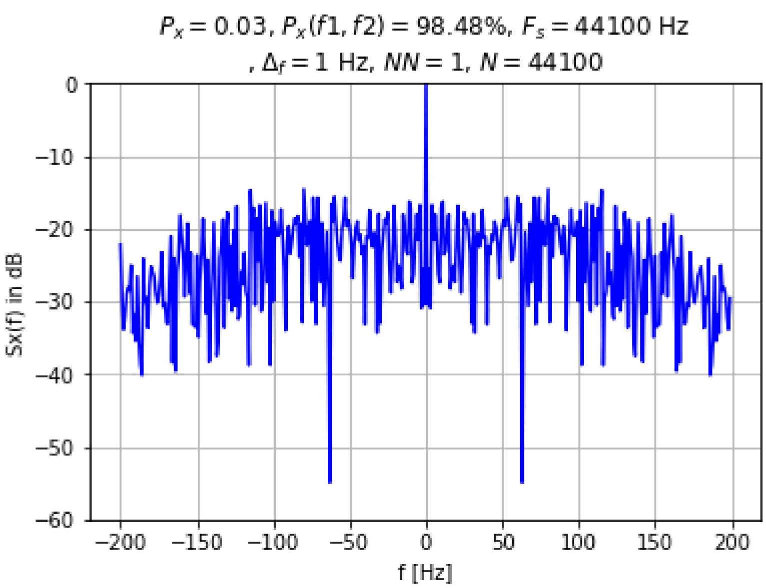
We observe that there is a useful distinctive spectral lines at Fb for the plot of the 4th power of the signal.

PSD plot for the square of pamsig402

In [2]:
```
run Q3B
```



In [2]:
```
run Q3B
```

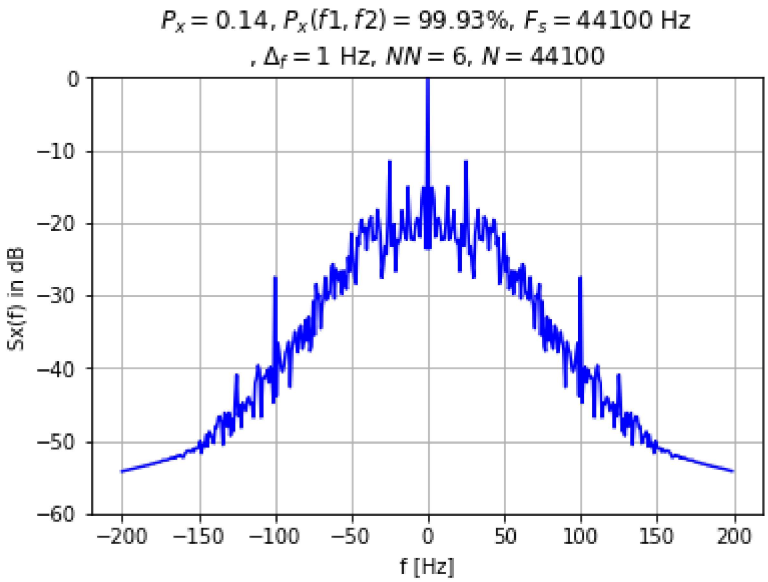


We observe that there is a useful distinctive spectral lines at Fb for the plot of the 2nd power of the signal.

PSD plot for the second power of the signal pr1sig401

In [3]:

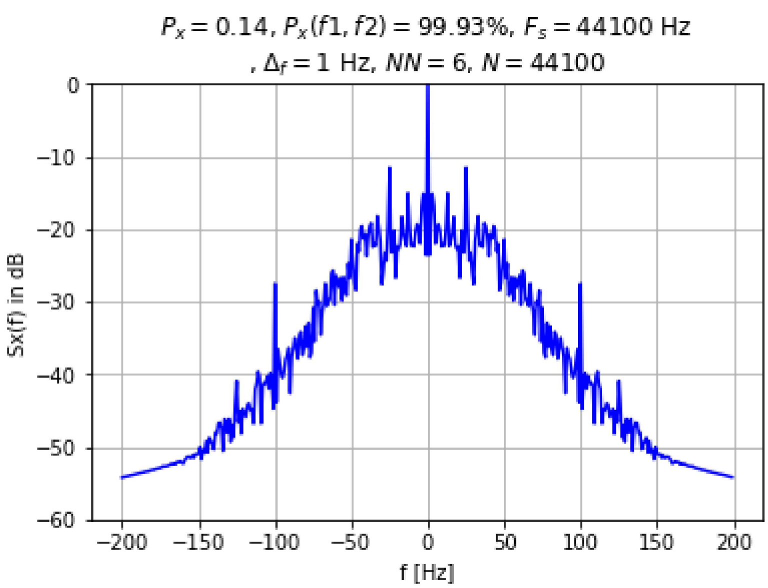
```
run Q3B
```



PSD plot for the fourth power of the signal

In [2]:

```
run Q3B
```



We observe that there is a useful distinctive spectral lines at Fb for the plot of the 2nd and the 4th power of the signal.