ANALOG SIGNAL ESTIMATION USING SOFTWARE DEFINED RADIO (SDR's)

SUMMER PROJECT REPORT

Submitted by

S VISHWAJEITH

(2021105061)

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY: CHENNAI 600 025

JULY- OCTOBER 2023

COLLEGE OF ENGINEERING-GUINDY

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that the summer project report “ANALOG SIGNAL ESTIMATION USING SOFTWARE DEFINED RADIO” is the bonafide work of “ S VISHWAJEITH (2021105061)”, who carried out the project work for EC-5512, 5th semester, Summer Project in the duration July - October 2023 under my supervision. Certified further that to the best of my knowledge the work reported here in does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

DR. M. MEENAKSHI
HEAD OF DEPARTMENT
Department Of Electronics &
Communication Engineering
College Of Engineering, Guindy
CHENNAI 600 025.

SIGNATURE

DR. LAXMIKANDAN
PROJECT SUPERVISOR
ASSOCIATE PROFESSOR
Department of Electronics &
Communication Engineering
College Of Engineering, Guindy
CHENNAI 600 025.

ACKNOWLEDGEMENT

I express my sincere gratitude to Dean, Dr.L.Suganthi, Professor, College of Engineering,Guindy for her support throughout the project.

I expresss my sincere gratitude to my Head of the Department Dr.M.Meenakshi, Professor, Department of Electronics & Communication Engineering for her enthusiastic encouragement and support throughout the project.

I present my sincere thanks and gratitude to my project supervisor and co-ordinator, Dr. LAXMIKANDAN , Associate Professor, Department of Electronics and Communication Engineering for her whole hearted support, patience, valuable guidance, technical expertise and encouragement in my project.

I thank all the teaching and non-teaching staff of the Department of Electronics and Communication Engineering, for their kind help and co-operation during the course of my project.

TABLE OF CONTENTS

INTRODUCTION:

Analog signal estimation is a fundamental task in the field of software defined radio (SDR) that plays a pivotal role in the modern wireless communication systems. This project presents an innovative approach to analog signal estimation utilizing AMITEC (Analog signal estimation using software defined radio and GNU companion software) framework. This integration of SDR technology with GNU companion software provides a versatile and powerful platform for real time analog signal estimation.

The proposed framework leverages the flexibility and adaptability of SDR hardware, allowing for the reception and processing of a wide range of analog signals. GNU Companion, a visual programming tool for GNU Radio, enhances the user friendliness and accessibility of the estimation process. By combining these technologies, the AMITEC framework facilitates rapid prototyping and deployment of analog signal estimation solutions.

Key components of the AMITEC framework include signal acquisition, preprocessing, feature extraction, and estimation algorithms. Real time signal acquisition is achieved using SDR hardware, offering exceptional flexibility in terms of signal bandwidth and frequency. Preprocessing modules are implemented within GNU Companion to filter, downsample and condition the acquired signals. Feature extraction techniques are applied to capture relevant characteristics of the analog signals, facilitating the subsequent estimation process.

To demonstrate the efficacy of the AMITEC SDR framework, we present our project in which it is applied to estimate various analog signals, including modulation and demodulation schemes, block estimations and output analysis. The result showcase the accuracy, flexibility and real time capabilities of the framework, making it a valuable tool for researchers, engineers and practitioners in the field of SDR’s

In conclusion, the AMITEC SDR framework represents a significant advancement in the domain of analog signal estimation using software defined radio and GNU Companion software. Its integration of versatile SDR hardware and user-friendly GNU companion interface empowers users to efficiently and effectively estimate a wide range of analog signals in real time, making it a valuable asset in the development of modern wireless communication systems.

WEEK 1

A) REQUIREMENTS FOR THE PROJECT

Signal estimation in this project have used the following hardware and software:
i) Software defined radio (AMITEC source and sink)
ii) Dipole Antennas : 2 Nos
iii) Signal hound
iv) GNU Radio software with Ubuntu installed.
v) Spike Software

i) SOFTWARE DEFINED RADIO:

Definition: SDR is a radio communication system where traditional hardware components like mixers, filters, amplifiers, and demodulators are replaced or greatly supplemented by software processing. Here we use AMITEC SDR's for our project. Here the transmitting and receiving frequency ranges from 0.4Ghz to 4Ghz. The RF out is taken from the transmitter part and RF in is taken from the receiver part of the setup.

Key Components:

RF Front-End: This is the hardware responsible for receiving or transmitting radio signals and converting them to/from digital signals.

ADC (Analog-to-Digital Converter): Converts analog RF signals into digital data.

DAC (Digital-to-Analog Converter): Converts digital data into analog RF signals.

Processor (FPGA or CPU): The heart of the SDR system where signal processing algorithms run.

Applications:

Wireless Communication: SDRs are used in mobile phones, Wi-Fi, Bluetooth, and other wireless standards.

Military: They are employed in military communications and electronic warfare.

Aerospace: SDRs are used in satellite communications and aircraft radios.

Research and Education: SDRs are valuable tools for experimentation and learning.

Challenges:

Signal Quality: The quality of the received signals can be affected by the SDR hardware and software.

Processing Power: Complex signal processing may require significant computational resources.

RF Front-End Limitations: SDRs depend on the quality of their RF front-end components.

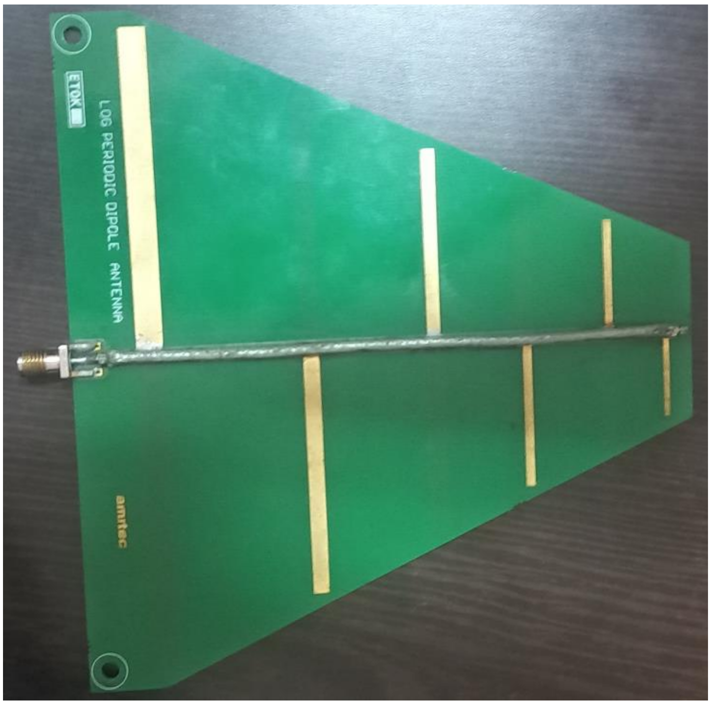
Future Trends:

5G and Beyond: SDR plays a crucial role in the development of advanced wireless communication standards.

IoT and Edge Computing: SDRs are used for low-power, wide-area IoT communication.

Machine Learning Integration: Combining SDR with machine learning can enhance signal processing capabilities.

ii) <u>DIPOLE ANTENNAS:</u>

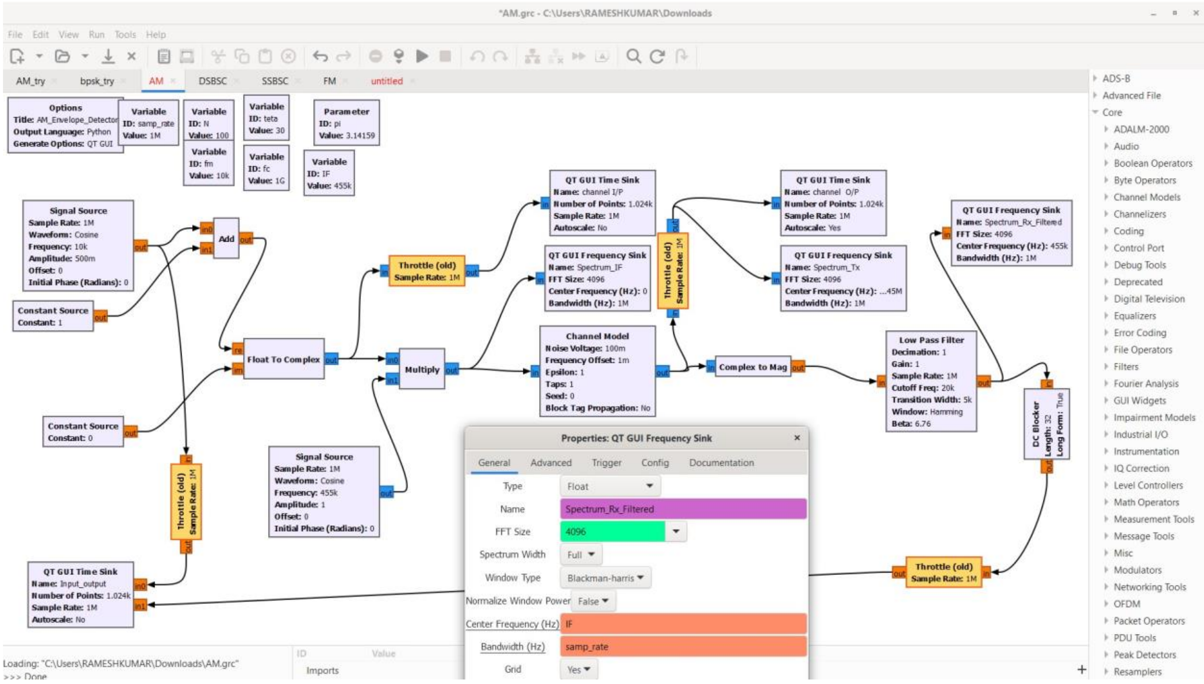


The Yagi-Uda antenna is mostly used for domestic purpose. However, for commercial purpose and to tune over a range of frequencies, we need to have another antenna known as the Log-periodic antenna. A Log-periodic antenna is that whose impedance is a logarithamically periodic function of frequency.

The frequency range, in which the log-periodic antennas operate is around 30 MHz to 3GHz which belong to the VHF and UHF bands.

iii) <u>SIGNAL HOUND:</u>

B) OPERATIONS ON GNU RADIO:



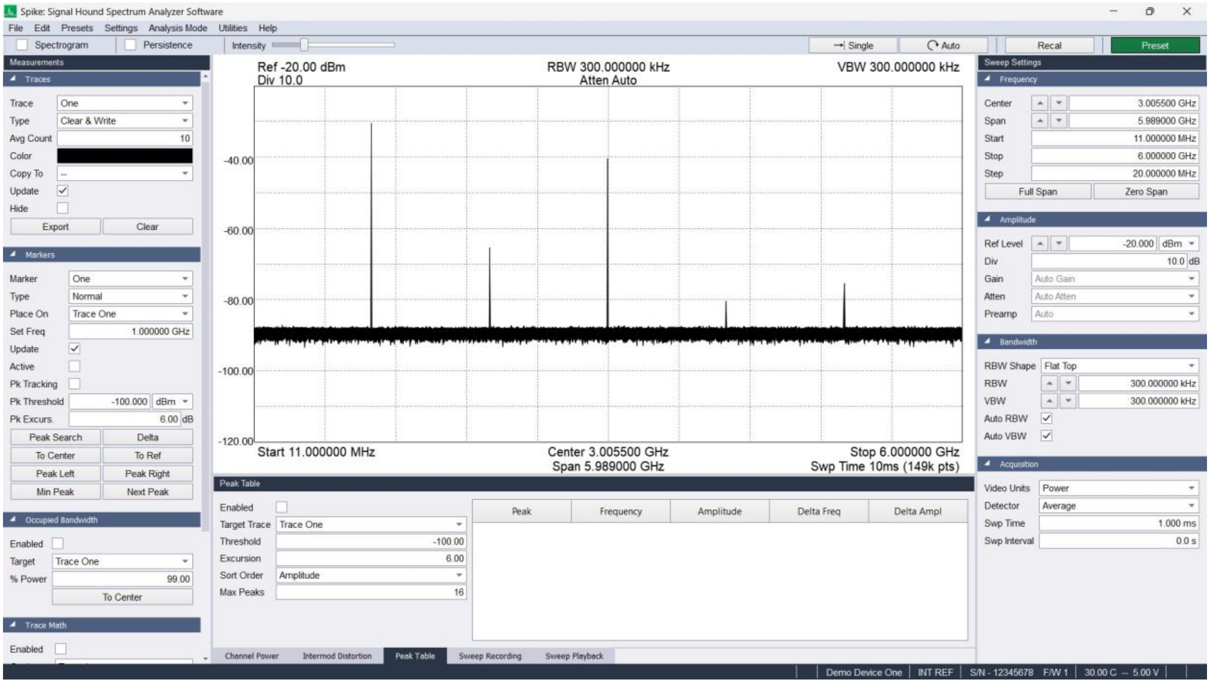
This is the general outlook of GNU Companion software. All the blocks for this current project is taken from the search bar. We can also create our own block by writing a python code and saving the code in gnu file.

The block specification can be edited by double clicking on the block. Here we have clicked on the GUI frequency sink block and the property box is shown.

All blocks can be connected according to our requirement by clicking the output port of a block and immediately clicking the input port of the required block.

Variables can be assigned to avoid repeated input from the user. Finally play/resume can be used to test the diagram.

C) OPERATIONS ON SPIKE:

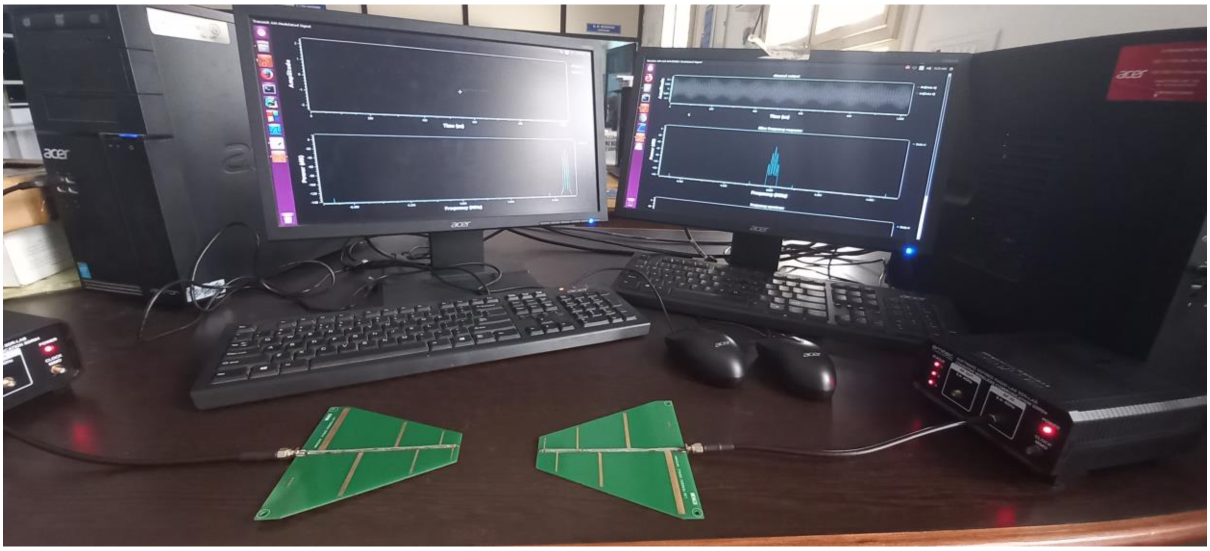


After configuring the signal hound with the spike software, generally appears like given above. If we have given a frequency component of 2GHz (generally used in upcoming experiments), to see the spectrum, we adjust the sweep settings.

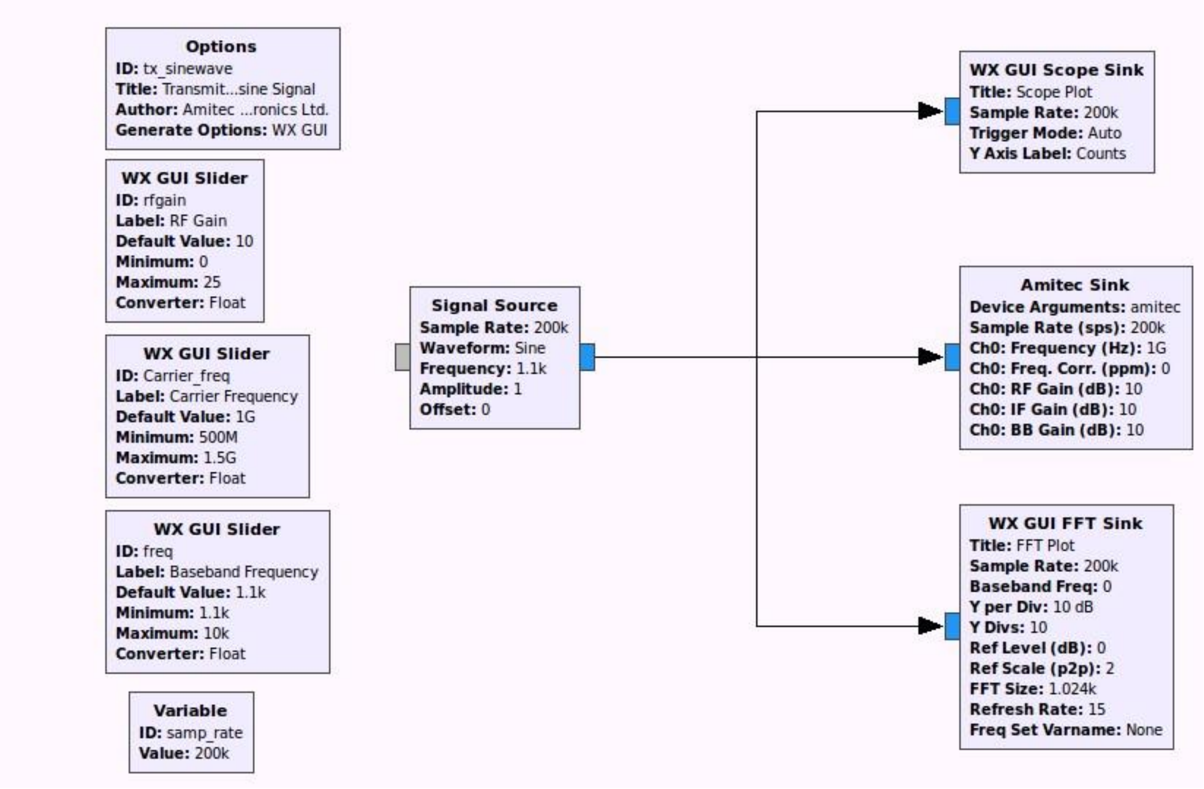
We give 2GHz as the center frequency and span as 0.1GHz and start at 1.5GHz. To adjust in the y axis db scale, we reduce the reference level and divisions per unit to view the spectrum.

To observe more than 2 peaks, to note down the readings, we use cursors/markers tab.

D) OVERALL SETUP FOR SIGNAL ANALYSIS:



E) SINUSOIDAL SIGNAL ESTIMATIONS USING SDR AND HOUND:

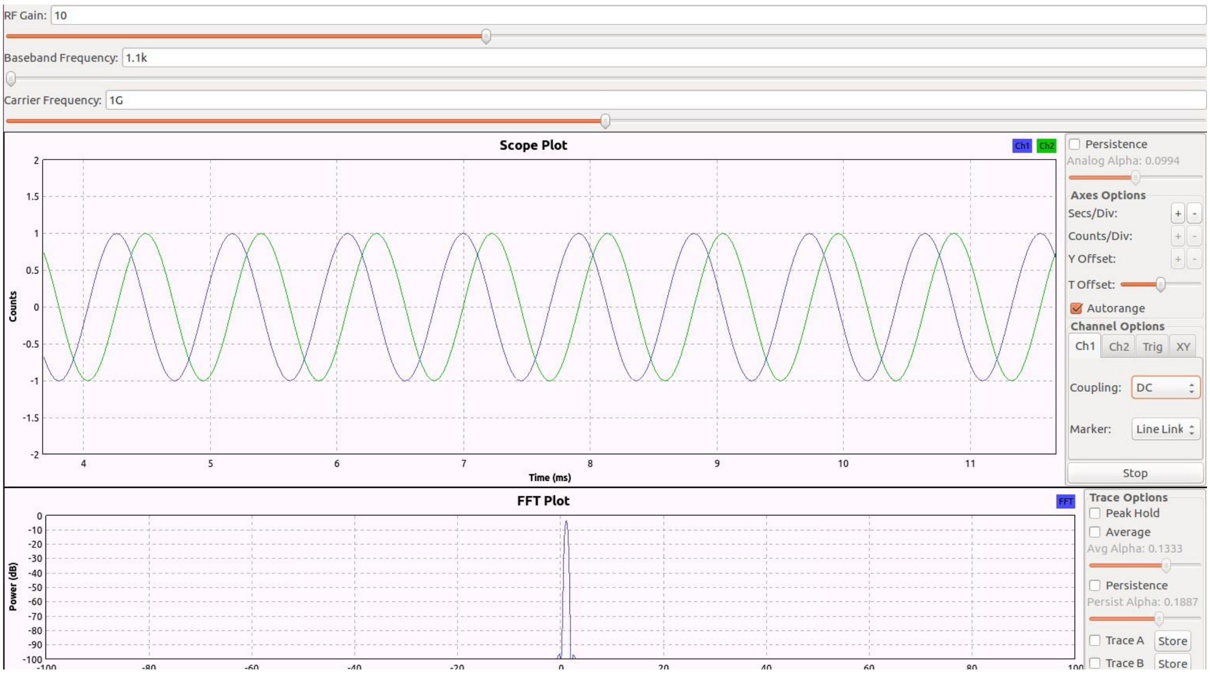


Here we use signal source with 200Khz sampling rate which produces a sine signal at a frequency of 1.1Khz and amplitude 1 volts. Signal source main objective is to produce the message signal which will be further modulated.

The WX GUI scope sink takes the output of the signal source, takes a set of complex streams and plots them in time domain with a similar sampling rate of that of signal source..

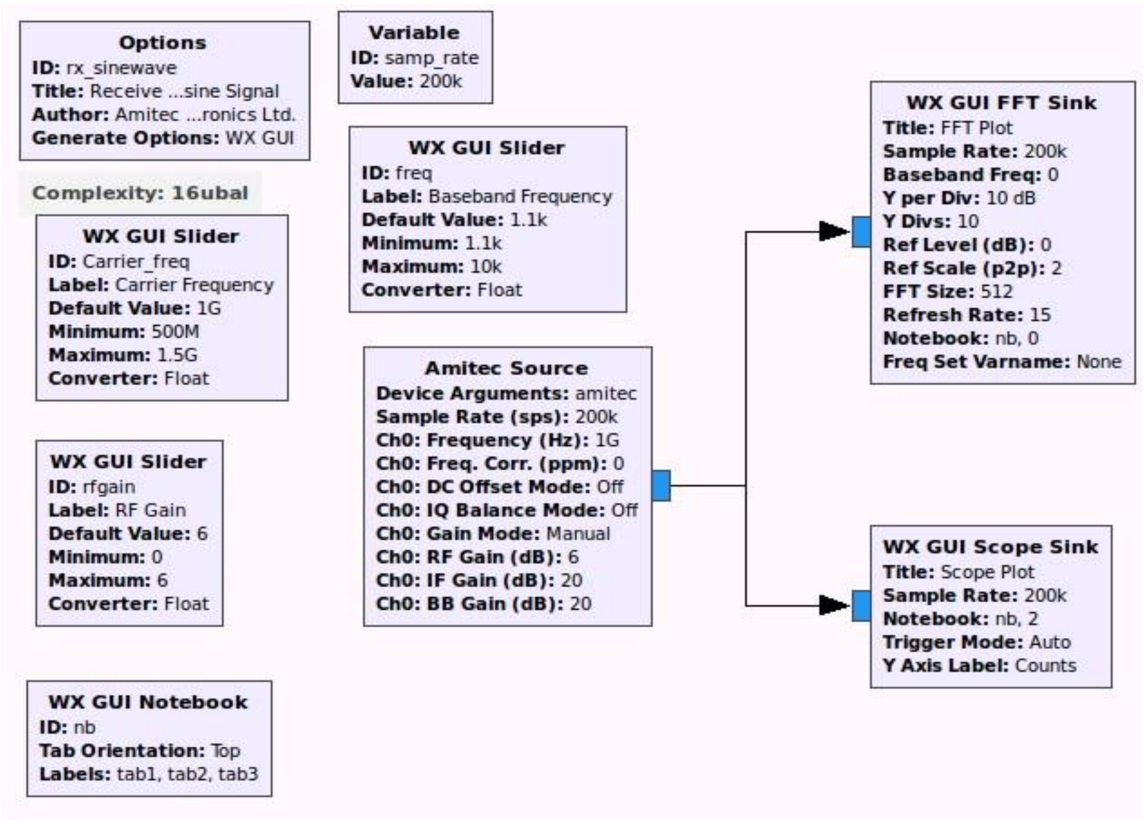
The WX GUI frequency sink plots the power spectrum density of the modulated signal . The center frequency is defined at 0Hz as we observe in the output graph given below. Number of points options are set to 1024 point fft.

Amitec sink is the interfacing between the software simulation and the hardware software defined radio. It enables the output of the modulated signal from the software be transmitted to the SDR and then through the Dipole antenna in order for the signal to transmit to the receiving end. Amitec sink works under 0.4 GHz to 4GHz, hence we hace taken the working frequency of 2GHz. Further radio frequency gains, infrared frequency gains and baseband gains are required as the signal strength goes down when transmitted. So these gains helps in retriving the signal to the best possible extent.



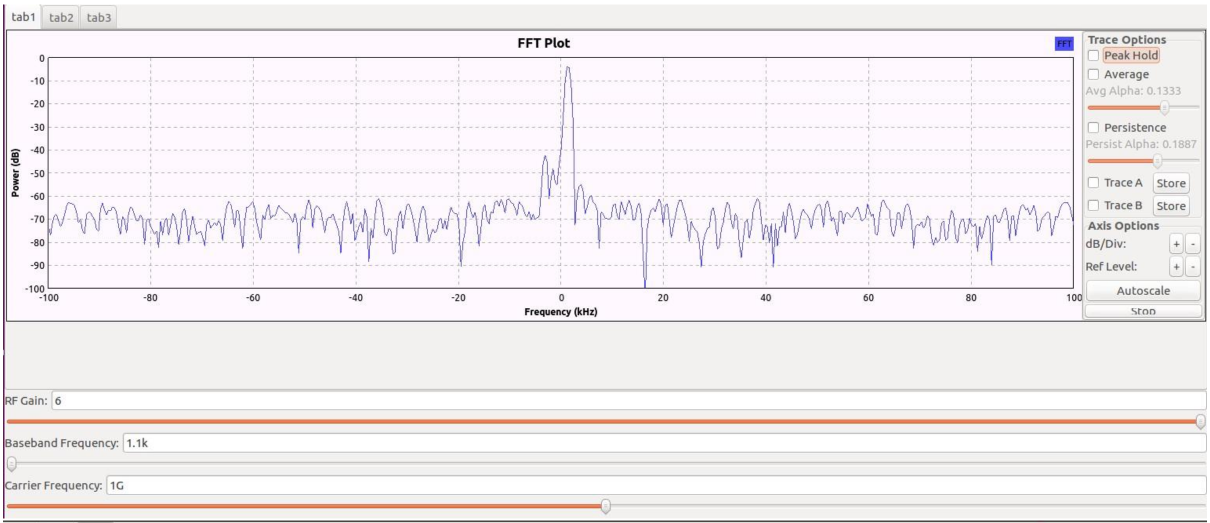
The scope plot gives the sine wave which is transmitted. The blue plot gives the original message sine signal and the green plot gives the wave which is transmitted through the amitec source. The green plot is slightly shifted due to unstability in the hardware radio sink which we are using here. Here time period is 0.9ms, so the frequency is 1/0.9, which is approximately 1.11 KHz. So this value is practically checked with the given frequency.

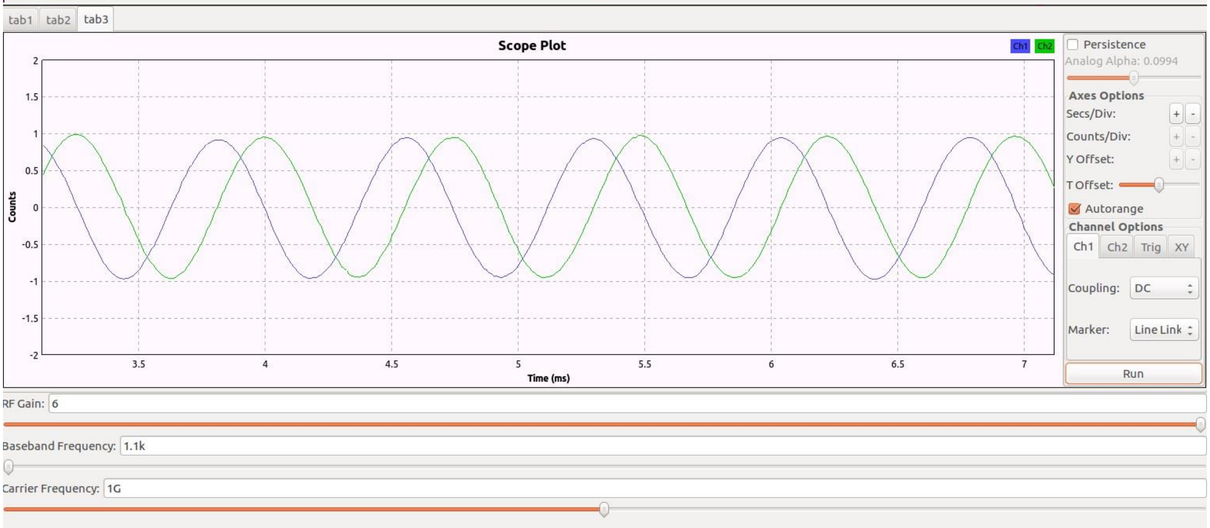
The second plot represents the frequency spectrum of the transmitted signal. Clearly the peak is obtained approximately at 1KHz which is tallied with the original used frequency.



In the receiver part, the amitec source is present, which should have the same specifications of that of the amitec sink, so that proper reception can happen from the transmitter part. The operating frequency can we anywhere between 0.6GHZ to 4Ghz, and hence chose 1GHz (due to range of operating frequency).

This received signal is then fed to frequency sink and scope sink. The frequency sink and scope sink should operate in the same sampling rate as that of amitec source, which is 200KHz.



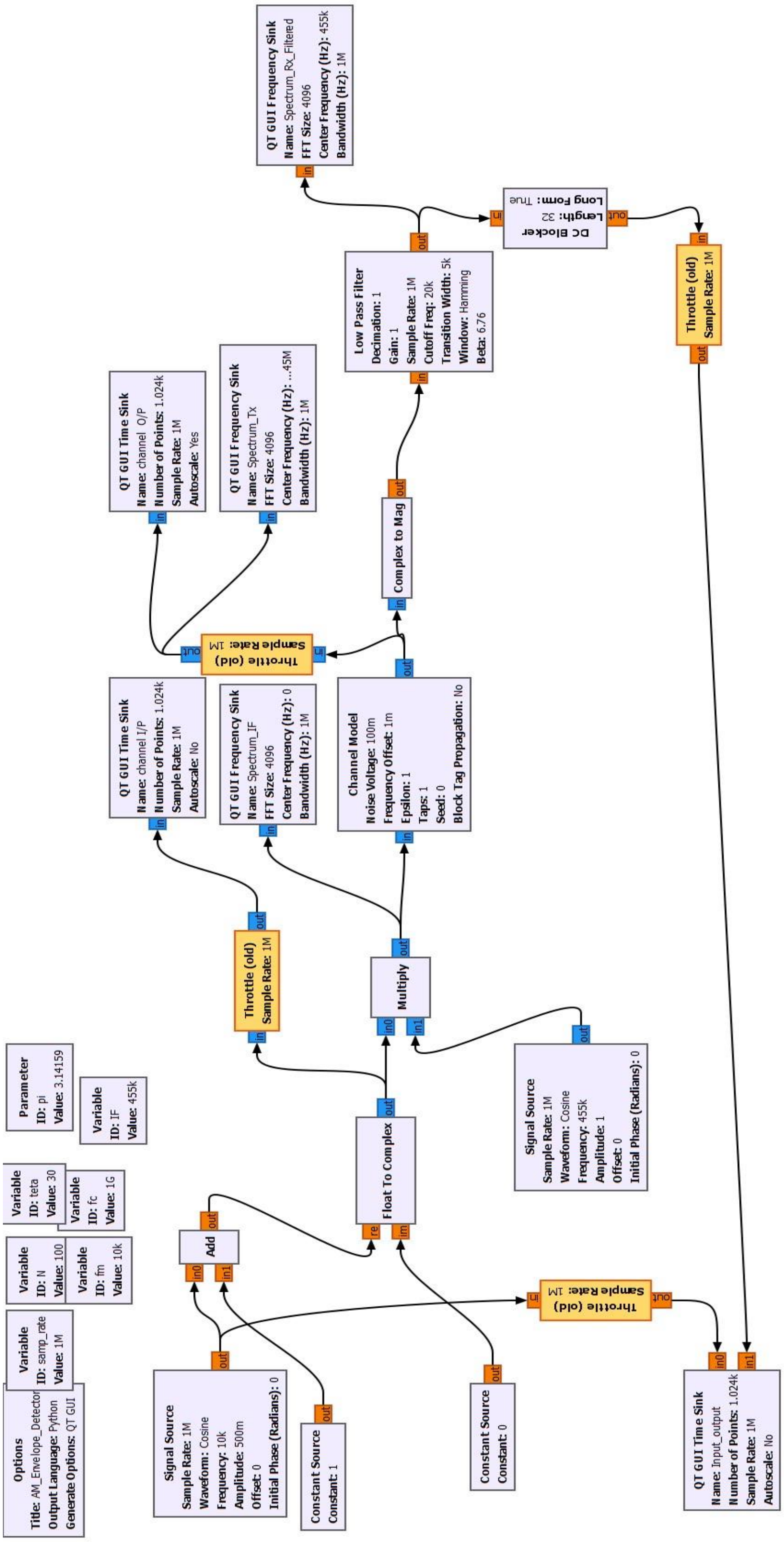


The frequency sink gives the spectrum of the received signal. Theoretically the transmitted and the received signal should have the same spectrum. Practically here we observe a peak at 1.1khz, and hence practically tallied with the transmitter.

The scope plot displays the output of received signal and the output of amitec source. The blue plot shows the time domain output and the green plot shows the amitec output, which is a shifted form of the original due to instability of the hardware radio.

WEEK 2

A) AMPLITUDE MODULATION AND DEMODULATION USING CHANNEL MODEL:



The above image is the channel modulated block diagram in GNU Radio. This is the framework from which we will be deriving the real time implementation of AM (Amplitude Modulated) signals. In the transmitter side of the AM channel model we have the signal source which will generating the message signal in form of a sine wave.

$$m(t) = A_m \cos(2\pi f_m t)$$

We have another carrier signal which will be used to modulate the message signal. Carrier signal will be having a higher frequency than the message signal.

$$c(t) = A_c \cos(2\pi f_c t)$$

The resultant AM modulated wave can be represented by:

$$s(t) = \left[A_c + A_m \cos\left(2\pi f_m t\right)\right]\cos\left(2\pi f_c t\right)$$

OR

$$s(t) = A_c\left[1 + \mu\cos(2\pi f_m t)\right]\cos(2\pi f_c t)$$

Similarly in the receiver side in the back end of the channel model, the already existing AM signal will be demodulated using square law demodulator by multipliying another shifted cosine signal and a low pass filter. Details of the receiver side will be given elaborately in the real time implementation in the receiver side.

Let the AM wave be

$$V_1(t) = A_c\left[1 + k_a m(t)\right]\cos(2\pi f_c t)$$

We know that the mathematical relationship between the input and the output of square law device is:

$$V_2(t) = k_1 V_1(t) + k_2 V_1^2(t)$$

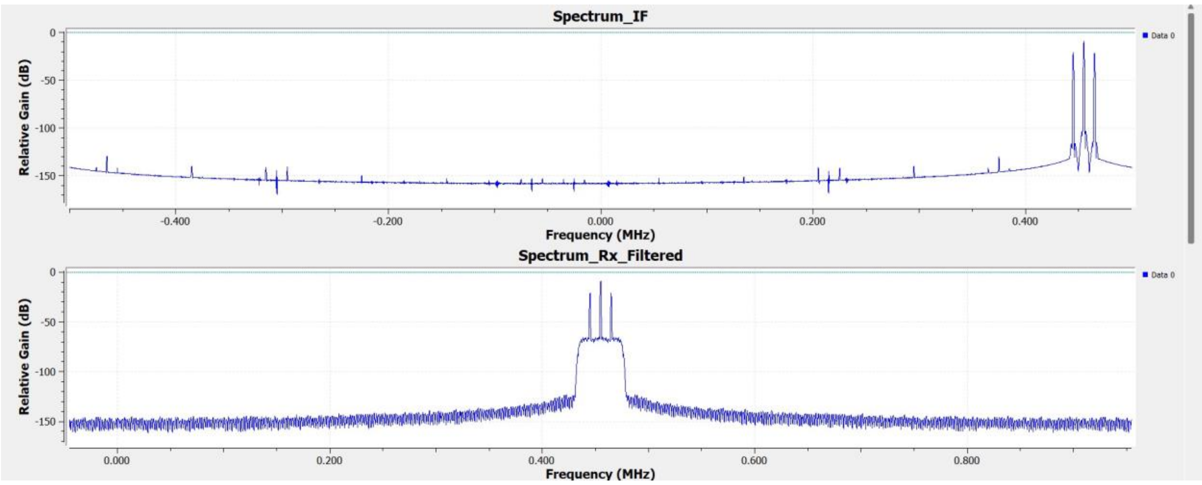
Where, V_1(t) is the input of the square law device, which is nothing but the AM wave
V_2(t) is the output of the square law device ;k_1 and k_2 are constants
Substituting V_1(t) in V_2(t) ,

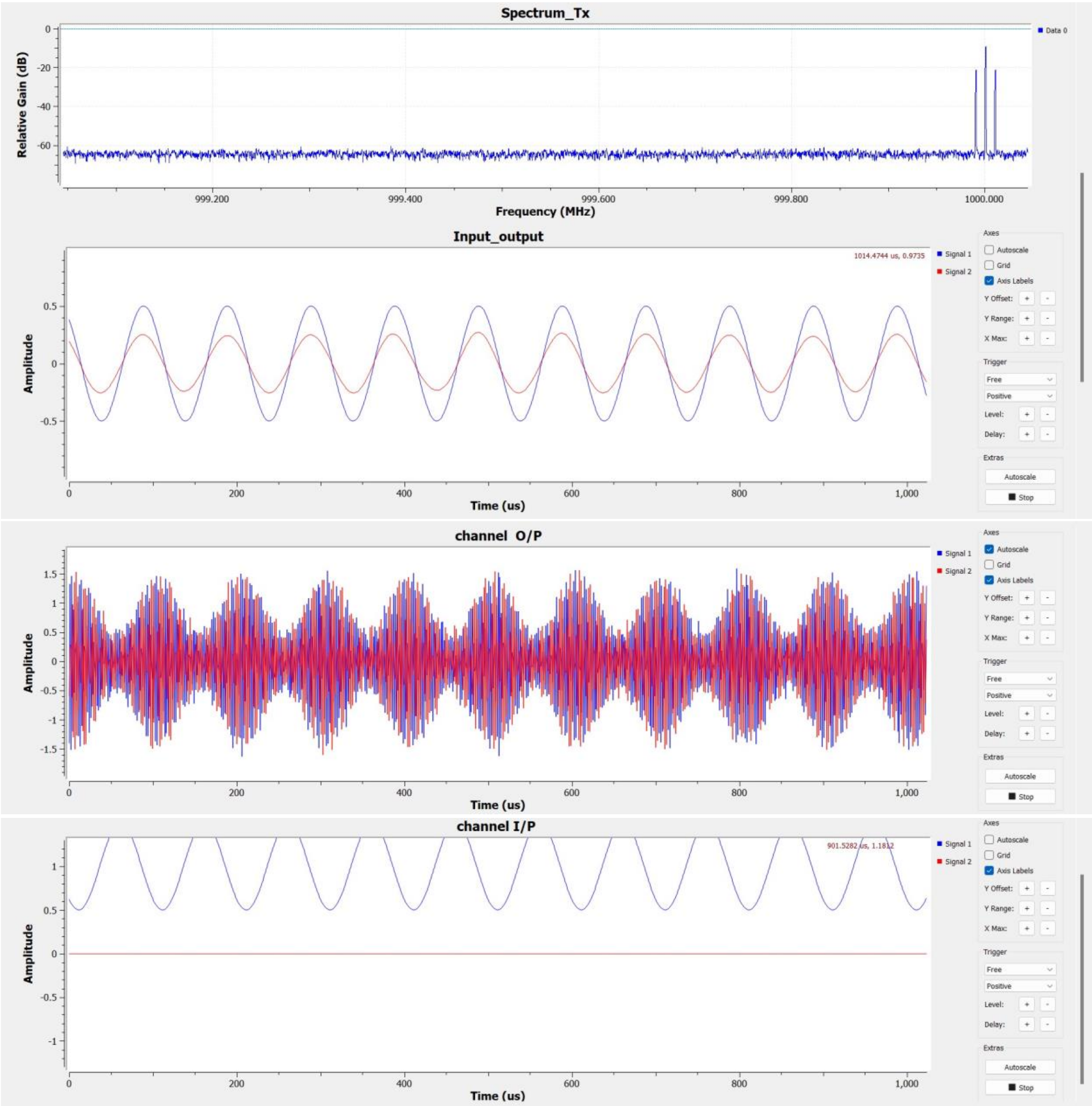
$$V_2(t) = k_1 A_c \cos(2\pi f_c t) + k_1 A_c k_a m(t)\cos(2\pi f_c t) + \frac{K_2 A_c{}^2}{2} + \frac{K_2 A_c{}^2}{2}\cos(4\pi f_c t) + \frac{k_2 A_c{}^2 k_a{}^2 m^2(t)}{2} + \frac{k_2 A_c{}^2 k_a{}^2 m^2(t)}{2}\cos(4\pi f_c t) + k_2 A_c{}^2 k_a m(t) + k_2 A_c{}^2 k_a m(t)\cos(4\pi f_c t)$$

Now passing through the low pass filter we get

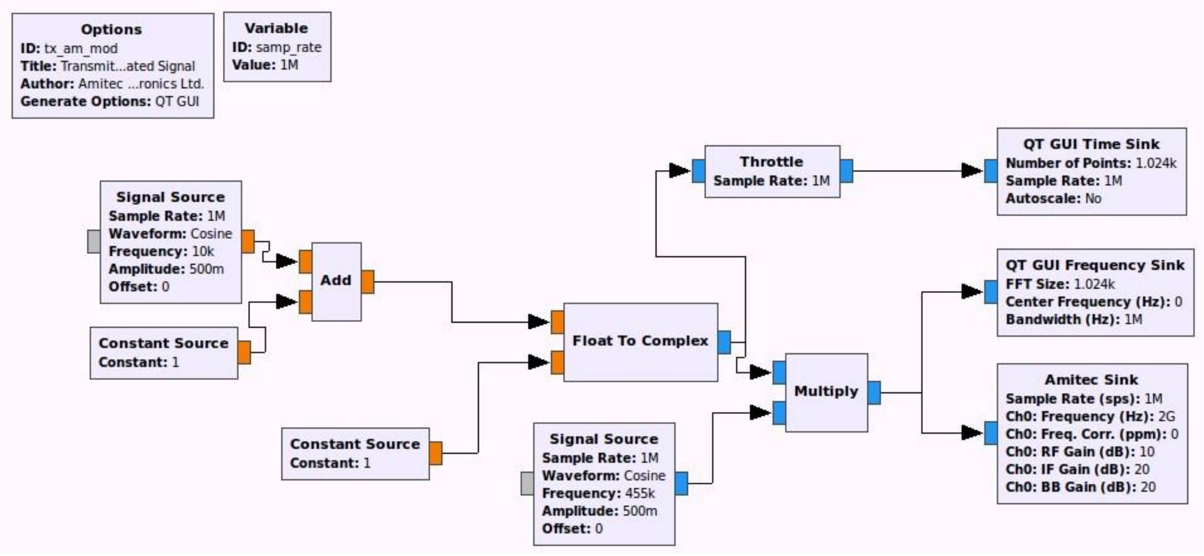
$$V_2(t) = k_2 A_c{}^2 k_a m(t)$$

Hence we get the original cosine wave form after demodulation. In the real time analysis, we will be discussing the working of each and every block.


Spectrum_IF
Relative Gain (dB)
Frequency (MHz)
Data 0
Spectrum_Rx_Filtered
Relative Gain (dB)
Frequency (MHz)
Data 0


Spectrum_Tx
Relative Gain (dB)
Frequency (MHz)
Data 0
Input_output
Amplitude
Time (us)
Signal 1
Signal 2
1014.4744 us, 0.9735
channel O/P
Amplitude
Time (us)
Signal 1
Signal 2
channel I/P
Amplitude
Time (us)
Signal 1
Signal 2
901.5282 us, 1.1812
Axes
Autoscale
Grid
Axis Labels
Y Offset:
Y Range:
X Max:
Trigger
Free
Positive
Level:
Delay:
Extras
Autoscale
Stop

B) REAL TIME AMPLITUDE MODULATED SIGNAL ANALYSIS FROM THE TRANSMITTER:



Here we use signal source with 1Mhz sampling rate which produces a cosine signal at a frequency of 10Khz and amplitude 0.5 volts. We can use either sine of cosine wave, but as in the previously mentioned AM equation as we have used cosine, we take a cosine signal. Signal source main objective is to produce the message signal which will be further modulated.

The constant source block here is used as the general AM theoretical equation uses [1+um(t)]*c(t) equation, so the constant source is added to theoretically match the AM equation.

Next the message signal is converted from floating point identity to a complex identity as the coming carrier signal is a complex identity. Float cannot be multiplied with the complex part and hence both the multiplicants should be in the same domain.

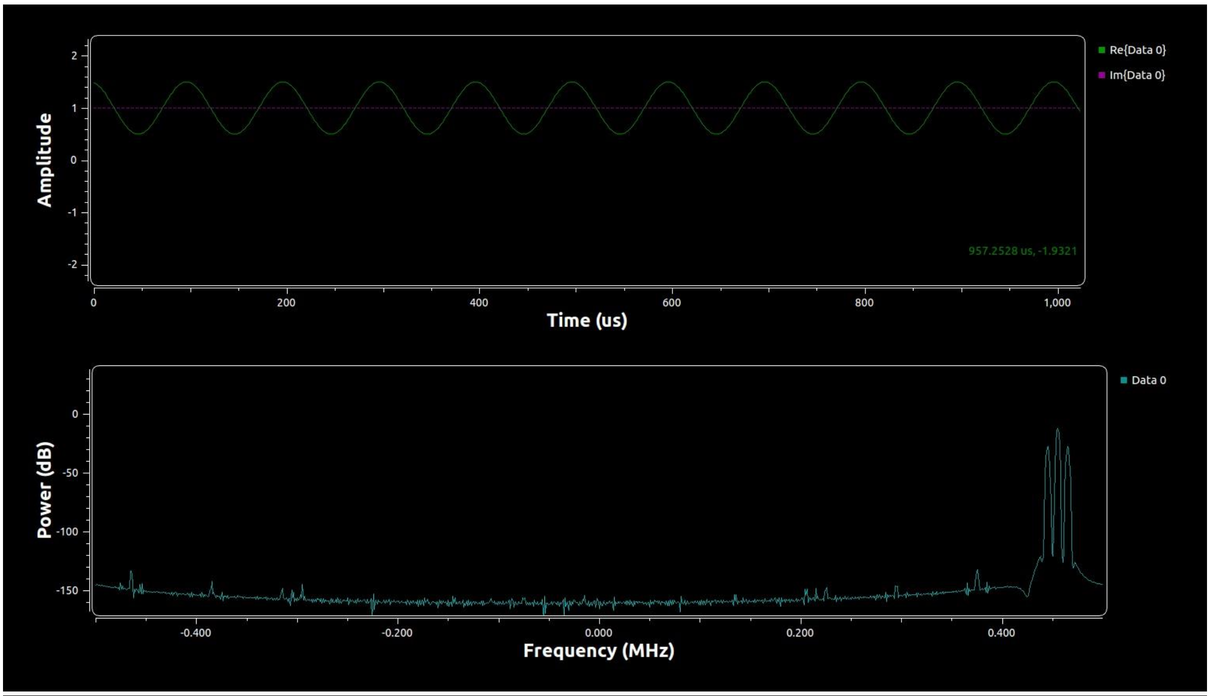
Now another carrier signal source with a cosine wave form with a higher frequency of 455Khz is used to modulate the message signal. This signal is multiplied with the previously generated complex signal in order to get the modulated signal.

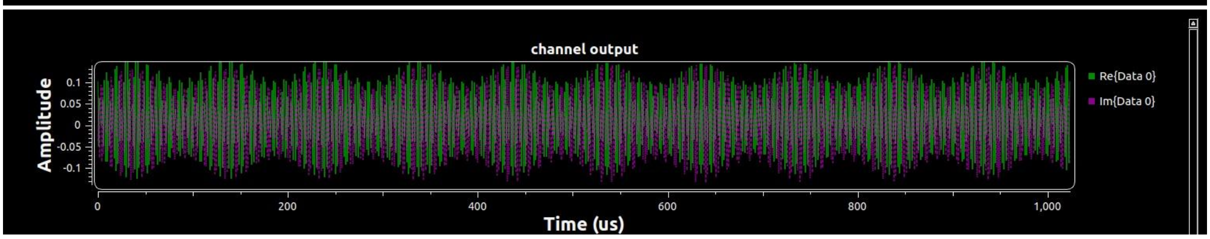
We use the throttle block to accelerate the complex signal in order to display the modulated signal. Throttle block is used in order to limit the rate at which that source block creates samples. We use 1Mhz sampling ratw for the throttle block. So basically the unthrottled signal will have much faster transition rate compared to throttled output.

The gui time sink takes the throttled output of the signal, takes a set of complex streams and plots them in time domain. Here we have not provided the autoscale option so the graph resets only if we re run the program. The number of points taken here is 1024 point. It gives an rough estimate of the spacing and clarity of the graph. More the point, more is the clarity over the graph. As the throttle rate is 1Mhz, we take the sampling rate of time sink similar to the throttle rate.

The QT GUI frequency sink plots the power spectrum density of the modulated signal . The center frequency is defined at 0Hz as we observe in the output graph given below. Number of points and autoscale optiins are set according the QT GUI time sink.

Amitec sink is the interfacing between the software simulation and the hardware software defined radio. It enables the output of the modulated signal from the software be transmitted to the SDR and then through the Dipole antenna in order for the signal to transmit to the receiving end. Amitec sink works under 0.4 GHz to 4GHz, hence we hace taken the working frequency of 2GHz. Further radio frequency gains, infrared frequency gains and baseband gains are required as the signal strength goes down when transmitted. So these gains helps in retriving the signal to the best possible extent.



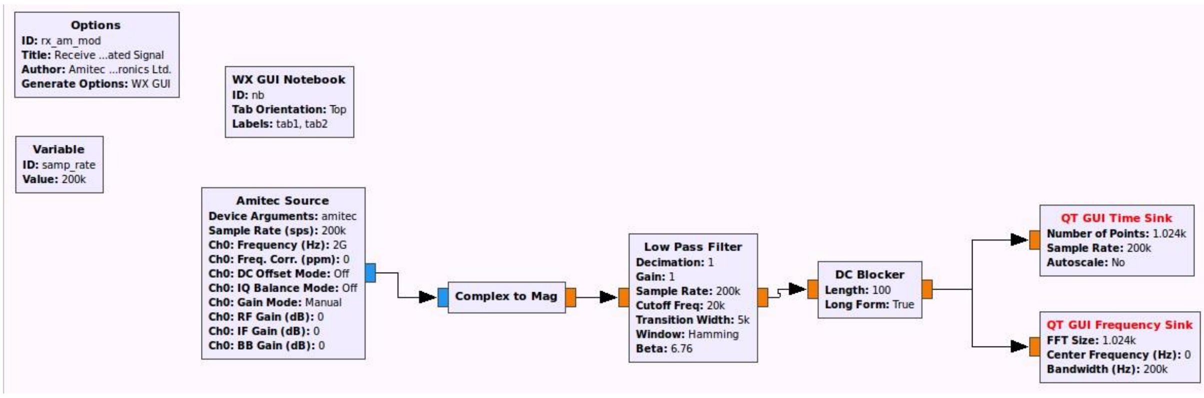


Here in the 1st output we have a sinusoidal output. This is because we have directly connected the throttled cosine signal to the time sink. In addition we also get a another line which indicates the constant that we have add which is 1 in this case.

In the 2nd output we have plotted the power spectrum density of the modulated signal. As this is AM signal, we should theoretically be having a full peak carrier with both sidebands bands in their original equal peak. This theory can be verified here as the real time output of the transmitter side correctly matches with the theory. Also the sideband must be present at fc+fm and fc-fm, which is at 445KHZ and 465KHz. This also can be practically seen here. The carrier peak is at 445KHz.

The 3rd output gives a AM modulated wave in QT GUI time sink. Here as modulation index is 2, we get this type of wave.

C) REAL TIME AMPLITUDE DEMODULATED SIGNAL ANALYSIS FROM THE RECIEVER:



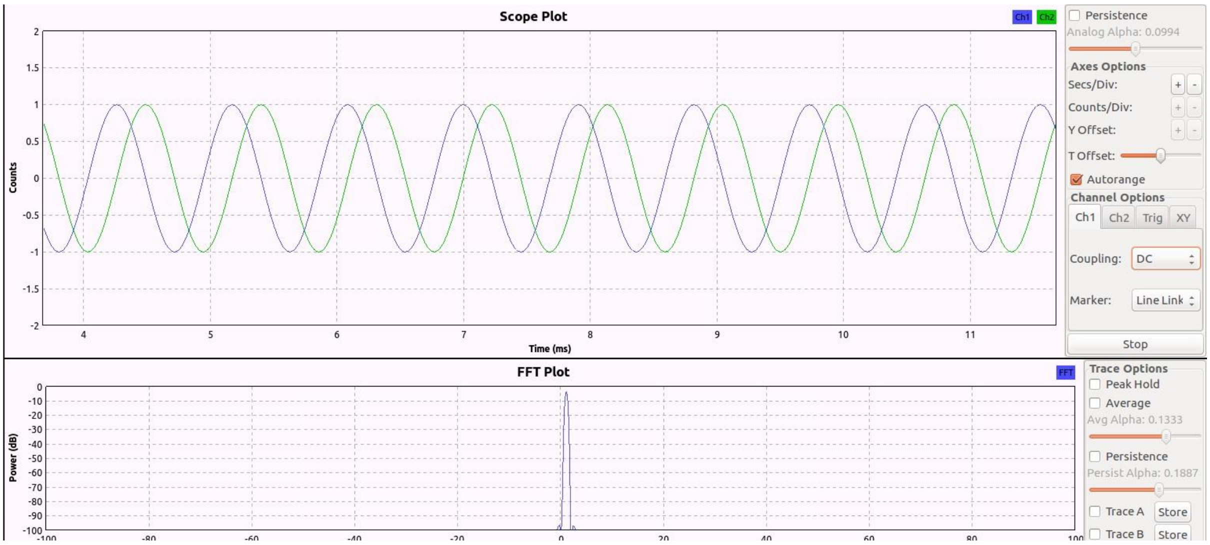
In the receiver side, we demodulate the signal to get back the original message signal. Now from the amitec sink and dipole antenna, the receiving dipole antenna collects the signal and sends it back to the receiving software part using amitec source. The amitec sink and source must work in sync, and hence they should have the same specifications.

Here to demodulate we use the square law demodulation method. So we first convert the AM modulated signal from complex illustration to a floating point illustration as floating point analysis is easy. Now pass it through the low pass filter as mention in demodulation theory. As we use simple LPF filter, the decimation and interpolation factor is taken as 1. Sample rate is also the same as that of throttle block.

Also here as FIR Filter is used, we define a window with one of the best compensations. So we use hamming window (blackmann windows can also be used). The beta parameter is only used for kaiser window, but the default value for beta is 6.76. The cutoff frequency denotes the frequency over which the signal components will not be transferred further or will be cut off. Transition width describes about the difference between the passband and stopband frequencies. Here we have taken TW as 5KHz.

The above filtered signal is further applied to the DC blocker which blocks the dc components of the message signal which was passed through the low pass filter unblocked. The output of this gives the original message signal with a multiplied factor, which will only disturb the peak of the signal.

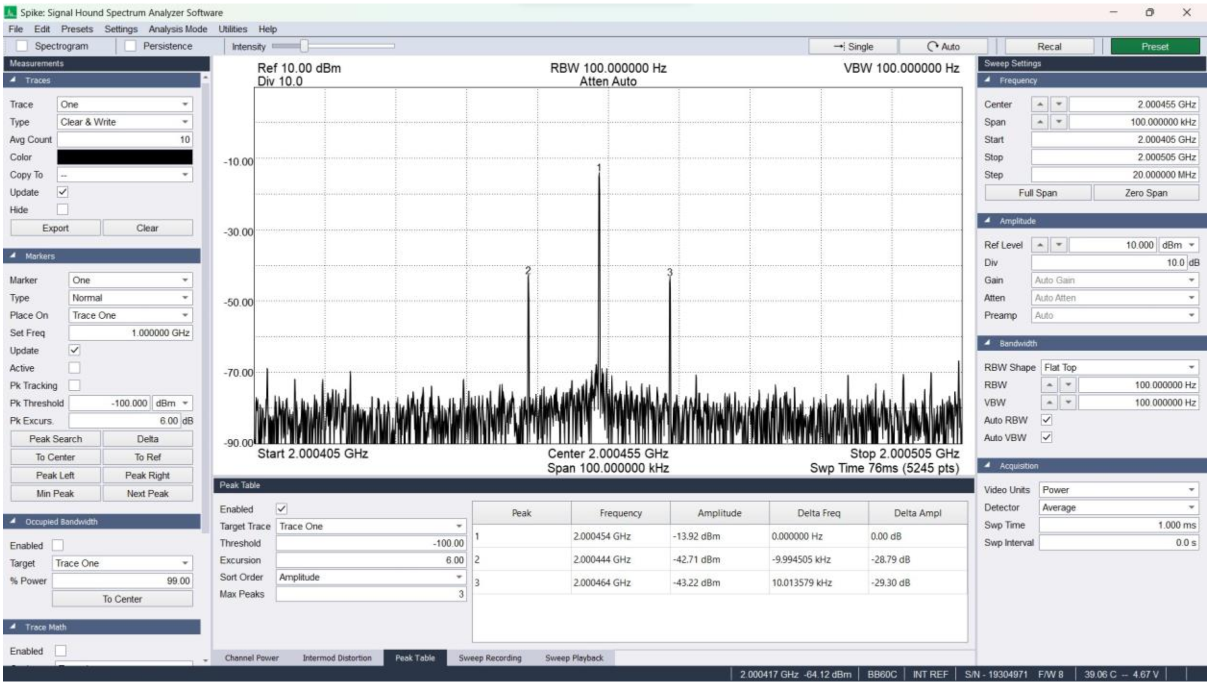
This produced signal is fed to QT GUI time and frequency sink with no of points 1024 which depicts the spacing and clarity of the output. Center frequency of the sink is given at 0khz and sample rate is 200Khz.

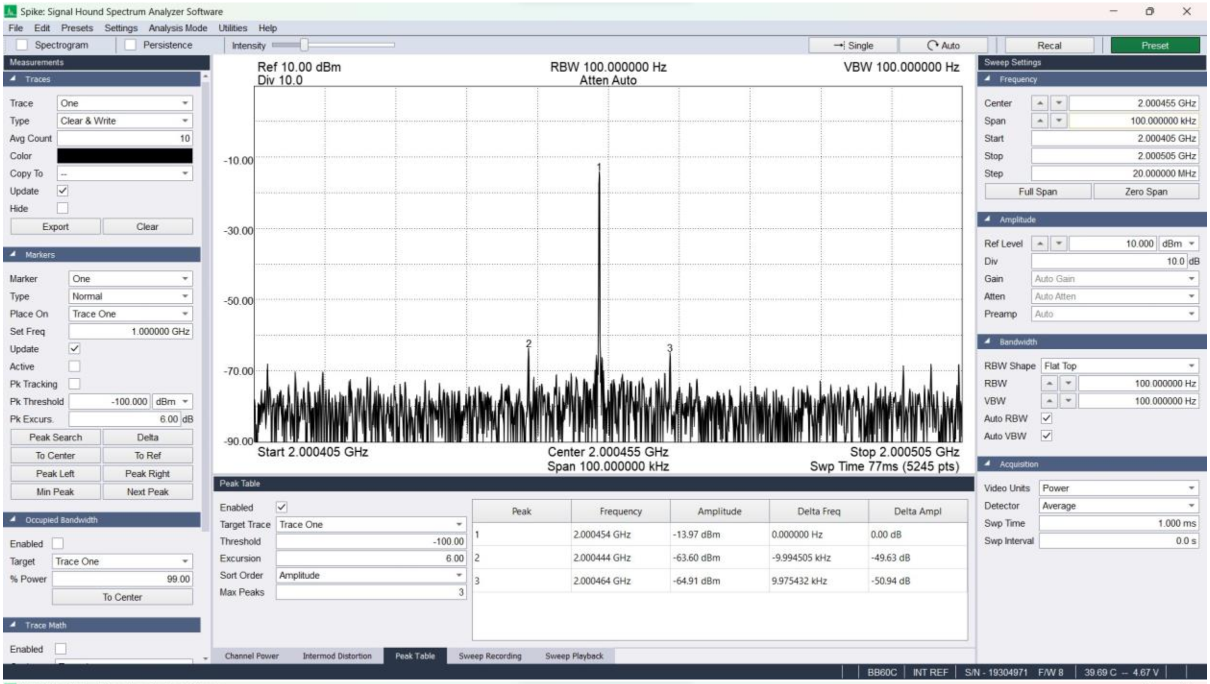


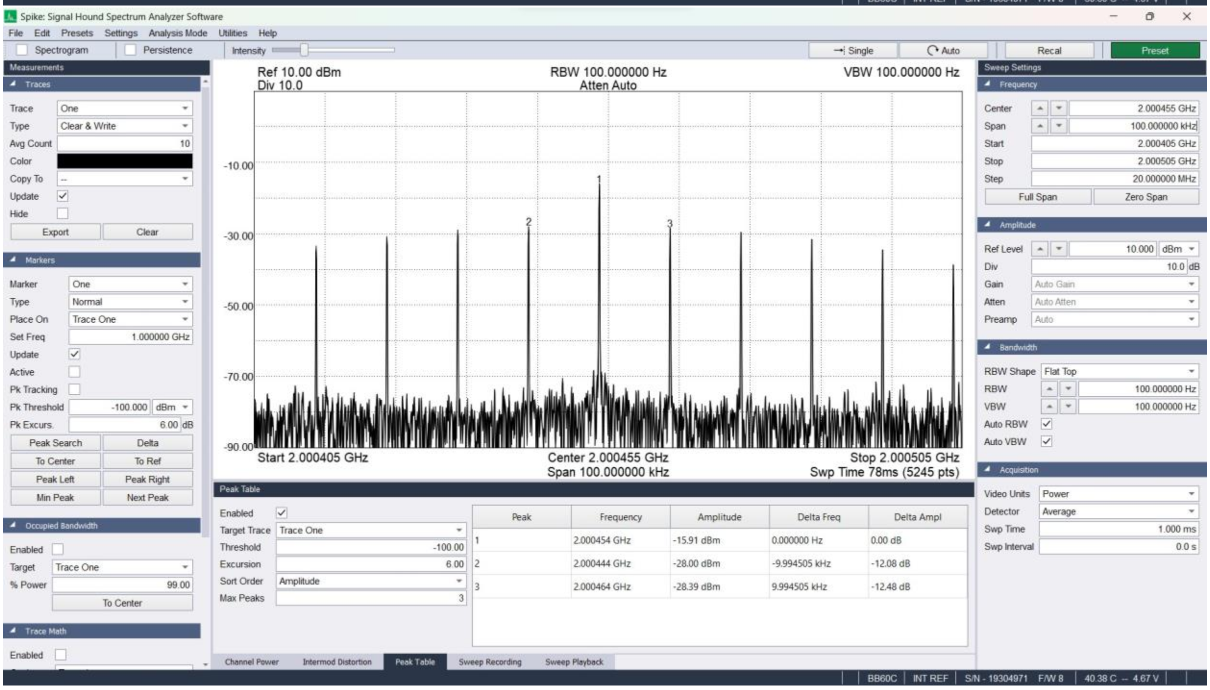
The 1st output is the channel output of is QT GUI time sink. It shows us the time demodulated AM signal (which is the message signal with a diminished amplitude due to amplitude factor).

The 2nd output is the power spectral density of the demodulated signal which is collected at the amitec source to a QT GUI frequency sink. This output is just a representation of message signal having its peak at 10khz, which matches with the existing theory.

D) <u>REAL TIME AMPLITUDE MODULATED SIGNAL ANALYSIS USING SPECTRUM HOUND:</u>



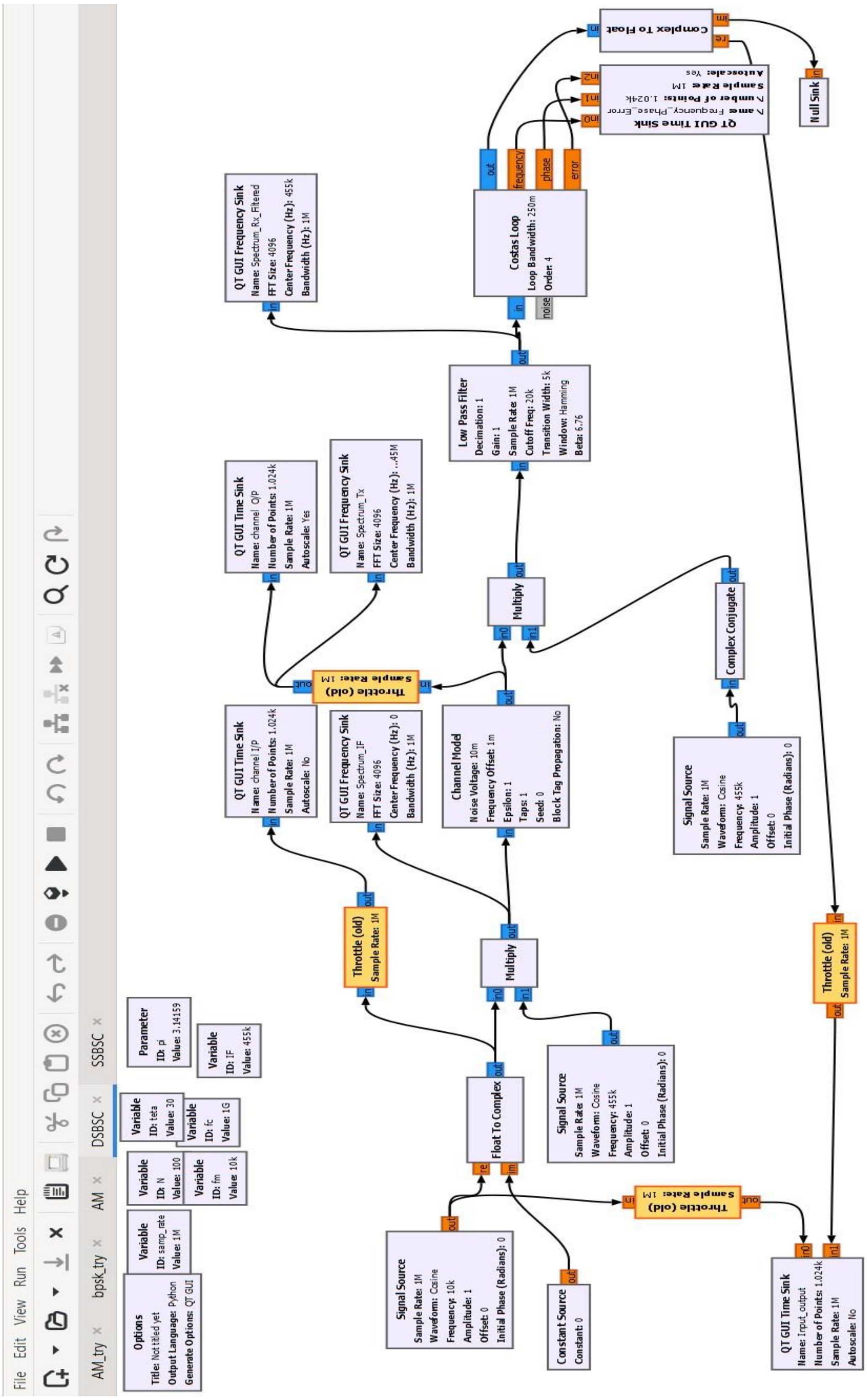




Here the 1st 2nd and 3rd spike plot represents the spectrum of AM modulated signal at 0.1, 0.01 and 1 Volts respectively. The difference between the 1st and 2nd plot is the attenuation of the sidebands. The 0.1v yields 29db attenuation and the 0.01v yields 49db attenuation. The 1v plot gives rise to harmonics which can be seen in the image.

WEEK 3

A) DSBSC MODULATION AND DEMODULATION USING CHANNEL MODEL:



The above image is the channel modulated block diagram in GNU Radio. This is the framework from which we will be deriving the real time implementation of DSBSC (Double side band supressed carrier) signals.

In the transmitter side of the DSBSC channel model we have the signal source which will generating the message signal in form of a sine wave.

$$m\left(t\right)=A_m\cos(2\pi f_m t)$$

We have another carrier signal which will be used to modulate the message signal. Carrier signal will be having a higher frequency than the message signal.

$$c\left(t\right)=A_c\cos(2\pi f_c t)$$

The resultant DSBSC modulated wave can be represented by:

$$s\left(t\right)=m\left(t\right)c\left(t\right)$$

$$s\left(t\right)=A_m A_c\cos(2\pi f_m t)\cos(2\pi f_c t)$$

Similarly in the receiver side in the back end of the channel model, the already existing DSBSC signal will be demodulated using coherent detector by multipliying another shifted cosine signal and a low pass filter. Details of the receiver side will be given elaborately in the real time implementation in the receiver side.

Let the DSBSC wave be

$$s\left(t\right)=A_c\cos(2\pi f_c t)m\left(t\right)$$

The output of the local oscillator is

$$c\left(t\right)=A_c\cos(2\pi f_c t+\phi)$$

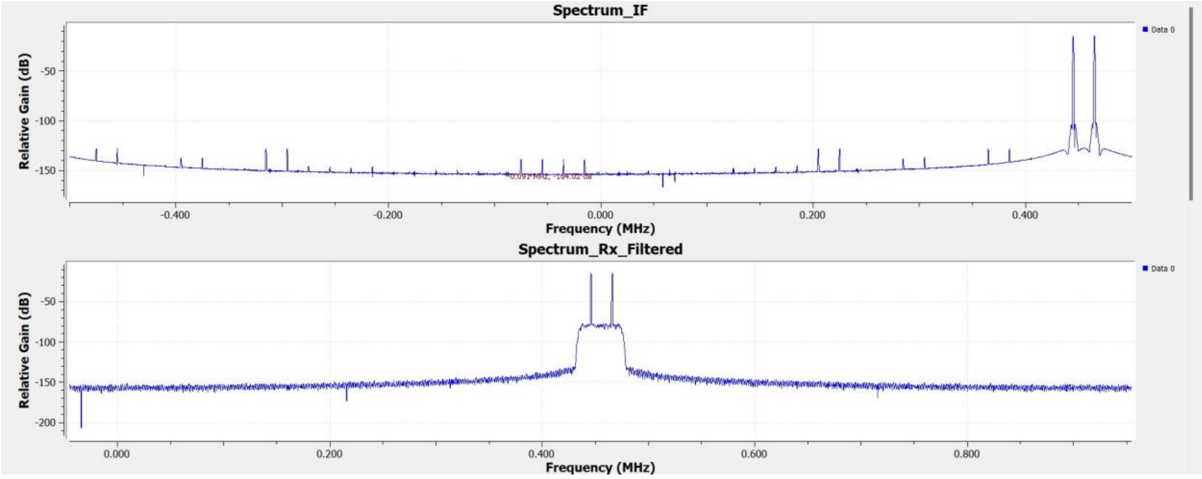
Where, ϕ is the phase difference between the local oscillator signal and the carrier signal, which is used for DSBSC modulation.

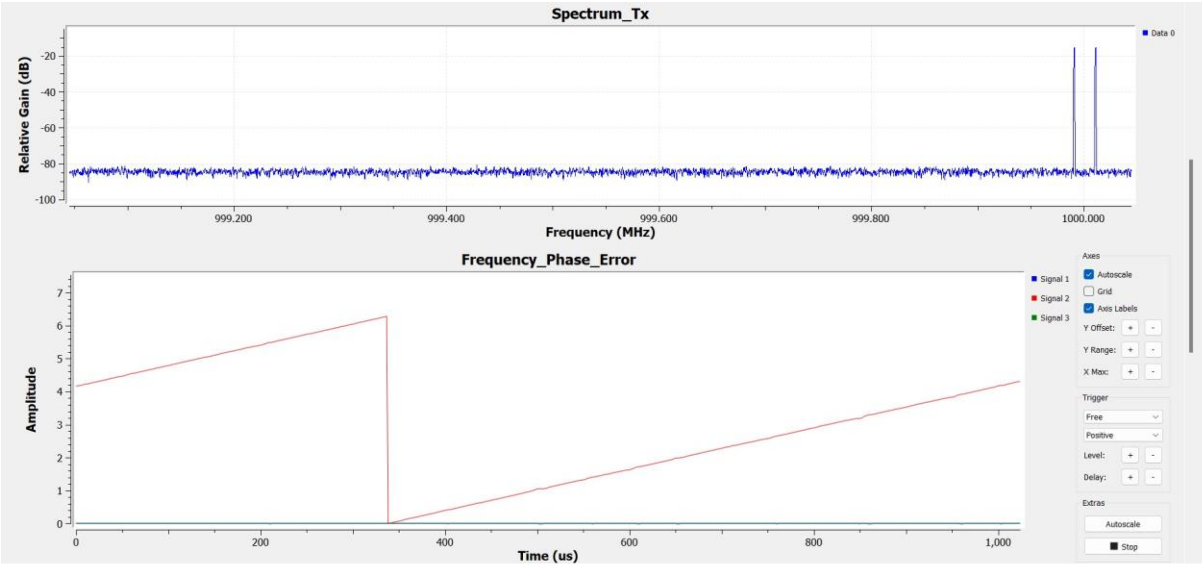
$$v\left(t\right)=\frac{{A_c}^2}{2}\cos\phi m\left(t\right)+\frac{{A_c}^2}{2}\cos(4\pi f_c t+\phi)m\left(t\right)$$

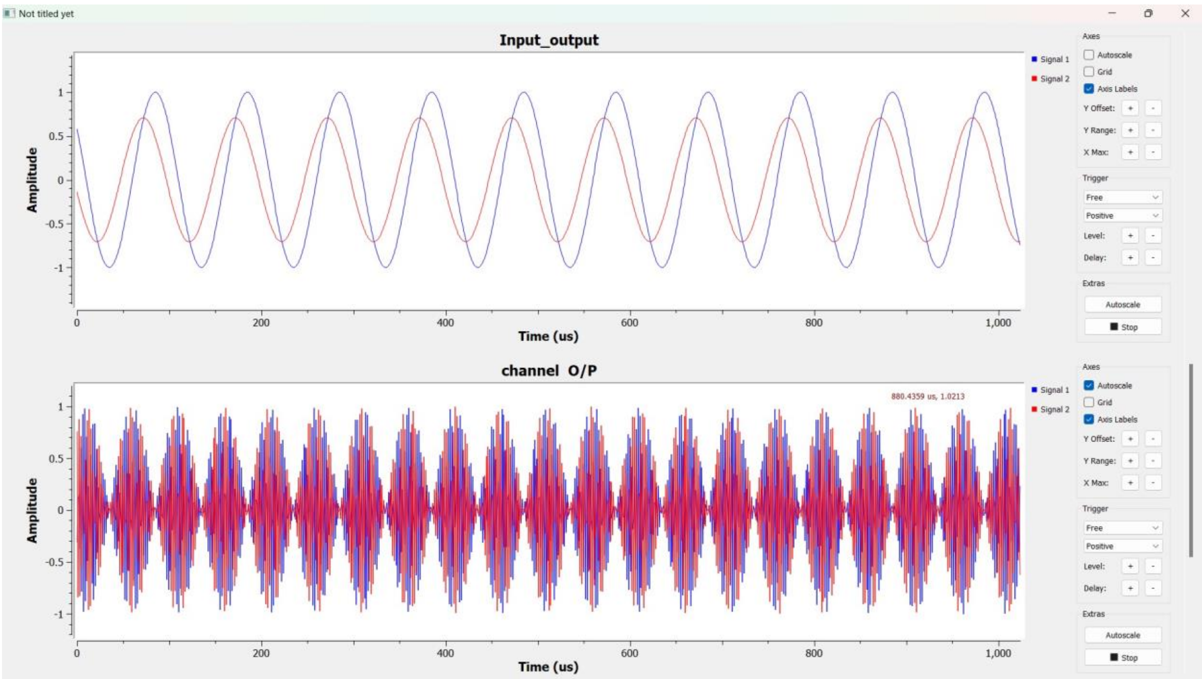
In the above equation, the first term is the scaled version of the message signal. It can be extracted by passing the above signal through a low pass filter. Therefore, the output of low pass filter is

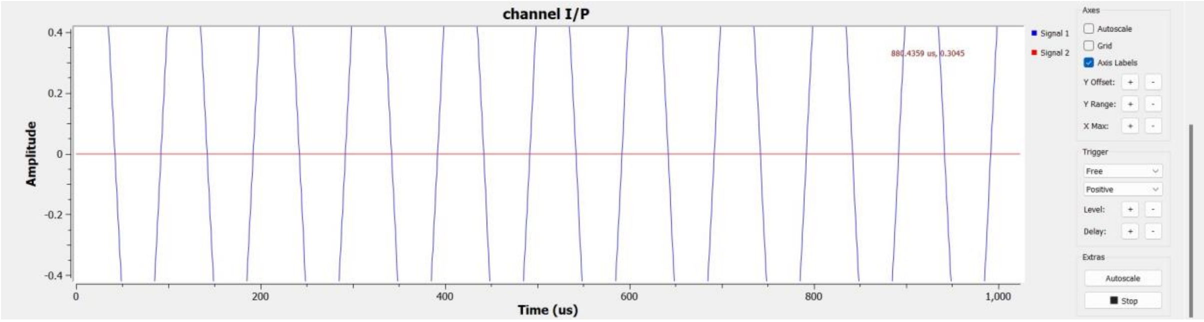
$$v_0 t=\frac{{A_c}^2}{2}\cos\phi m\left(t\right)$$

Hence we get the original cosine wave form after demodulation. In the real time analysis, we will be discussing the working of each and every block.

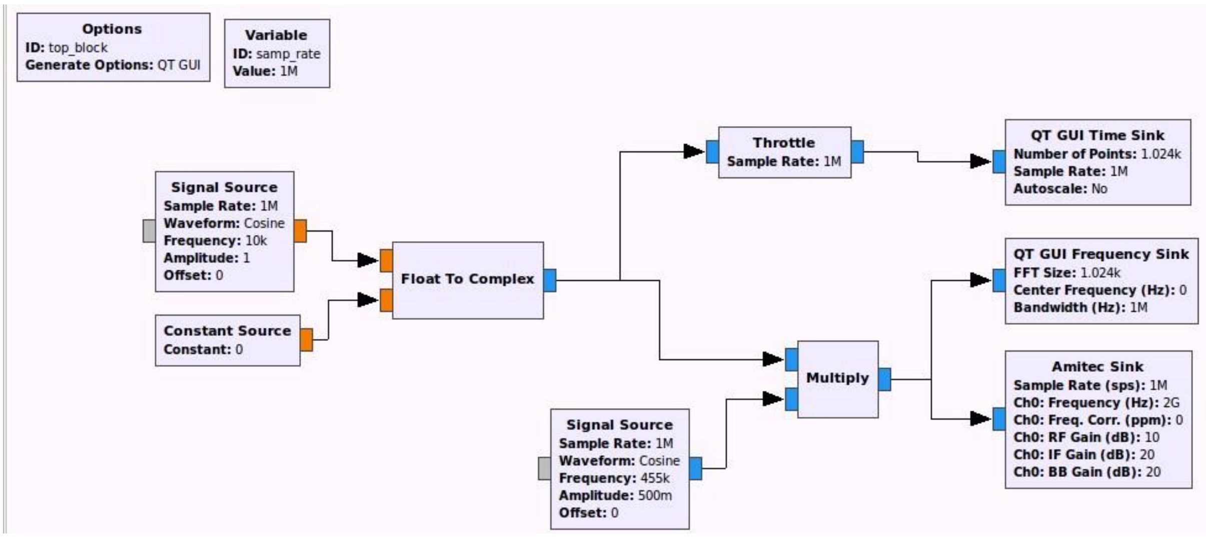

Spectrum_IF
Relative Gain (dB)
Frequency (MHz)
Data 0
Spectrum_Rx_Filtered
Relative Gain (dB)
Frequency (MHz)
Data 0


Spectrum_Tx
Relative Gain (dB)
Frequency (MHz)
Data 0
Frequency_Phase_Error
Amplitude
Time (us)
Signal 1
Signal 2
Signal 3
Axes
Autoscale
Grid
Axis Labels
Y Offset:
Y Range:
X Max:
Trigger
Free
Positive
Level:
Delay:
Extras
Autoscale
Stop


Not titled yet
Input_output
Amplitude
Time (us)
Signal 1
Signal 2
channel O/P
880.4359 us, 1.0213
Amplitude
Time (us)
Signal 1
Signal 2
Axes
Autoscale
Grid
Axis Labels
Y Offset:
Y Range:
X Max:
Trigger
Free
Positive
Level:
Delay:
Extras
Autoscale
Stop



B) REAL TIME DSBSC MODULATED SIGNAL ANALYSIS FROM THE TRANSMITTER:



Here we use signal source with 1Mhz sampling rate which produces a cosine signal at a frequency of 10Khz and amplitude 1 volts. We can useeither sine of cosine wave, but as in the previously mentioned dsbsc equation as we have used cosine, we take a cosine signal. Signal source main objective is to produce the message signal which will be further modulated.

The constant source block here is of no use as the constant added is 0, but is kept as such to have a comparison with the AM counterpart.

Next the message signal is converted from floating point identity to a complex identity as the coming carrier signal is a complex identity. Float cannot be multiplied with thecompax part and hence both the multiplicants should be in the same domain.

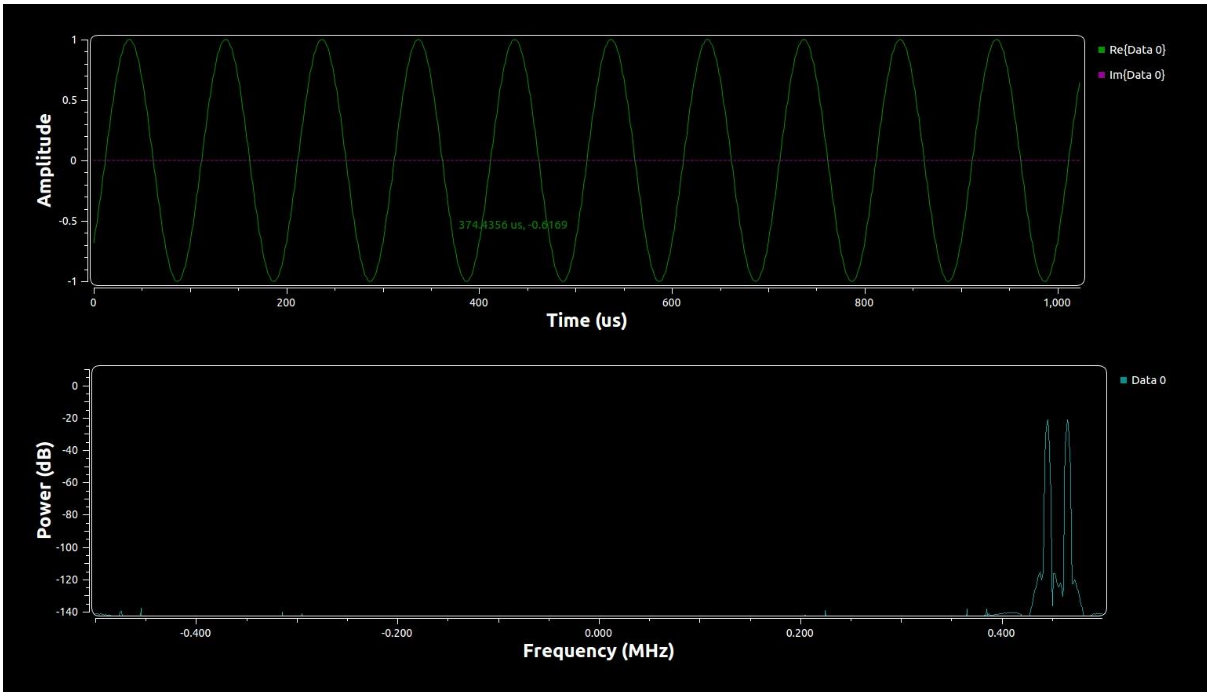
Now another carrier signal source with a cosine wave form with a higher frequency of 455Khz is used to modulate the message signal. This signal is multiplied with the previously generated message signal in order to get the modulated signal.

We use the throttle block to accelerate the complex signal in order to display the modulated signal. Throttle block is used in order to limit the rate at which that source block creates samples. We use 1Mhz sampling ratw for the throttle block. So basically the unthrottled signal will have much faster transition rate compared to throttled output.

The gui time sink takes the throttled output of the signal, takes a set of complex streams and plots them in time domain. Here we have not provided the autoscale option so the graph resets only if we re run the program. The number of points taken here is 1024 point. It gives an rough estimate of the spacing and clarity of the graph. More the point, more is the clarity over the graph. As the throttle rate is 1Mhz, we take the sampling rate of time sink similar to the throttle rate.

The QT GUI frequency sink plots the power spectrum density of the modulated signal . the center frequency is defined at 0Hz as we observe in the output graph given below. Number of points and autoscale optiins are set according the QT GUI time sink.

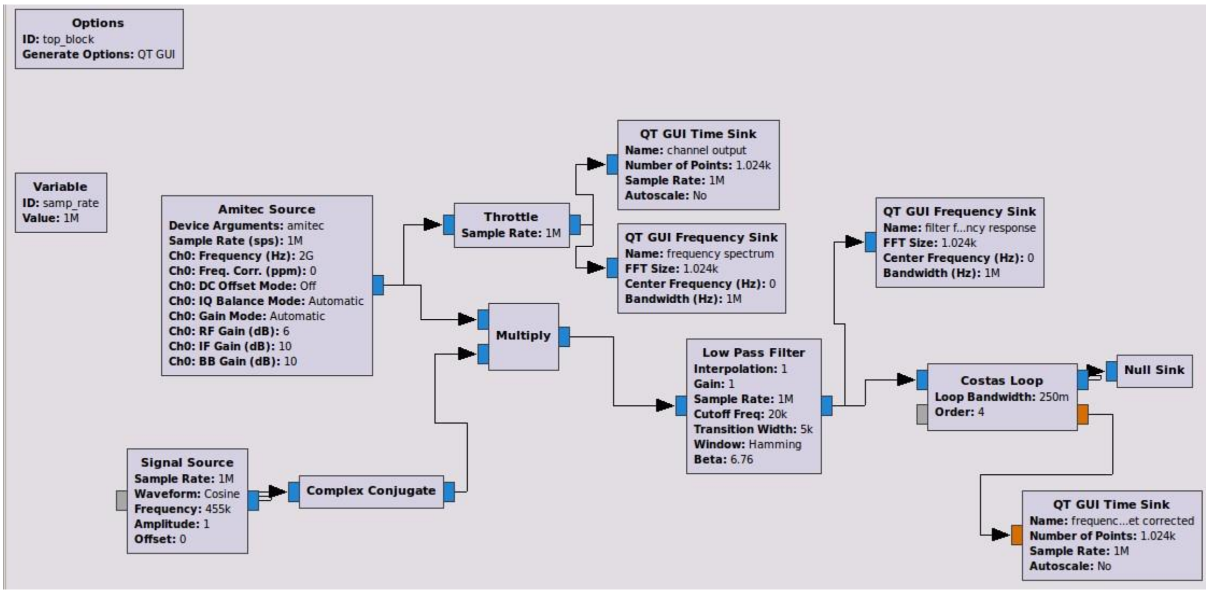
Amitec sink is the interfacing between the software simulation and the hardware software defined radio. It enables the output of the modulated signal from the software be transmitted to the SDR and then through the Dipole antenna in order for the signal to transmit to the receiving end. Amitec sink works under 0.4 GHz to 4GHz, hence we hace taken the working frequency of 2GHz. Further radio frequency gains, infrared frequency gains and baseband gains are required as the signal strength goes down when transmitted. So these gains helps in retriving the signal to the best possible extent.



Here in the 1st output we have a sinusoidal output. This is because we have directly connected the throttled cosine signal to the time sink. In addition we also get a another line which indicates the constant that we have add which is 0 in this case.

In the 2nd output we have plotted the power spectrum density of the modulated signal. As this is dsbsc signal, we should theoretically be having a supressed/no carrier with both sidebands bands in their original equal peak. This theory can be verified here as the real time output of the transmitter side correctly matches with the theory. Also the sideband must be present at fc+fm and fc-fm, which is at 445KHZ and 465KHz. This also can be practically seen here.

C) REAL TIME DSBSC DEMODULATED SIGNAL ANALYSIS FROM THE RECIEVER:



In the receiver side, we demodulate the signal to get back the original message signal. Now from the amitec sink and dipole antenna, the receiving dipole antenna collects the signal and sends it back to the receiving software part using amitec source. The amitec sink and source must work in sync, and hence they should have the same specifications.

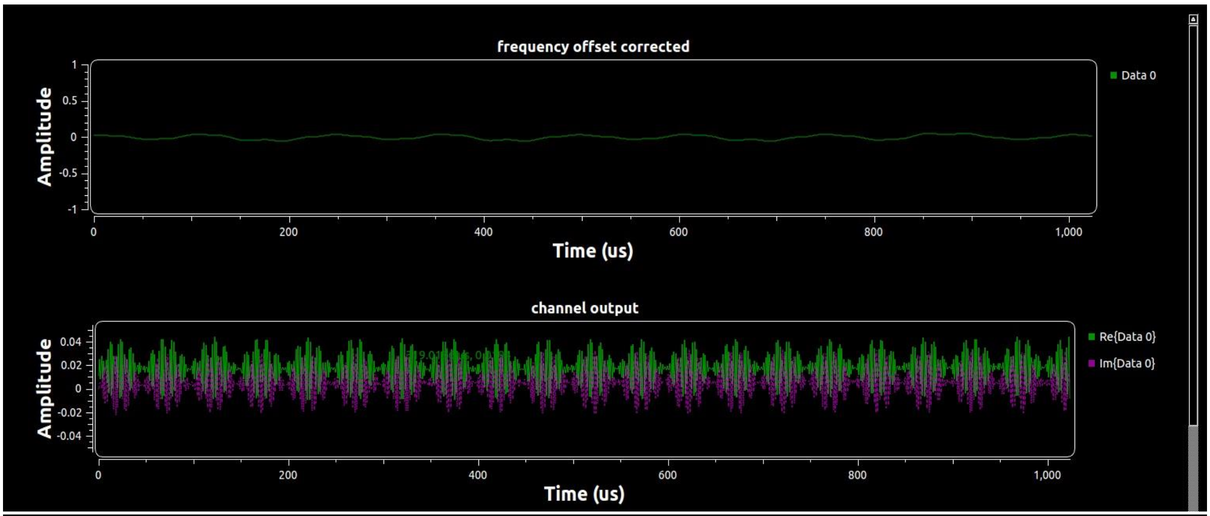
Again we use throttle here as the signal has been transmitted and received, so there must be lot of disturbances in the signal, which is reduced by the throttle. This signal is again fed to QT GUI time and frequency sink.

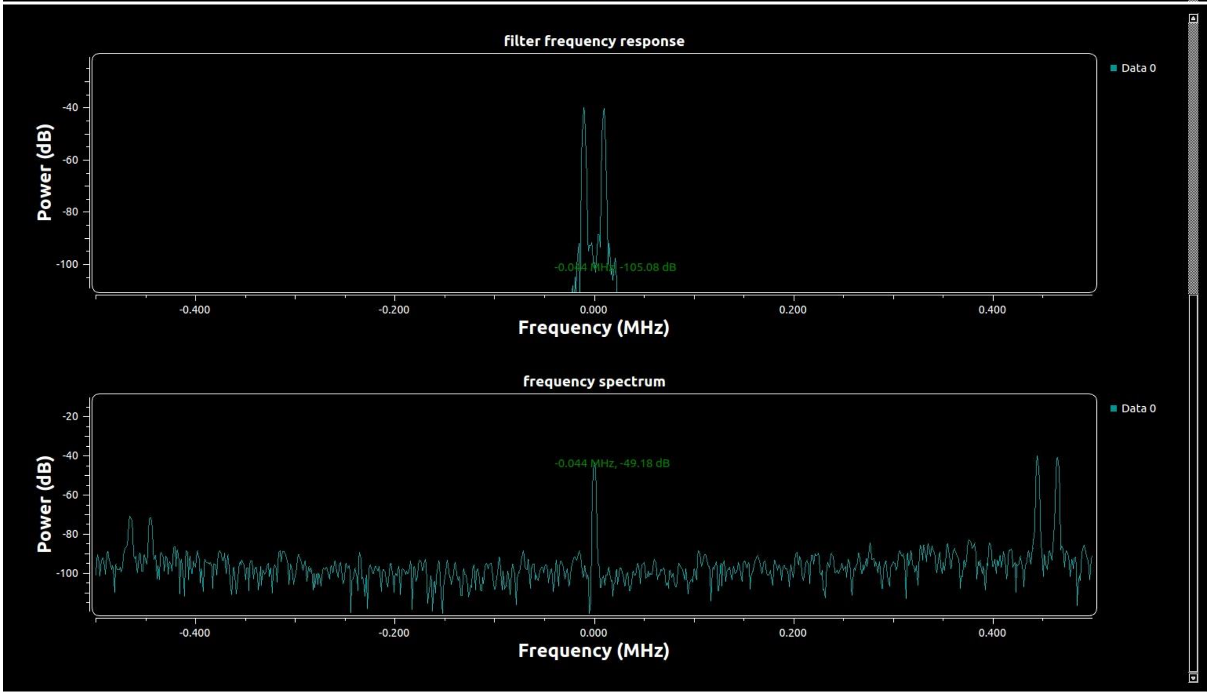
Here to demodulate we use the costas loop demodulation method. So we first multiply the received dsbsc signal with another carrier signal. And pass it through the low pass filter as mention in demodulation theory. As we use simple LPF filter, the decimation and interpolation factor is taken as 1. Sample rate is also the same as that of throttle block.

Also here as FIR Filter is used, we define a window with one of the best compensations. So we use hamming window (blackmann windows can also be used). The beta parameter is only used for kaiser window, but the default value for beta is 6.76. The cutoff frequency denotes the frequency over which the signal components will not be transferred further or will be cut off. Transition width describes about the difference between the passband and stopband frequencies. Here we have taken TW as 5KHz.

The above filtered signal is further applied to the costas loop. Here we get a problem regarding the carrier frequency offset. This offset is removed using the costas loop. Carrier frequency offset occurs when the signal after decimation in the low pass filter do not synchronise with the carrier signal which we have multiplied initially.

This offset is reduced by the costas loop block, which is having a bandwidth of 250mHz and a 4th order loop. The blue output of the costas loop block indicates the frequency output which is given to the null sink, to eliminate frequency offset. The gray box output of the costas loop is the frequency compensated signal which is then displayed in the time sink. This time sink will have the demodulated original message signal which is the sinusoidal signal.





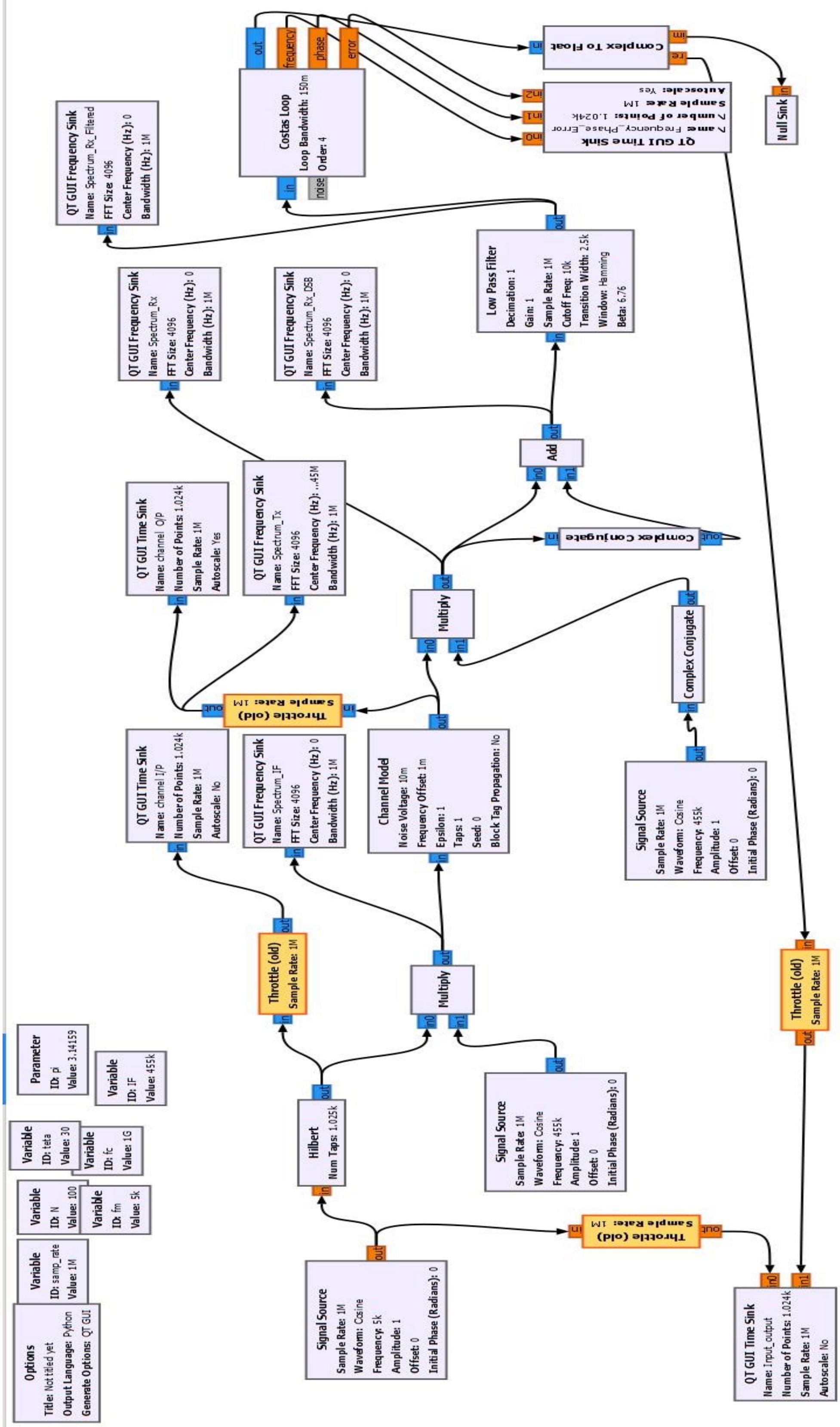
In the output, the 1st output depicts the frequency offset present in the demodulated signal. As the pattern tends to zero/constant it clearly shows that the output of the costas loop has given us an offset free signal.

The 2nd output is the channel output which is directly taken from the amitec source to QT GUI time sink. It shows us the time modulated dsbsc signal. As we have taken into account the compleax part of the signal also, the upper have of the dsbsc signal consists of the real part and the lower half gives the details on the imaginary part.

The 3rd output is the low pass filter output at QT GUI frequency sink. As the cutoff frequency of the low pass filter is 20KHz, we get 2 peaks of the dsbsc signals at +20 and -20KHz.

The 4th output is the power spectral density of the modulated signal which is collected at the amitec source to a QT GUI frequency sink. This output is just a representation of dsbsc signal having its peak at 445 and 465 Khz, which matchess with the existing theory.

D) SSBSC MODULATION AND DEMODULATION USING CHANNEL MODEL:



The above image is the channel modulated block diagram in GNU Radio. This is the framework from which we will be deriving the real time implementation of SSBSC (Single side band supressed carrier) signals.

In the transmitter side of the SSBSC channel model we have the signal source which will generating the message signal in form of a cosine wave.

$$m(t) = A_m \cos(2\pi f_m t)$$

We have another carrier signal which will be used to modulate the message signal. Carrier signal will be having a higher frequency than the message signal.

$$c(t) = A_c \cos(2\pi f_c t)$$

The resultant SSBSC modulated wave can be represented by:

$$s(t) = \frac{A_m A_c}{2}\cos[2\pi(f_c + f_m)t]$$ for upper side band

Or

$$s(t) = \frac{A_m A_c}{2}\cos[2\pi(f_c - f_m)t]$$ for lower side band.

Similarly in the receiver side in the back end of the channel model, the already existing SSBSC signal will be demodulated using coherent detector by multiplying another shifted cosine signal and a low pass filter. Details of the receiver side will be given elaborately in the real time implementation in the receiver side.

Let the SSBSC wave be

$$s(t) = \frac{A_m A_c}{2}\cos[2\pi(f_c - f_m)t]$$

The output of the local oscillator is

$$c(t) = A_c \cos(2\pi f_c t)$$

The input to the low pass filter is given by:

$$v(t) = s(t)\,c(t)$$

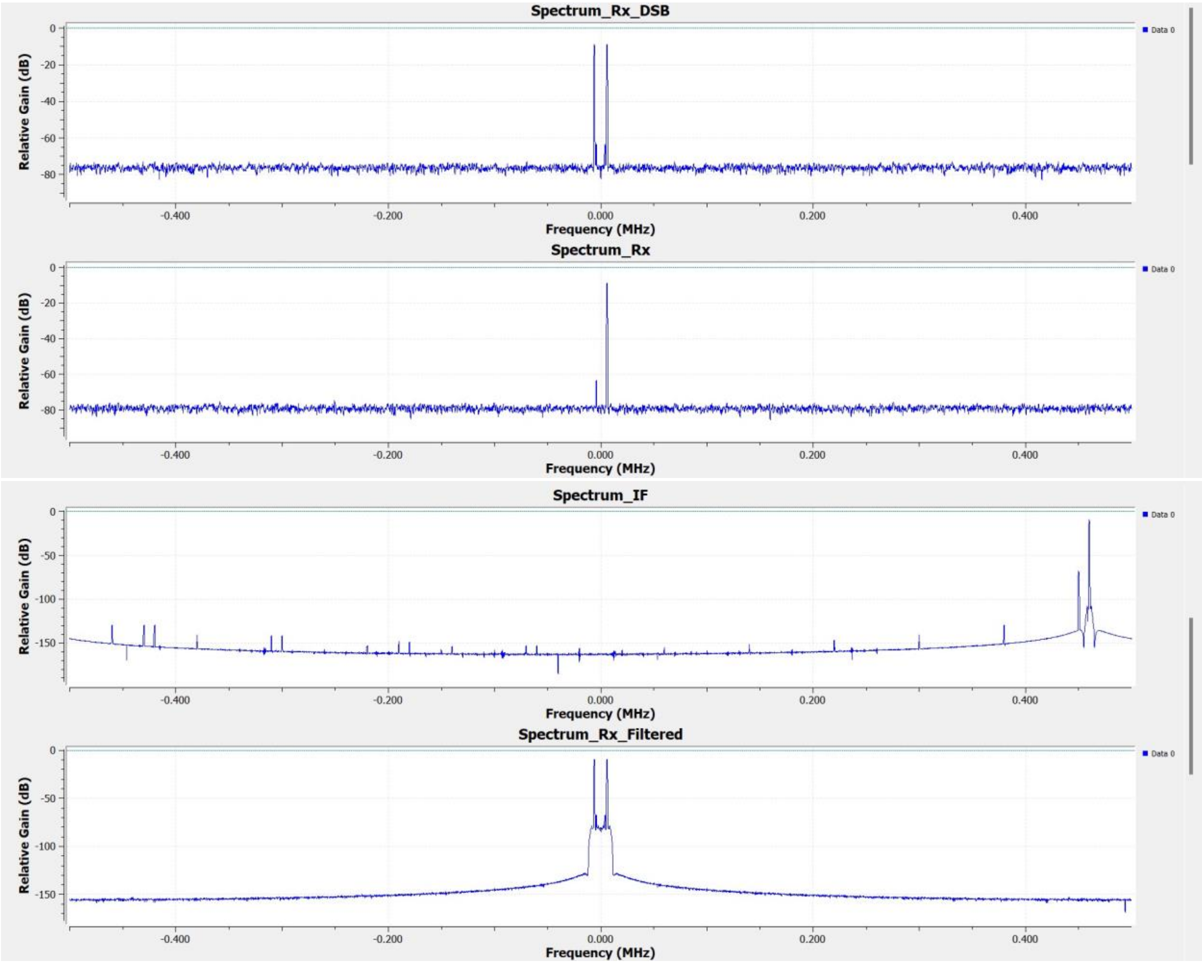
Now substituting the equations, we get the signal which goes into the low pass filter.

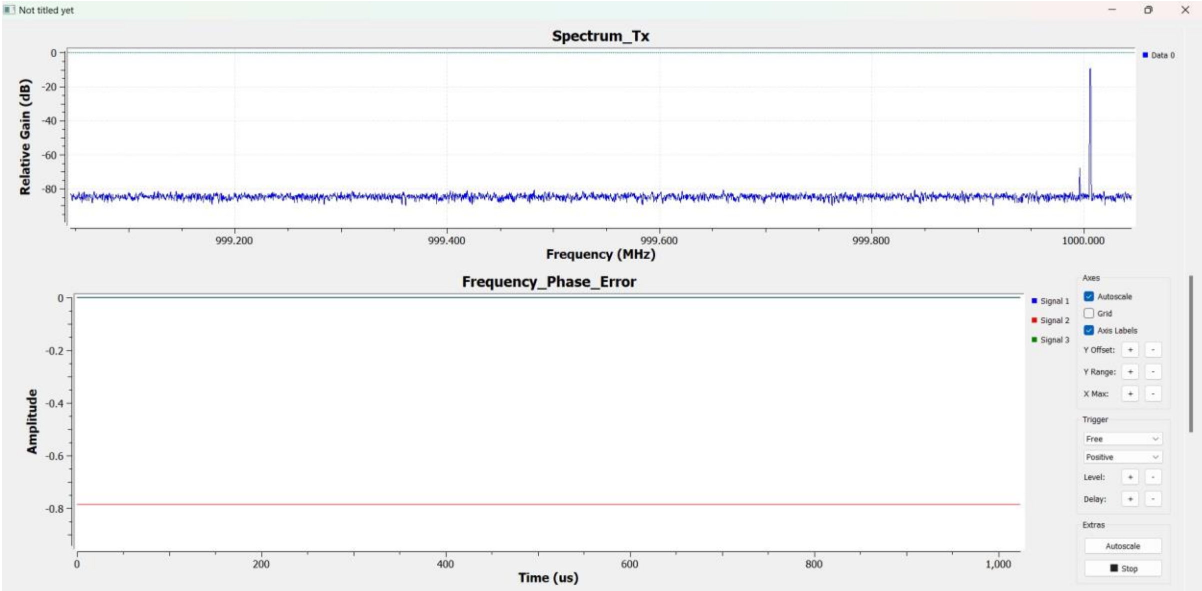
$$v(t) = \frac{A_m A_c^2}{4}\cos(2\pi f_m t) + \frac{A_m A_c^2}{4}\cos[2\pi(2f_c - f_m)t]$$

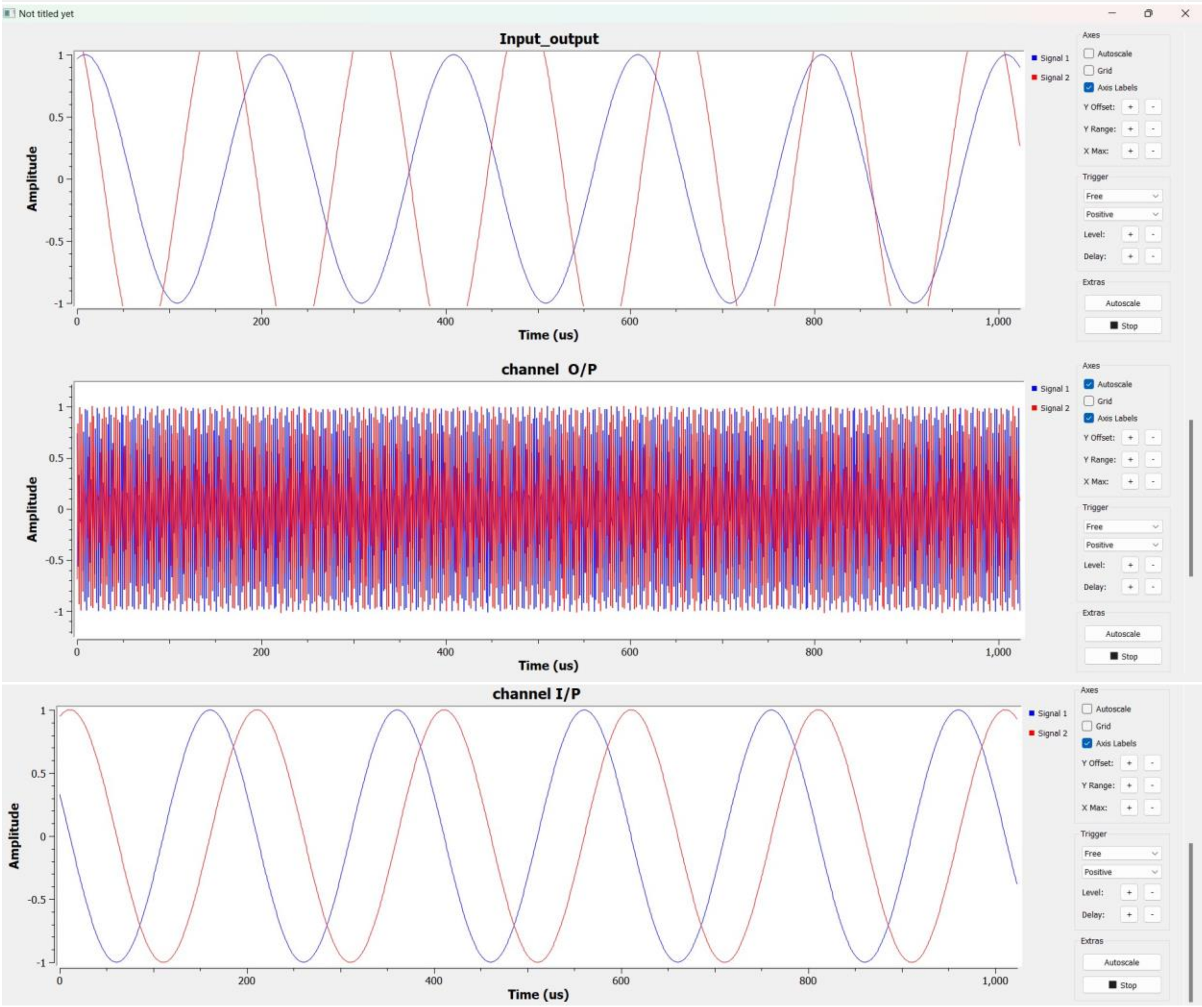
The output of the low pass filter is given by

$$v_0\,(t) = \frac{A_m {A_c}^2}{4}\cos(2\pi f_m t)$$

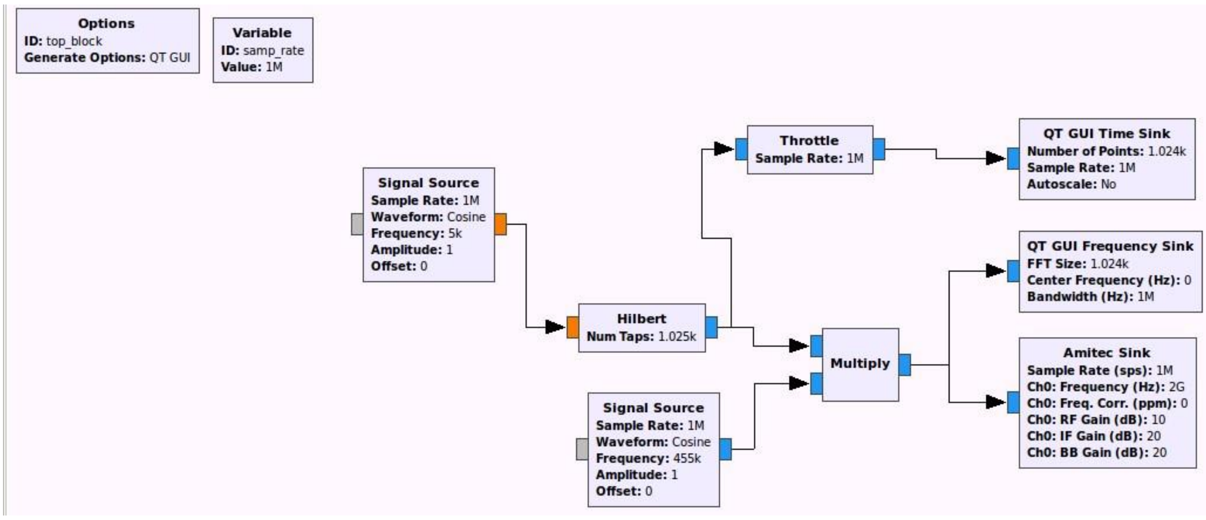
Hence we get the original cosine wave form after demodulation. In the real time analysis, we will be discussing the working of each and every block.




Not titled yet
Spectrum_Tx
Relative Gain (dB)
Frequency (MHz)
Data 0
Frequency_Phase_Error
Amplitude
Time (us)
Signal 1
Signal 2
Signal 3
Axes
Autoscale
Grid
Axis Labels
Y Offset:
Y Range:
X Max:
Trigger
Free
Positive
Level:
Delay:
Extras
Autoscale
Stop


Not titled yet
Input_output
channel O/P
channel I/P
Amplitude
Time (us)
Signal 1
Signal 2
Axes
Autoscale
Grid
Axis Labels
Y Offset:
Y Range:
X Max:
Trigger
Free
Positive
Level:
Delay:
Extras
Autoscale
Stop

E) REAL TIME SSBSC MODULATED SIGNAL ANALYSIS FROM THE TRANSMITTER:



This is the transmitter block diagram for SSBSC signal modulation. Here we use signal source with 1Mhz sampling rate which produces a cosine signal at a frequency of 5Khz and amplitude 1 volts. We can use either sine of cosine wave, but as in the previously mentioned ssbsc equation as we have used cosine, so we take a cosine signal.
Signal source main objective is to produce the message signal which will be further modulated.

Now we use hilbert transform block. We know that hilbert transform of cosine signal is negative sine signal. The hilbert transform block here performs two operations, which is first it transforms the original message signal and after that multiplies the original signal with the the transformed signal and produces the output. The hilbert transform also converts floating point to complex representation.

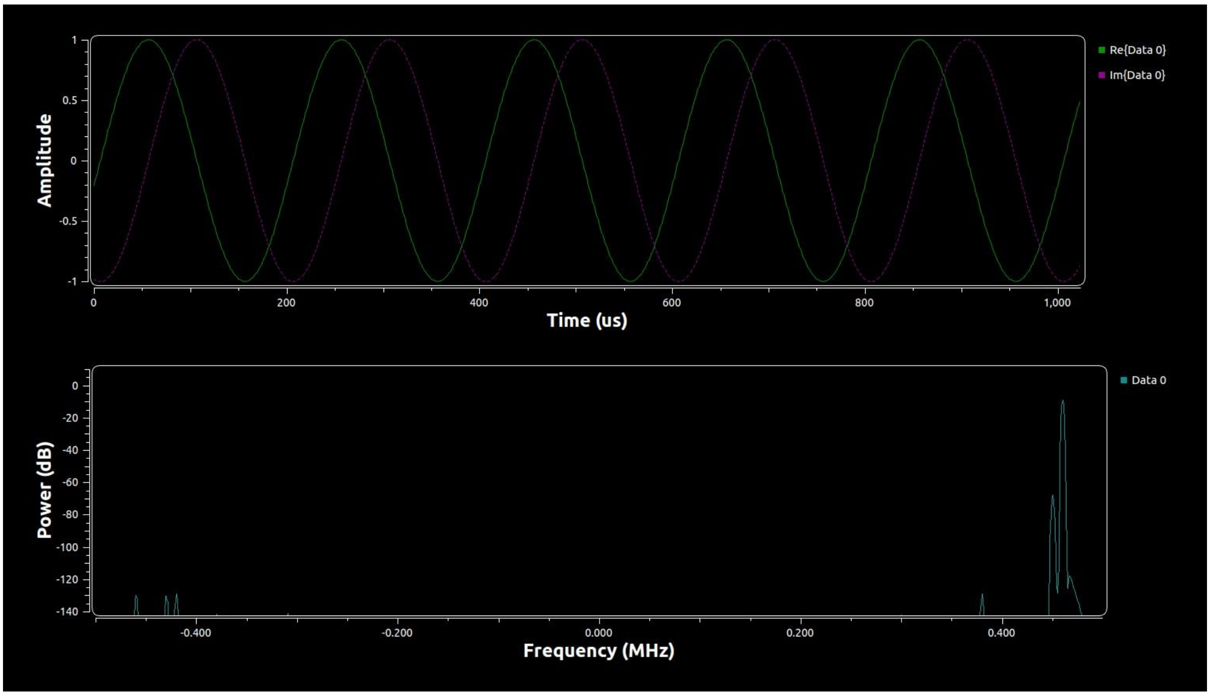
The above output is multiplied with the another signal which is the carrier signal with a frequency of 455KHz. The output of this stage gives the original SSBSC signal.

We use the throttle block to accelerate the complex signal in order to display the less disturbed signal. Throttle block is used in order to limit the rate at which that source block creates samples. We use 1Mhz sampling rate for the throttle block. So basically the unthrottled signal will have much faster transition rate compared to throttled output.

The gui time sink takes the throttled output of the signal, takes a set of complex streams and plots them in time domain. Here we have not provided the autoscale option so the graph resets only if we re run the program. The number of points taken here is 1024 point. It gives an rough estimate of the spacing and clarity of the graph. More the point, more is the clarity over the graph. As the throttle rate is 1Mhz, we take the sampling rate of time sink similar to the throttle rate.

The QT GUI frequency sink plots the power spectrum density of the modulated signal .The center frequency is defined at 0Hz as we observe in the output graph given below. Number of points and autoscale options are set according the QT GUI time sink.

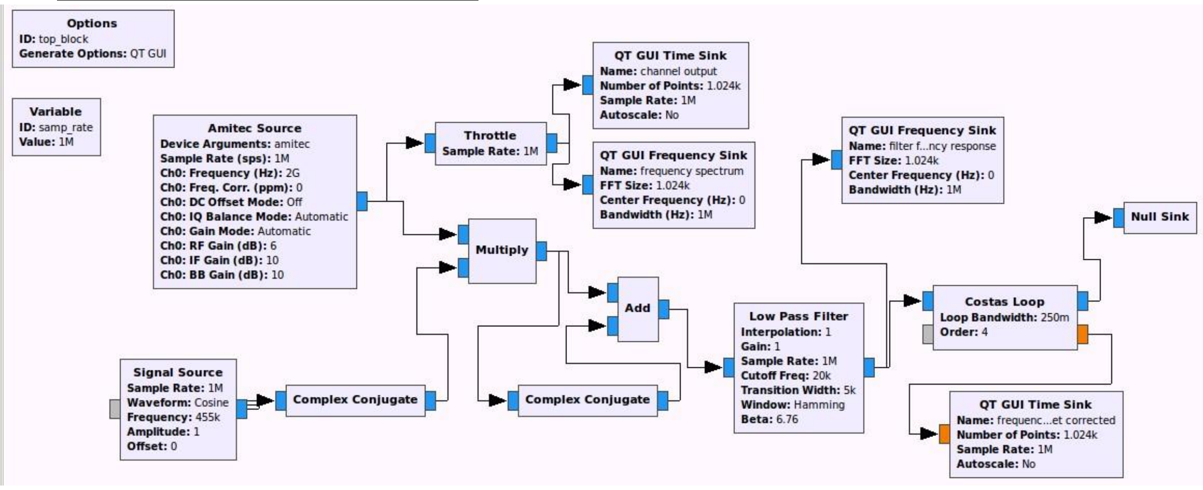
Amitec sink is the interfacing between the software simulation and the hardware software defined radio. It enables the output of the modulated signal from the software be transmitted to the SDR and then through the Dipole antenna in order for the signal to transmit to the receiving end. Amitec sink works under 0.4 GHz to 4GHz, hence we hace taken the working frequency of 2GHz. Further radio frequency gains, infrared frequency gains and baseband gains are required as the signal strength goes down when transmitted. So these gains helps in retriving the signal to the best possible extent.



Here in the 1st output we have a sinusoidal output. This is because we have directly connected the throttled Hilbert transformed cosine signal to the time sink. Here we get a sine wave and a cosine wave as Hilbert transform has multiplied the original cosine wave and the sine wave, thus indicating two waves with different colours.

In the 2nd output we have plotted the power spectrum density of the SSBSC Modulated signal. As this is SSBSC signal, we should theoretically be having a supressed/no carrier with one side band with higher peak and other sideband partially or fully supressed. This theory can be verified here as the real time output of the transmitter side correctly matches with the theory. Also the upper sideband must be present at fc+fm , which is at 465KHz. This also can be practically seen here.

F) REAL TIME SSBSC DEMODULATED SIGNAL ANALYSIS FROM THE RECIEVER:



In the receiver side, we demodulate the signal to get back the original message signal. Now from the amitec sink and dipole antenna, the receiving dipole antenna collects the signal and sends it back to the receiving software part using amitec source. The amitec sink and source must work in sync, and hence they should have the same specifications.

Again we use throttle here as the signal has been transmitted and received, so there must be lot of disturbances in the signal, which is reduced by the throttle. This signal is again fed to QT GUI time and frequency sink.

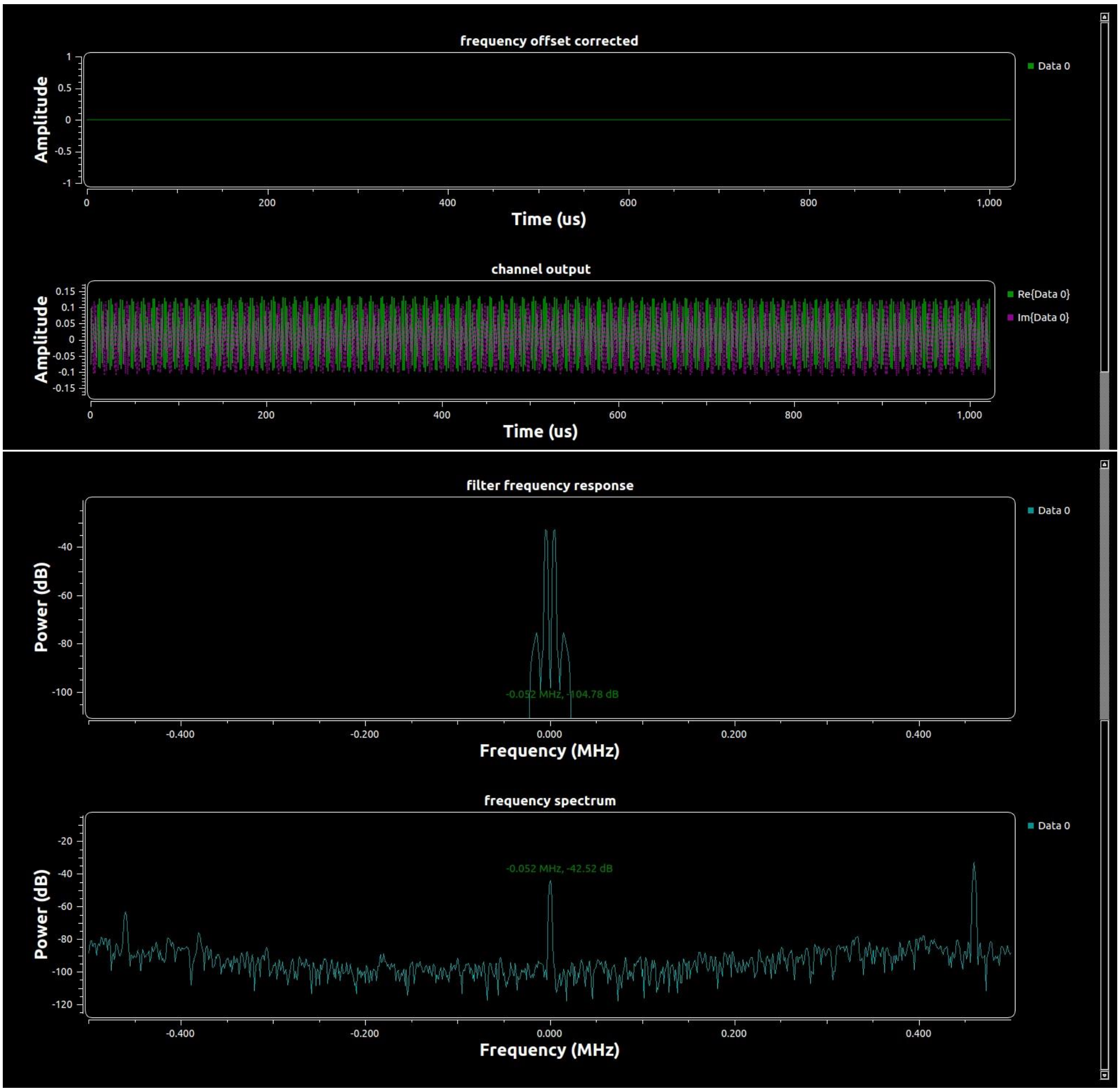
Here to demodulate we use the costas loop demodulation method similar to DSBSC demodulation. So we first multiply the received SSBSC signal with another carrier signal which is complex conjugated. Now as we have used Hilbert transform in the modulation side, we add the multiplied signal with its conjugate signal in order to remove the effect of the transform (indirectly performing inverse Hilbert transform as the block is not available in the older version of gnu companion).

Now pass it through the low pass filter as mention in demodulation theory. As we use simple LPF filter, the decimation and interpolation factor is taken as 1. Sample rate is also the same as that of throttle block.

Also here as FIR Filter is used, we define a window with one of the best compensations. So we use hamming window (blackmann windows can also be used). The beta parameter is only used for kaiser window, but the default value for beta is 6.76. The cutoff frequency denotes the frequency over which the signal components will not be transferred further or will be cut off. Transition width describes about the difference between the passband and stopband frequencies. Here we have taken TW as 5KHz.

The above filtered signal is further applied to the costas loop. Here we get a problem regarding the carrier frequency offset. This offset is removed using the costas loop. Carrier frequency offset occurs when the signal after decimation in the low pass filter do not synchronise with the carrier signal which we have multiplied initially.

This offset is reduced by the costas loop block, which is having a bandwidth of 250mHz and a 4th order loop. The blue output of the costas loop block indicates the frequency output which is given to the null sink, to eliminate frequency offset. The gray box output of the costas loop is the frequency compensated signal which is then displayed in the time sink. This time sink will have the demodulated original message signal which is the sinusoidal signal.



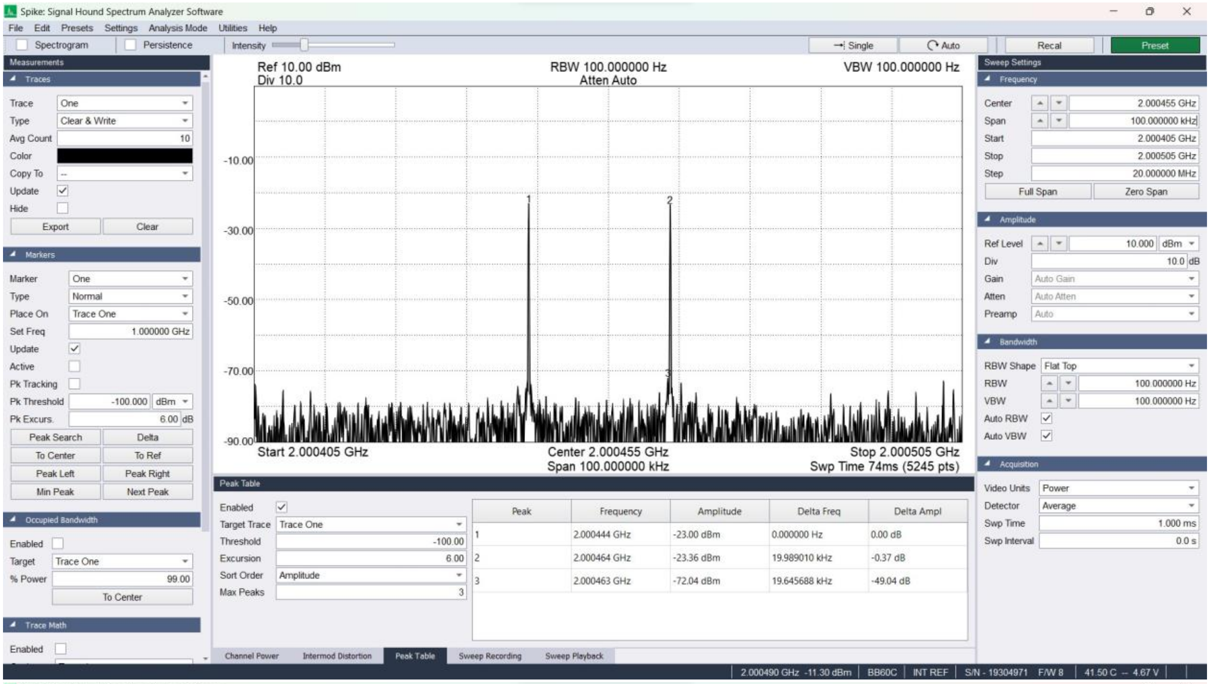
In the output, the 1st output depicts the frequency offset present in the demodulated signal. As the pattern tends to zero/constant it clearly shows that the output of the costas loop has given us an offset free signal.

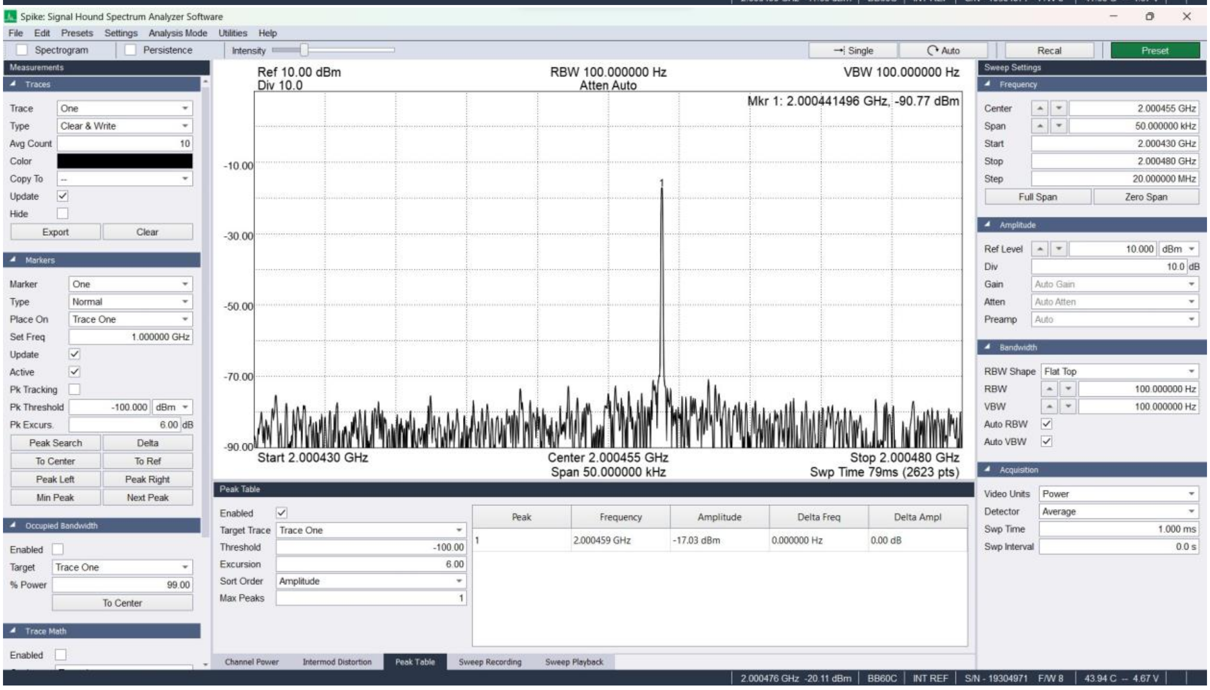
The 2nd output is the channel output which is directly taken from the amitec source to QT GUI time sink. It shows us the time modulated ssbsc signal. As we have taken into account the complex part of the signal also, the upper have of the ssbsc signal consists of the real part and the lower half gives the details on the imaginary part.

The 3rd output is the low pass filter output at QT GUI frequency sink. As the cutoff frequency of the low pass filter is 20KHz, we get 2 peaks on either side of center frequency as while removing the transform effect, we have added two signals. This has led to formation of 2 full peaks and 2 supressed peaks of ssbsc signals with the domaination of lower sideband.

The 4th output is the power spectral density of the modulated signal which is collected at the amitec source to a QT GUI frequency sink. This output is just a representation of ssbsc signal having its peak at 465 Khz, which matchess with the existing theory (upper sideband).

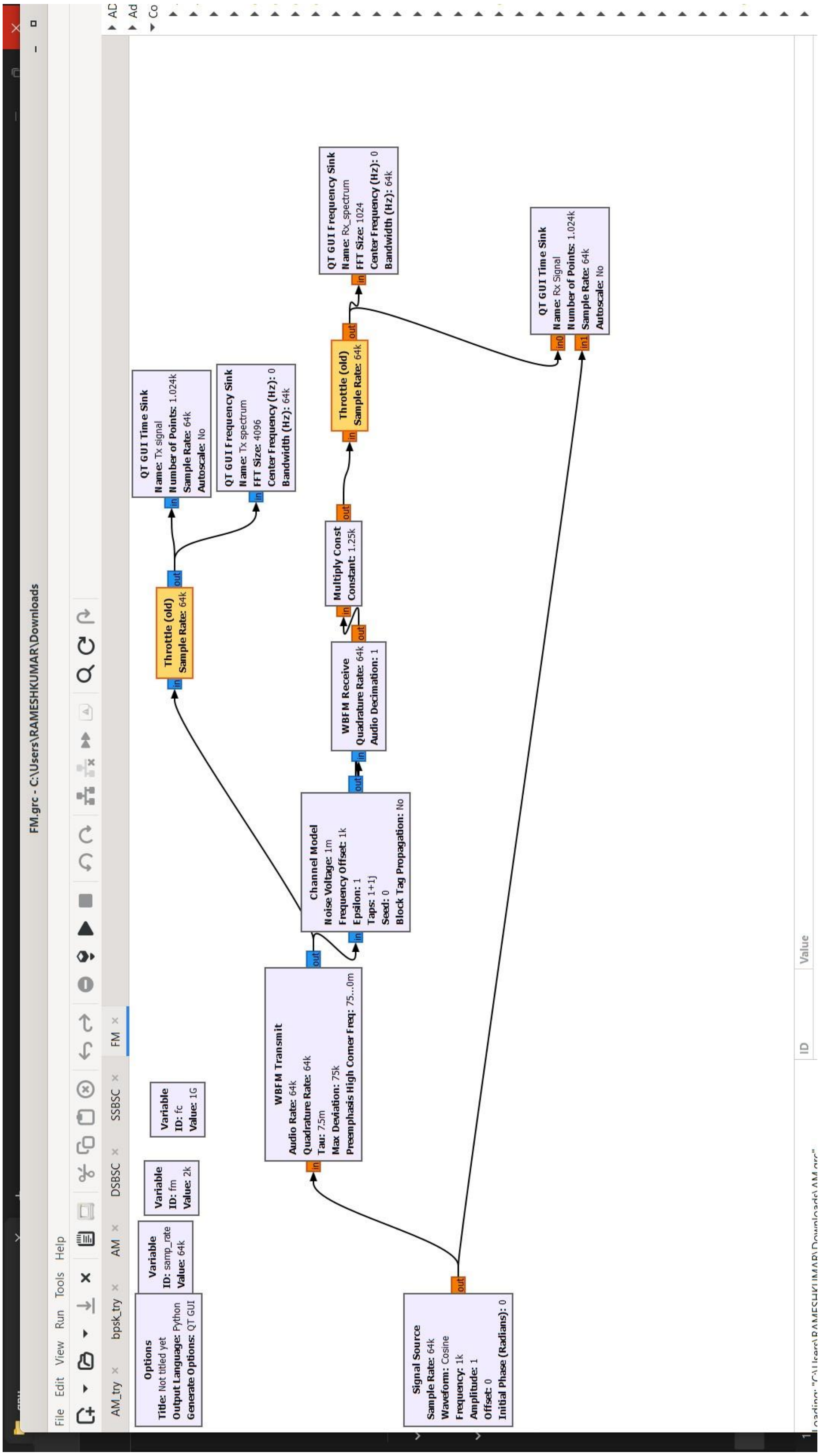
G) <u>REAL TIME DSBSC AND SSBSC MODULATED SIGNA ANALYSIS USING SPECTRUM ANALYSER:</u>





WEEK 4

A) WBFM AND NBFM MODULATION AND DEMODULATION USING CHANNEL MODEL:



We know that the standard equation of FM wave is:

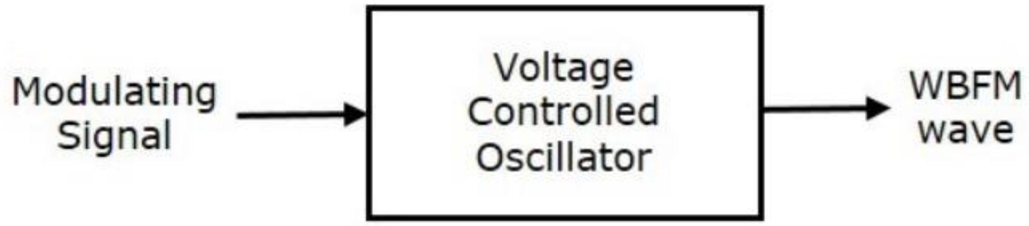
$$s(t) = A_c \cos\left(2\pi f_c t + 2\pi k_f \int m(t)\,dt\right)$$

$$\Rightarrow s(t) = A_c \cos(2\pi f_c t)\cos(2\pi k_f \int m(t)\,dt) -$$

$$A_c \sin(2\pi f_c t)\sin(2\pi k_f \int m(t)\,dt)$$

DIRECT METHOD

This method is called as the Direct Method because we are generating a wide band FM wave directly. In this method, Voltage Controlled Oscillator (VCO) is used to generate WBFM. VCO produces an output signal, whose frequency is proportional to the input signal voltage. This is similar to the definition of FM wave. The block diagram of the generation of WBFM wave is shown in the following figure.



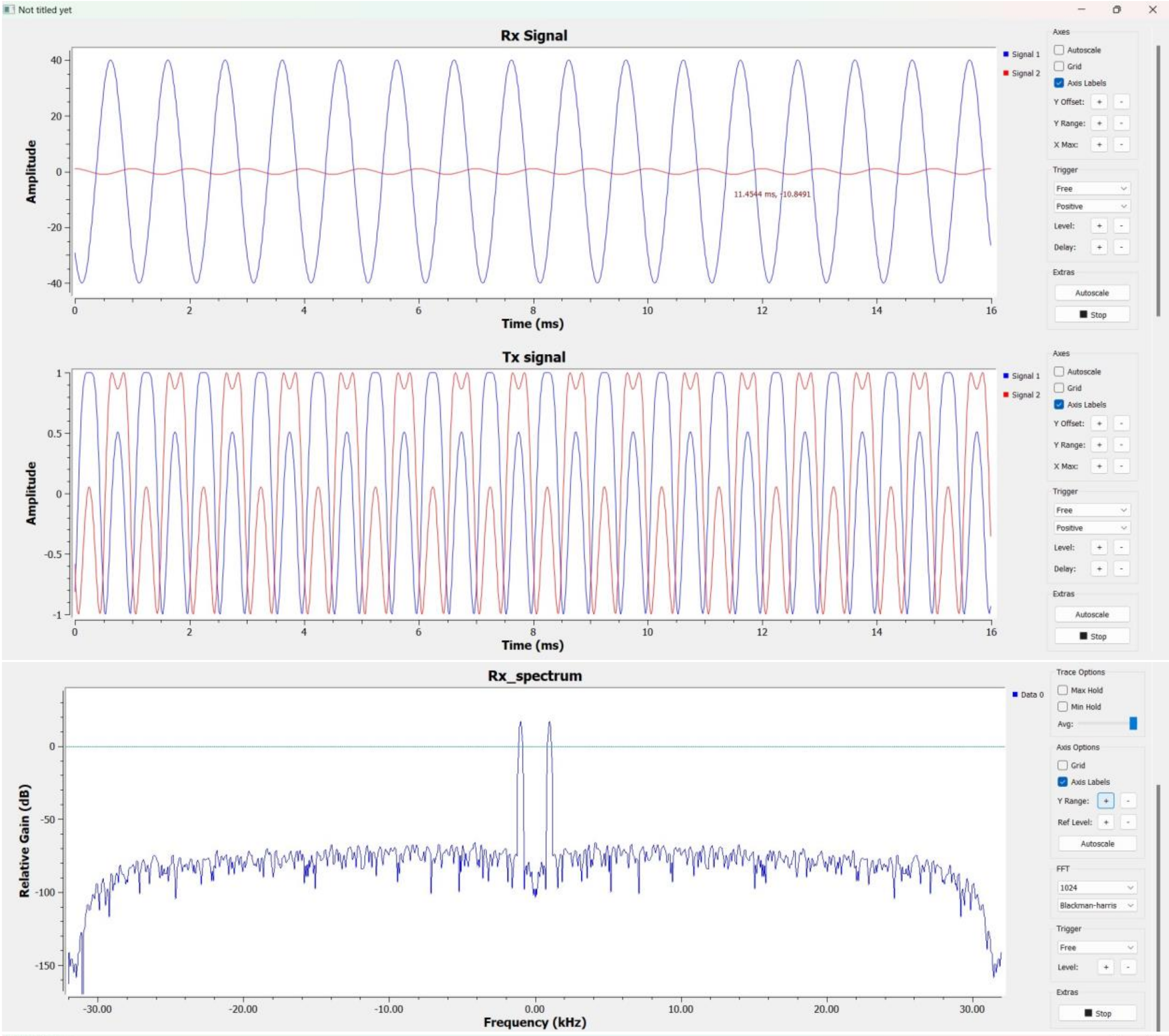
Here, the modulating signal m(t) is applied as an input of Voltage Controlled Oscillator (VCO). VCO produces an output, which is nothing but the WBFM.

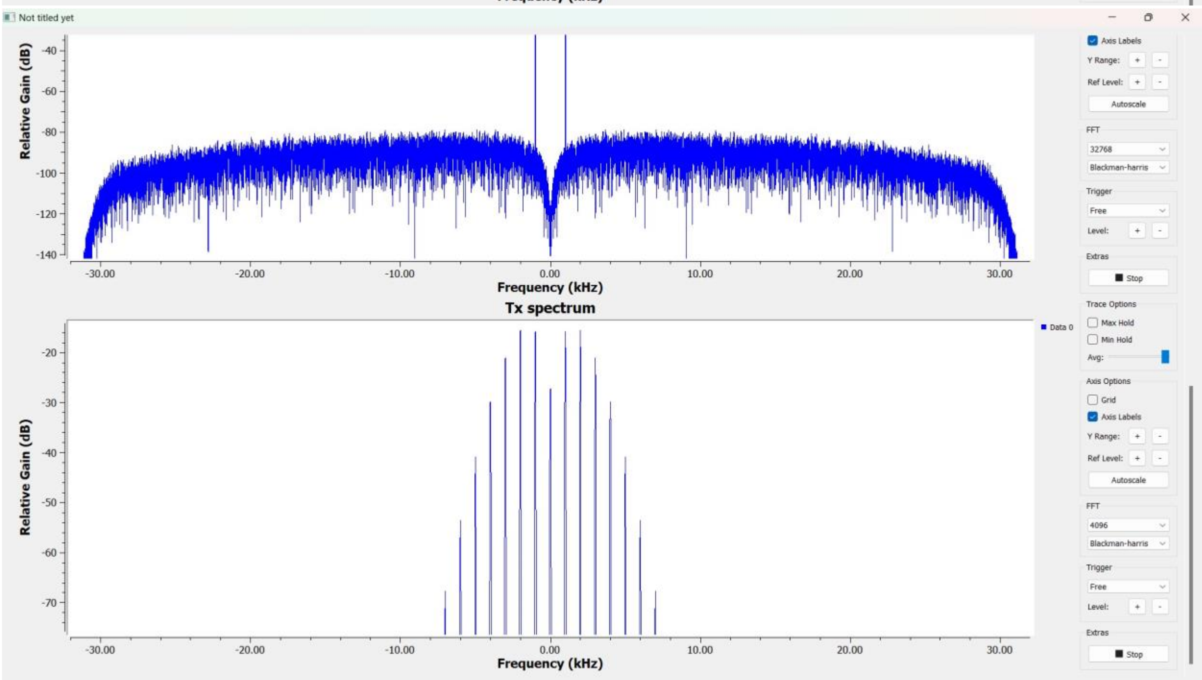
$$f_i \; \alpha \; m(t)$$

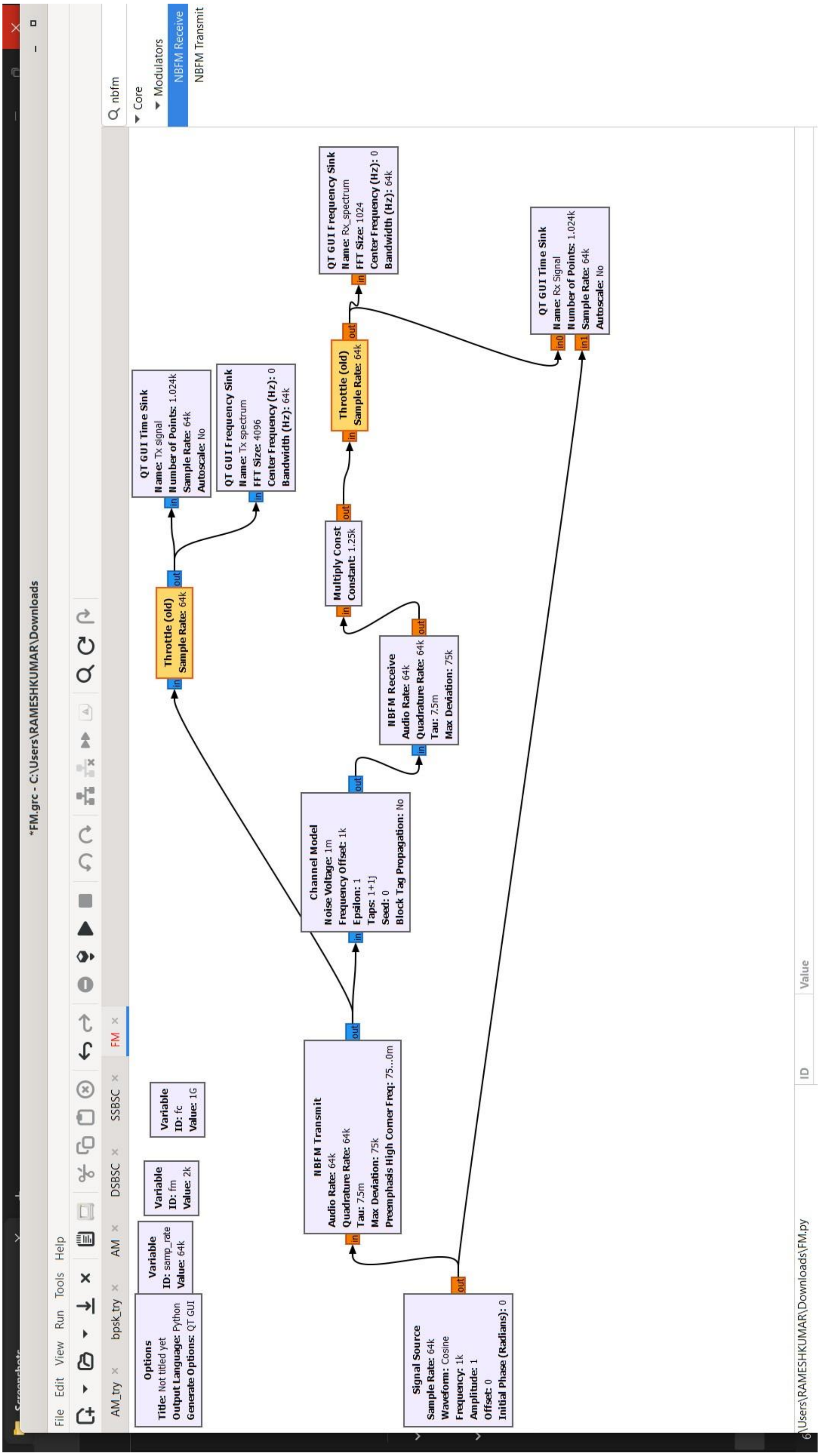
$$\Rightarrow f_i = f_c + k_f m(t)$$

Where,

fi is the instantaneous frequency of WBFM wave.


Not titled yet
Rx Signal
Amplitude
Time (ms)
Signal 1
Signal 2
11.4544 ms, -10.8491
Axes
Autoscale
Grid
Axis Labels
Y Offset:
Y Range:
X Max:
Trigger
Free
Positive
Level:
Delay:
Extras
Autoscale
Stop
Tx signal
Amplitude
Time (ms)
Signal 1
Signal 2
Rx_spectrum
Relative Gain (dB)
Frequency (kHz)
Data 0
Trace Options
Max Hold
Min Hold
Avg:
Axis Options
Ref Level:
FFT
1024
Blackman-harris


Not titled yet
Relative Gain (dB)
Frequency (kHz)
Axis Labels
Y Range:
Ref Level:
Autoscale
FFT
32768
Blackman-harris
Trigger
Free
Level:
Extras
Stop
Tx spectrum
Data 0
Trace Options
Max Hold
Min Hold
Avg:
Axis Options
Grid
4096


*FM.grc - C:\Users\RAMESHKUMAR\Downloads
File Edit View Run Tools Help
AM_try bpsk_try AM DSBSC SSBSC FM
Options
Title: Not titled yet
Output Language: Python
Generate Options: QT GUI
Variable
ID: samp_rate
Value: 64k
Variable
ID: fm
Value: 2k
Variable
ID: fc
Value: 1G
Signal Source
Sample Rate: 64k
Waveform: Cosine
Frequency: 1k
Amplitude: 1
Offset: 0
Initial Phase (Radians): 0
NBFM Transmit
Audio Rate: 64k
Quadrature Rate: 64k
Tau: 7.5m
Max Deviation: 75k
Preemphasis High Corner Freq: 75...0m
Channel Model
Noise Voltage: 1m
Frequency Offset: 1k
Epsilon: 1
Taps: 1+1j
Seed: 0
Block Tag Propagation: No
Throttle (old)
Sample Rate: 64k
QT GUI Time Sink
Name: Tx signal
Number of Points: 1.024k
Sample Rate: 64k
Autoscale: No
QT GUI Frequency Sink
Name: Tx spectrum
FFT Size: 4096
Center Frequency (Hz): 0
Bandwidth (Hz): 64k
NBFM Receive
Audio Rate: 64k
Quadrature Rate: 64k
Tau: 7.5m
Max Deviation: 75k
Multiply Const
Constant: 1.25k
Throttle (old)
Sample Rate: 64k
QT GUI Frequency Sink
Name: Rx_spectrum
FFT Size: 1024
Center Frequency (Hz): 0
Bandwidth (Hz): 64k
QT GUI Time Sink
Name: Rx Signal
Number of Points: 1.024k
Sample Rate: 64k
Autoscale: No
Q nbfm
Core
Modulators
NBFM Receive
NBFM Transmit
ID
Value
\Users\RAMESHKUMAR\Downloads\FM.py

We know that the standard equation of FM wave is:

$$s(t) = A_c \cos\left(2\pi f_c t + 2\pi k_f \int m(t)\,dt\right)$$

$$\Rightarrow s(t) = A_c \cos(2\pi f_c t)\cos(2\pi k_f \textstyle\int m(t)\,dt) -$$
$$A_c \sin(2\pi f_c t)\sin(2\pi k_f \textstyle\int m(t)\,dt)$$

For NBFM,

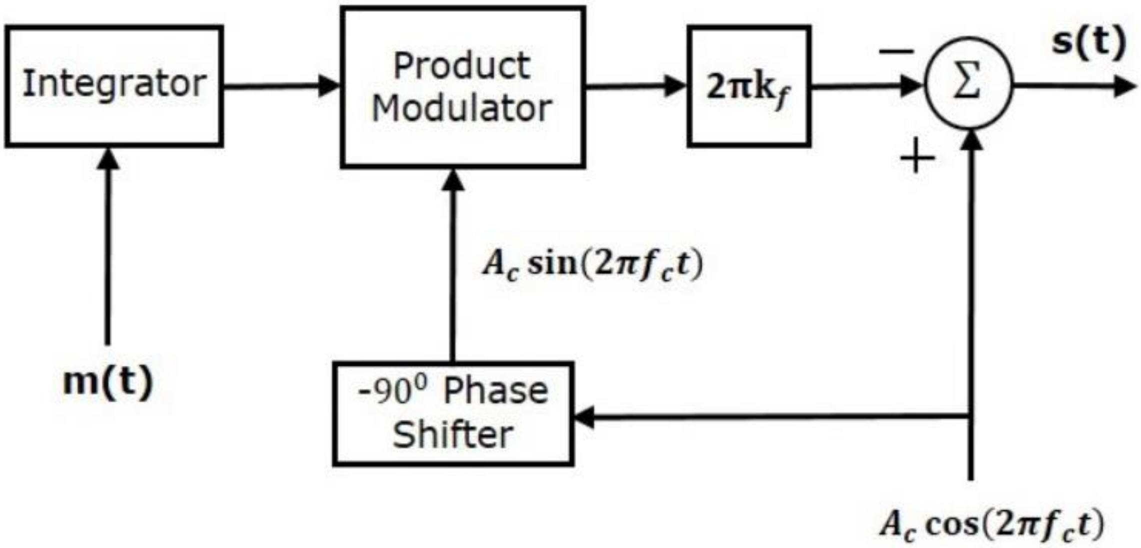
$$\left|2\pi k_f \int m(t)\,dt\right| << 1$$

We know that cosθ≈1 and sinθ≈1 when θ is very small.

By using the above relations, we will get the **NBFM equation** as

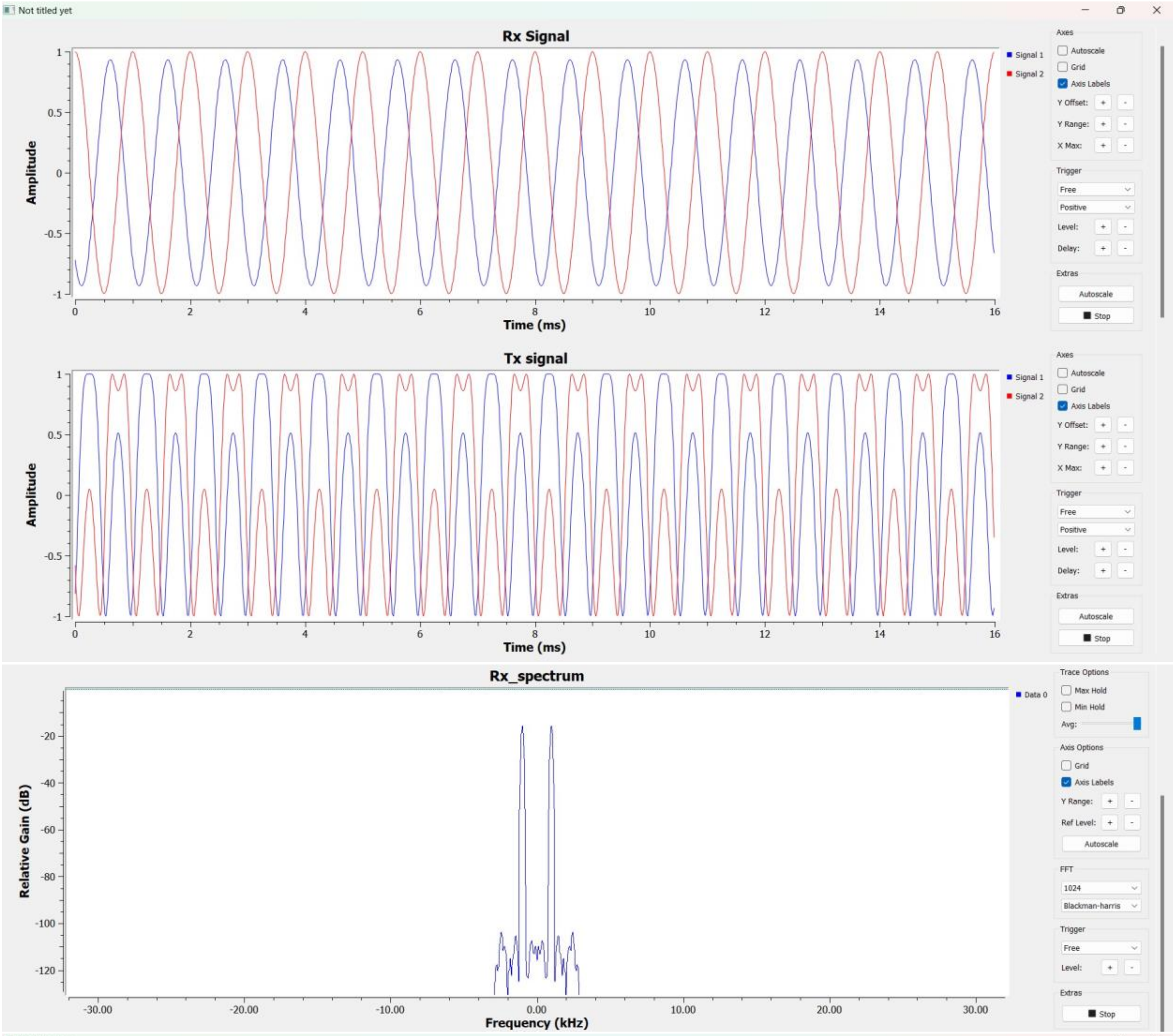
$$s(t) = A_c \cos(2\pi f_c t) - A_c \sin(2\pi f_c t) 2\pi k_f \int m(t)\,dt$$

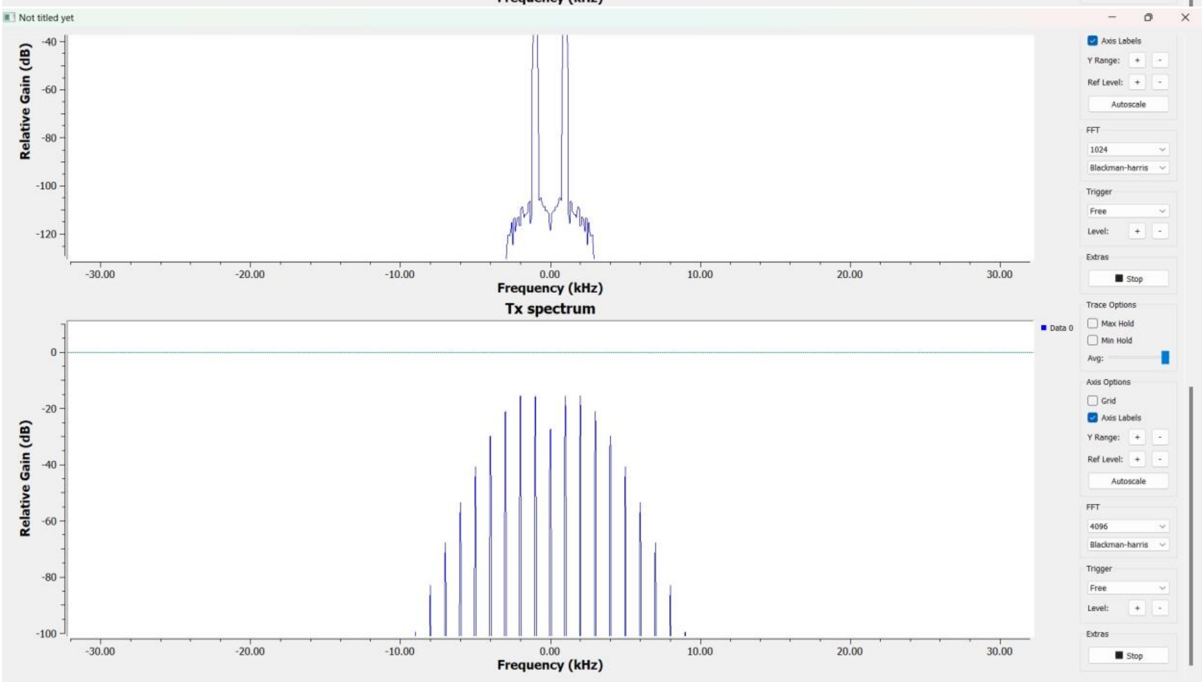
The block diagram of NBFM modulator is shown in the following figure.



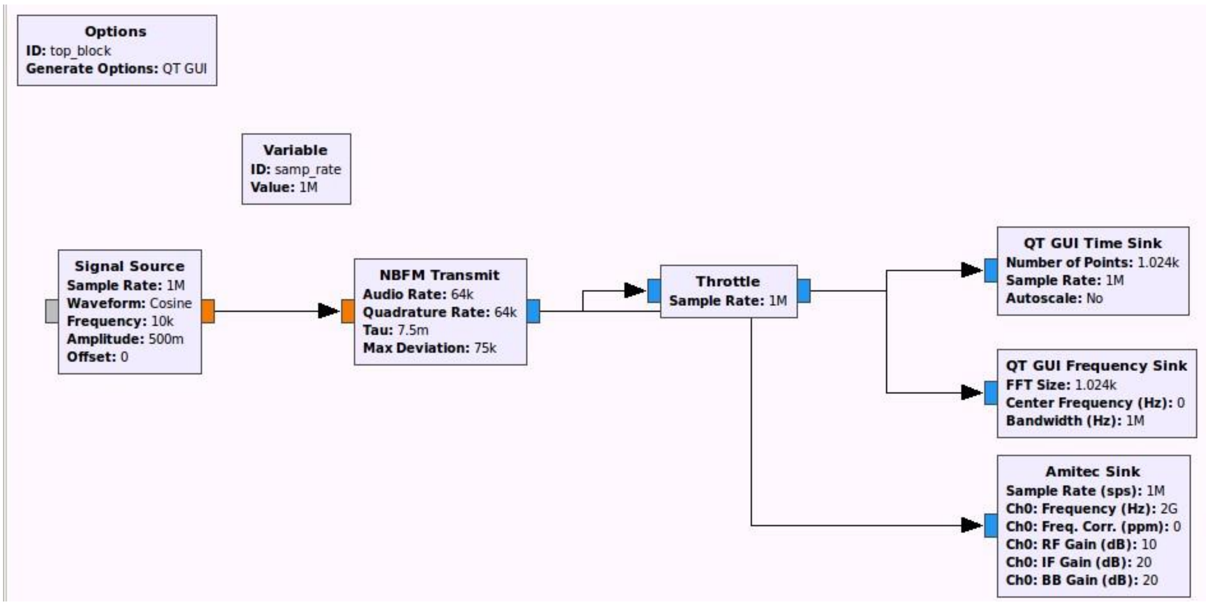
Here, the integrator is used to integrate the modulating signal m(t). The carrier signal Accos(2πfct) is the phase shifted by −900−900 to get Acsin(2πfct) with the help of −90 phase shifter. The product modulator has two inputs ∫m(t)dt∫ and Acsin(2πfct). It produces an output, which is the product of these two inputs.

This is further multiplied with 2πkf2 by placing a block 2πkf2 in the forward path. The summer block has two inputs, which are nothing but the two terms of NBFM equation. Positive and negative signs are assigned for the carrier signal and the other term at the input of the summer block. Finally, the summer block produces NBFM wave.


Not titled yet
Rx Signal
Signal 1
Signal 2
Amplitude
Time (ms)
Tx signal
Signal 1
Signal 2
Amplitude
Time (ms)
Rx_spectrum
Data 0
Relative Gain (dB)
Frequency (kHz)
Axes
Autoscale
Grid
Axis Labels
Y Offset:
Y Range:
X Max:
Trigger
Free
Positive
Level:
Delay:
Extras
Autoscale
Stop
Trace Options
Max Hold
Min Hold
Avg:
Axis Options
Ref Level:
FFT
1024
Blackman-harris


Not titled yet
Relative Gain (dB)
Frequency (kHz)
Tx spectrum
Data 0
Relative Gain (dB)
Frequency (kHz)
Axis Labels
Y Range:
Ref Level:
Autoscale
FFT
1024
4096
Blackman-harris
Trigger
Free
Level:
Extras
Stop
Trace Options
Max Hold
Min Hold
Avg:
Axis Options
Grid

B) REAL TIME NBFM MODULATED SIGNAL ANALYSIS FROM THE TRANSMITTER:



Here we use signal source with 1Mhz sampling rate which produces a cosine signal at a frequency of 10Khz and amplitude 0.5 volts. We can use either sine of cosine wave, but as in the previously mentioned NBFM equation as we have used cosine, we take a cosine signal. Signal source main objective is to produce the message signal which will be further modulated.

Narrow Band FM Transmitter.

Takes a single float input stream of audio samples in the range [-1,+1] and produces a single FM modulated complex baseband output.

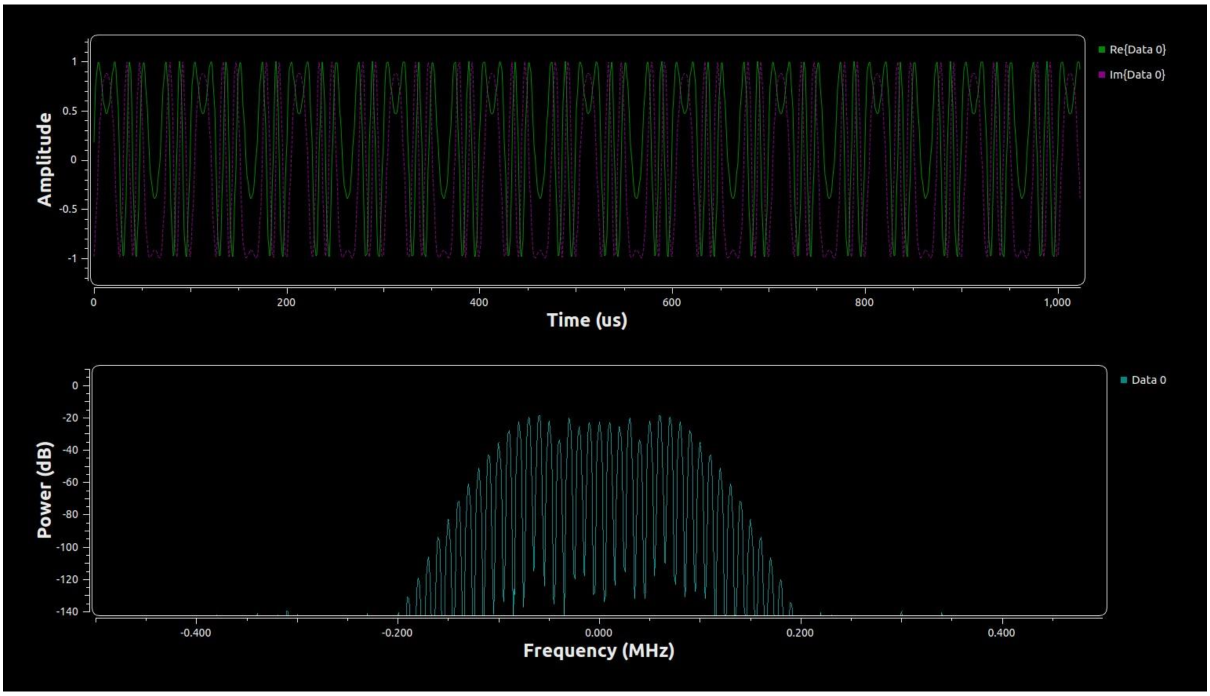
The only difference with WBFM Transmit is the size of the internal low pass filter for interpolation. Here it has a cutoff frequency of 4.5kHz with 2.5KHz of transition width.

We use the throttle block to accelerate the complex signal in order to display the modulated signal. Throttle block is used in order to limit the rate at which that source block creates samples. We use 1Mhz sampling ratw for the throttle block. So basically the unthrottled signal will have much faster transition rate compared to throttled output.

The gui time sink takes the throttled output of the signal, takes a set of complex streams and plots them in time domain. Here we have not provided the autoscale option so the graph resets only if we re run the program. The number of points taken here is 1024 point. It gives an rough estimate of the spacing and clarity of the graph. More the point, more is the clarity over the graph. As the throttle rate is 1Mhz, we take the sampling rate of time sink similar to the throttle rate.

The QT GUI frequency sink plots the power spectrum density of the modulated signal . The center frequency is defined at 0Hz as we observe in the output graph given below. Number of points and autoscale optiins are set according the QT GUI time sink.

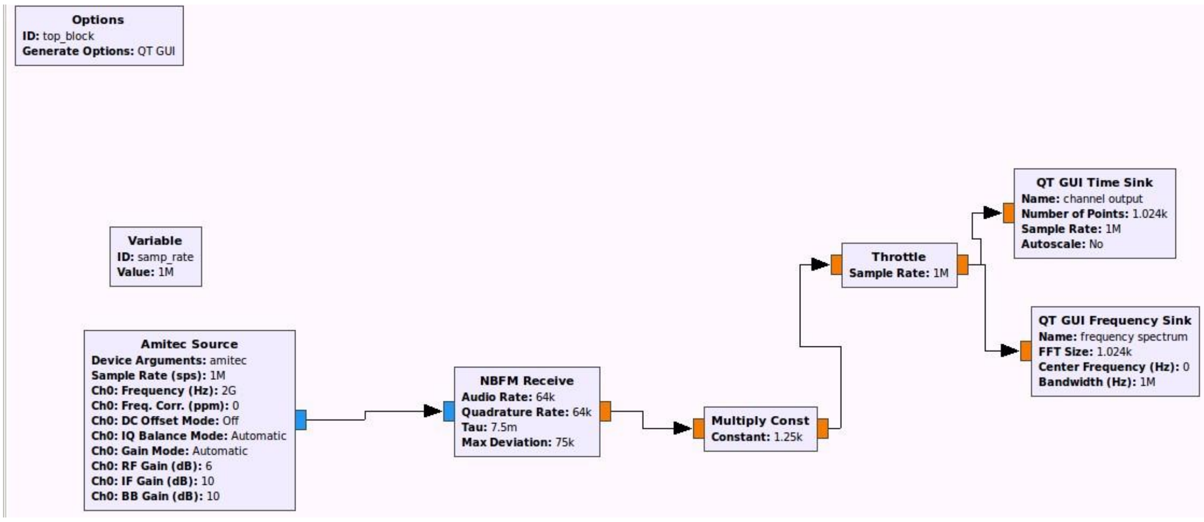
Amitec sink interfaces the software to the software defined radio. It enables the output of the modulated signal from the software be transmitted to the SDR and then through the Dipole antenna in order for the signal to transmit to the receiving end. Amitec sink works under 0.4 GHz to 4GHz, hence we hace taken the working frequency of 2GHz. Further radio frequency gains, infrared frequency gains and baseband gains are required as the signal strength goes down when transmitted. So these gains helps in retriving the signal to the best possible extent.



Here in the 1st output we have a nbfm Modulated output. This is because we have directly connected the throttled nbfm signal to the time sink..

In the 2nd output we have plotted the power spectrum density of the modulated signal. As this is NBFM signal, we should theoretically be having a peak at fc and fc+n*fm (where n is positive integers).

C) REAL TIME NBFM DEMODULATED SIGNAL ANALYSIS FROM THE RECIEVER:

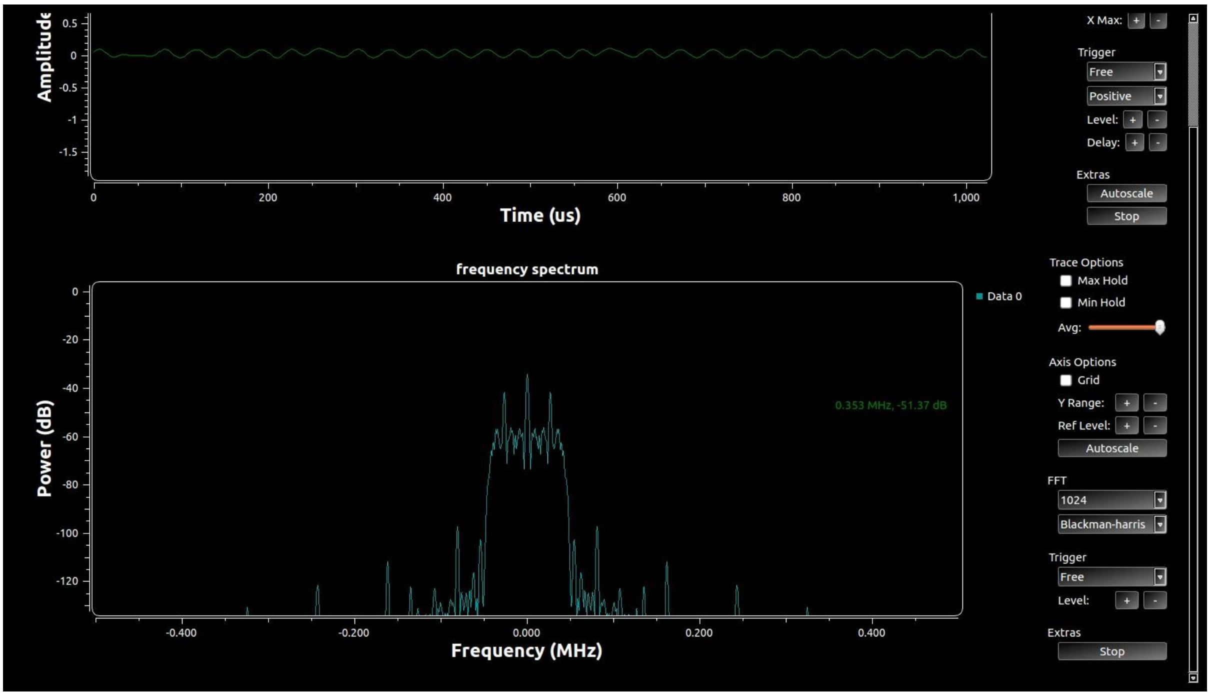


In the receiver side, we demodulate the signal to get back the original message signal. Now from the amitec sink and dipole antenna, the receiving dipole antenna collects the signal and sends it back to the receiving software part using amitec source. The amitec sink and source must work in sync, and hence they should have the same specifications.

Here to demodulate we use the NBFM receive block. Audio Rate **is** the sample rate of the input audio stream. Quadrature Rate is the sample rate of the input complex baseband signal in NBFM Receive block. Tau **is t**he Pre-emphasis time constant and the default value is 75e-5 (float). Max Deviation **is t**he maximum frequency deviation in Hertz and the default value is 75e-3 (float). We also use the multiplication factor to improve the amplitude of the signal.

Again we use throttle here as the signal has been transmitted and received, so there must be lot of disturbances in the signal, which is reduced by the throttle. This signal is again fed to QT GUI time and frequency sink.

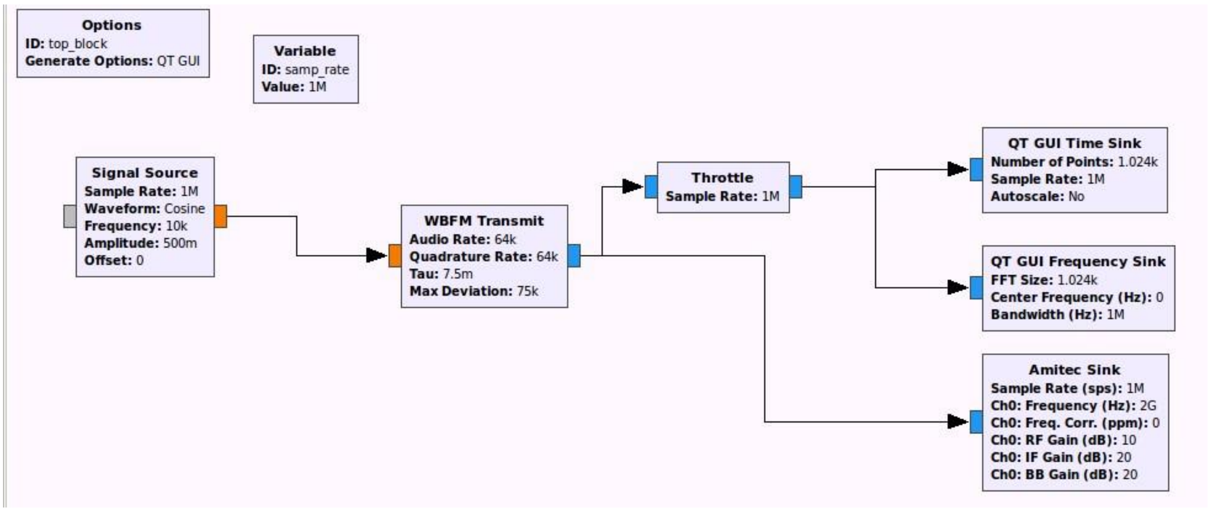
This produced signal is fed to QT GUI time and frequency sink with no of points 1024 which depicts the spacing and clarity of the output. Center frequency of the sink is given at 0khz and sample rate is 200Khz.



The 1st output is the channel output of is QT GUI time sink. It shows us the time demodulated NBFM signal (which is the message signal with a diminished amplitude due to amplitude factor).

The 2nd output is the power spectral density of the demodulated signal which is collected at the amitec source to a QT GUI frequency sink. This output is just a representation of message signal having its peak at 10khz, which matches with the existing theory.

D) REAL TIME WBFM MODULATED SIGNAL ANALYSIS FROM THE TRANSMITTER:

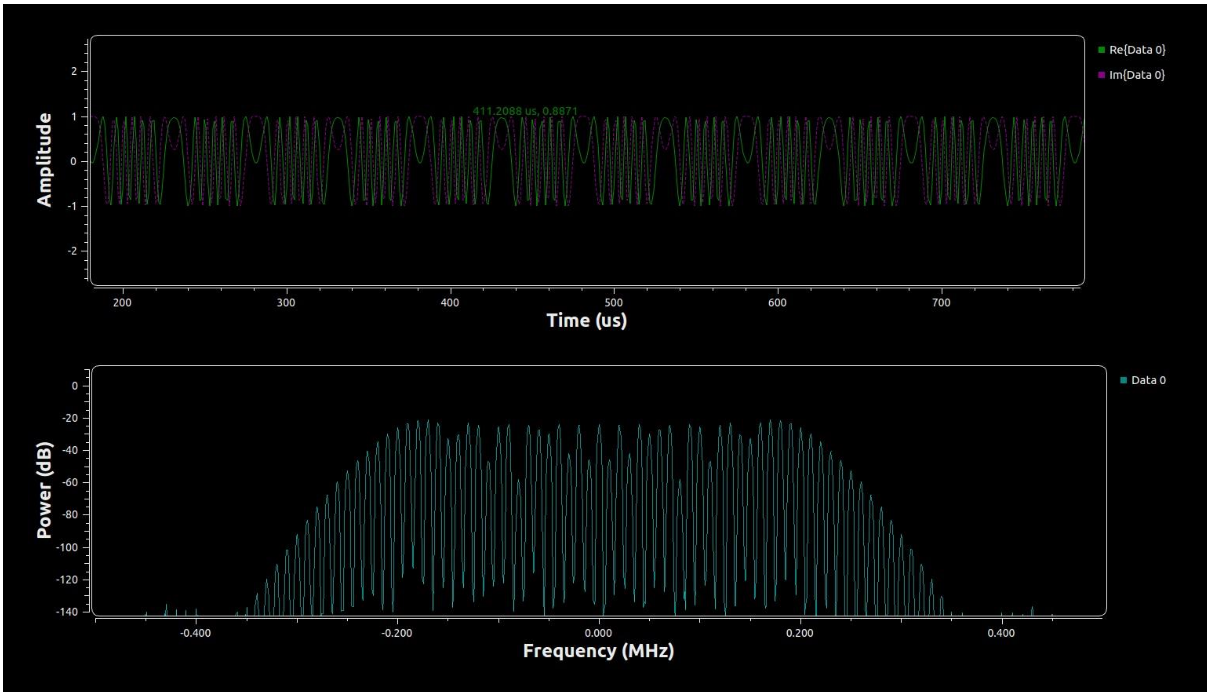


The WBFM uses the similar blocks and theory content of that of NBFM transmitter part, but the modulation index of wbfm is greater than 1, whereas that of nbfm is less than 1. Also the bandwidth of wbfm is approximately 15 times that of nbfm bandwidth.

Wide Band FM Transmit block takes a single float input stream of audio samples in the range [-1,+1] and produces a single FM modulated complex baseband output.

The only difference with NBFM Transmit is the size of the internal low pass filter for interpolation. Here it has a cutoff frequency of 16kHz with 2KHz of transition width.

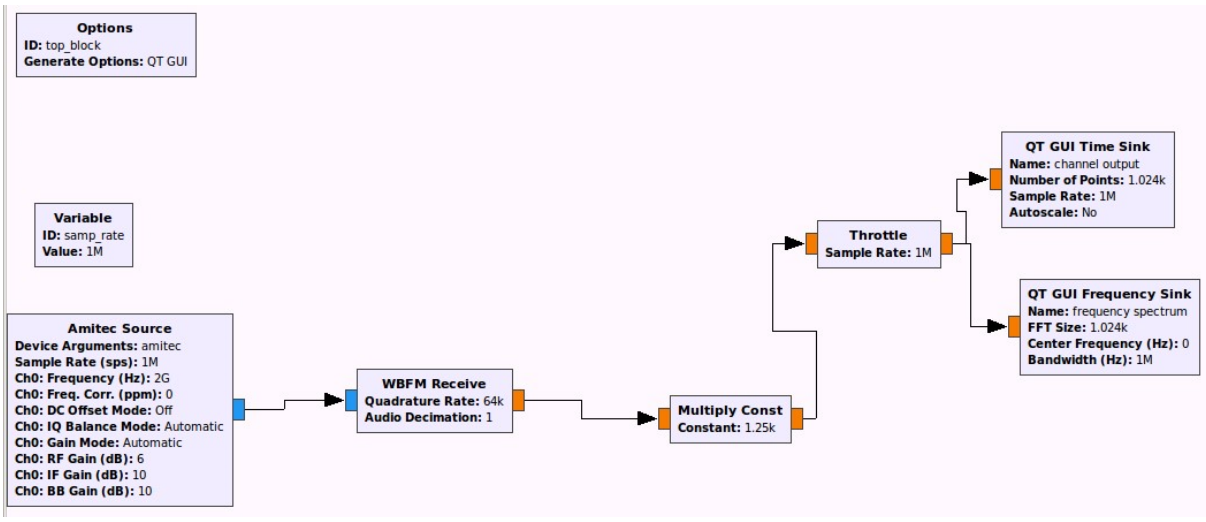
Audio Rate: Sample rate of audio stream, >= 16k (integer)
Quadrature Rate: Sample rate of output stream (integer). Must be an integer multiple of audio rate.
Tau: Pre-emphasis time constant (default 75e-6) (float)
Max Deviation: Maximum deviation in Hz (default 75e3) (float)



Here in the 1st output we have a wbfm Modulated output. This is because we have directly connected the throttled nbfm signal to the time sink..

In the 2nd output we have plotted the power spectrum density of the modulated signal. As this is NBFM signal, we should theoretically be having a peak at fc and fc+n*fm (where n is positive integers).

E) REAL TIME WBFM DEMODULATED SIGNAL ANALYSIS FROM THE RECIEVER:

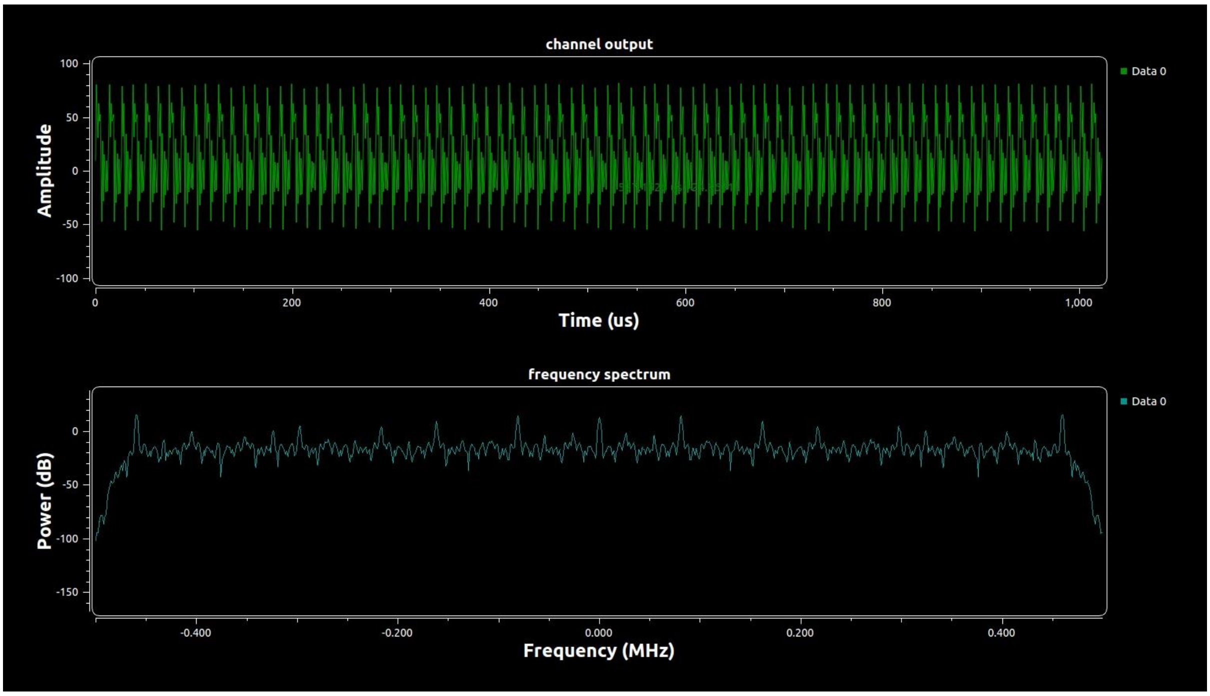


In the receiver side, we demodulate the signal to get back the original message signal. Now from the amitec sink and dipole antenna, the receiving dipole antenna collects the signal and sends it back to the receiving software part using amitec source. The amitec sink and source must work in sync, and hence they should have the same specifications.

Here to demodulate we use the WBFM receive block. Audio Rate **is** the sample rate of the input audio stream. Quadrature Rate is the sample rate of the input complex baseband signal in NBFM Receive block. Tau **is t**he Pre-emphasis time constant and the default value is 75e-5 (float). Max Deviation **is t**he maximum frequency deviation in Hertz and the default value is 75e-3 (float). We also use the multiplication factor to improve the amplitude of the signal.

Again we use throttle here as the signal has been transmitted and received, so there must be lot of disturbances in the signal, which is reduced by the throttle. This signal is again fed to QT GUI time and frequency sink.

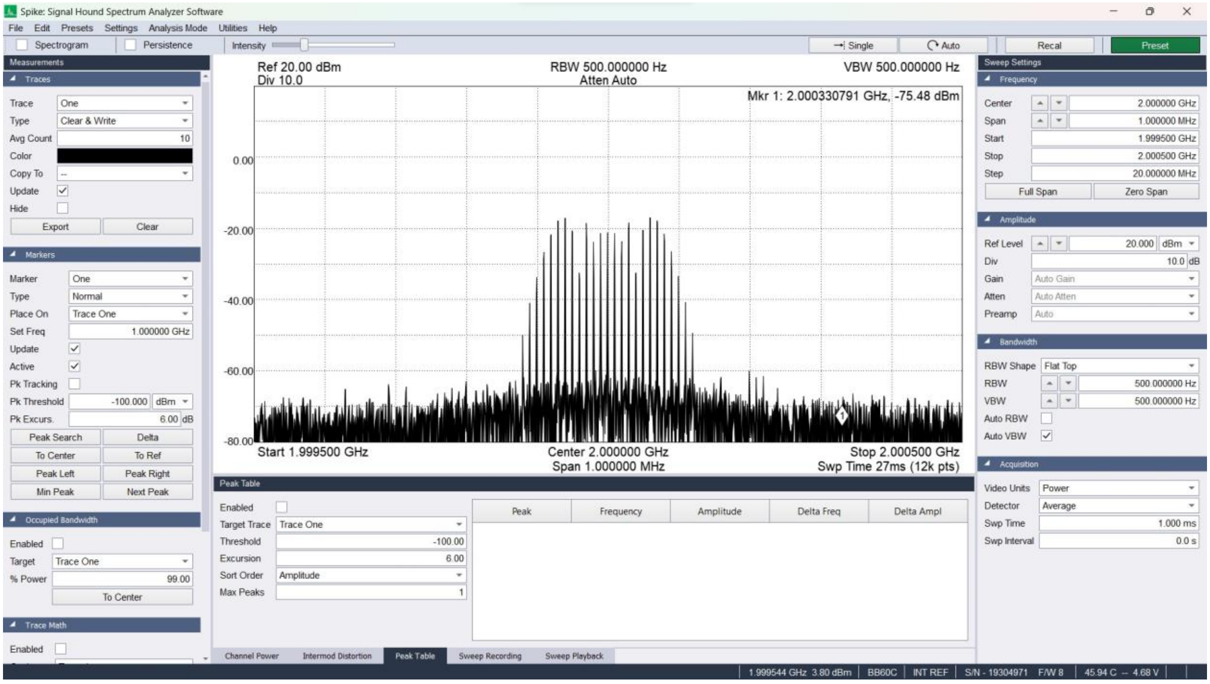
This produced signal is fed to QT GUI time and frequency sink with no of points 1024 which depicts the spacing and clarity of the output. Center frequency of the sink is given at 0khz and sample rate is 200Khz.



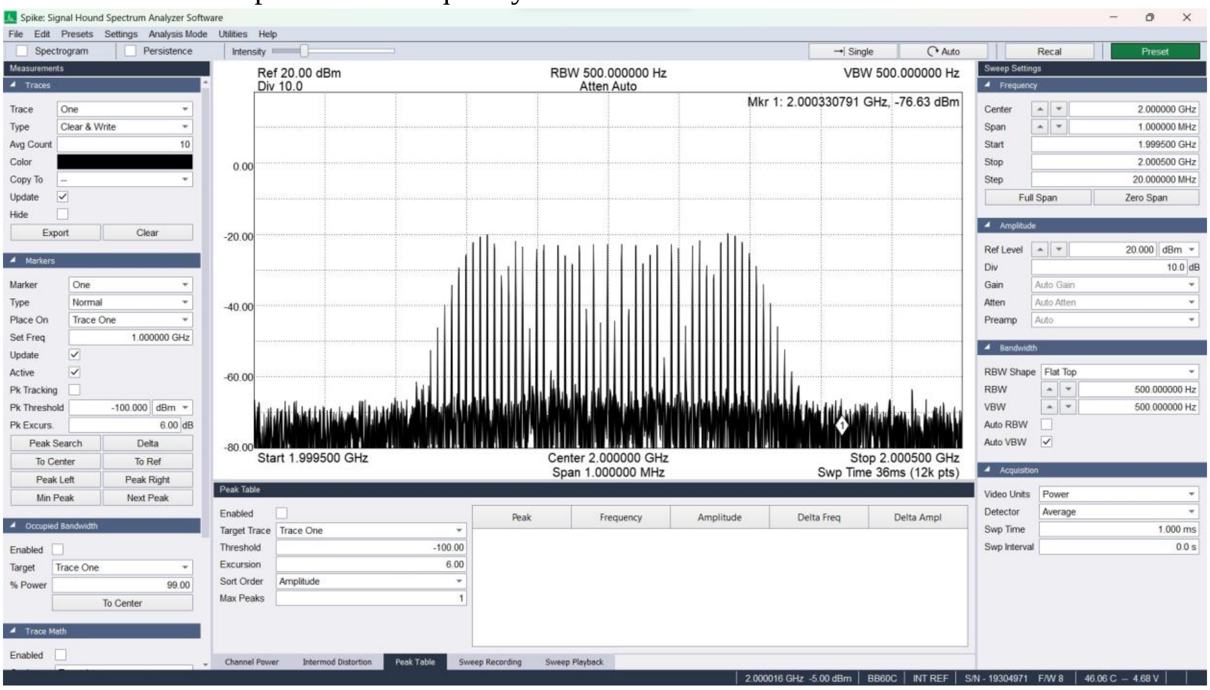
The 1st output is the channel output of is QT GUI time sink. It shows us the time demodulated NBFM signal (which is the message signal with a diminished amplitude due to amplitude factor).

The 2nd output is the power spectral density of the demodulated signal which is collected at the amitec source to a QT GUI frequency sink. This output is just a representation of message signal having its peak at 10khz, which matches with the existing theory.

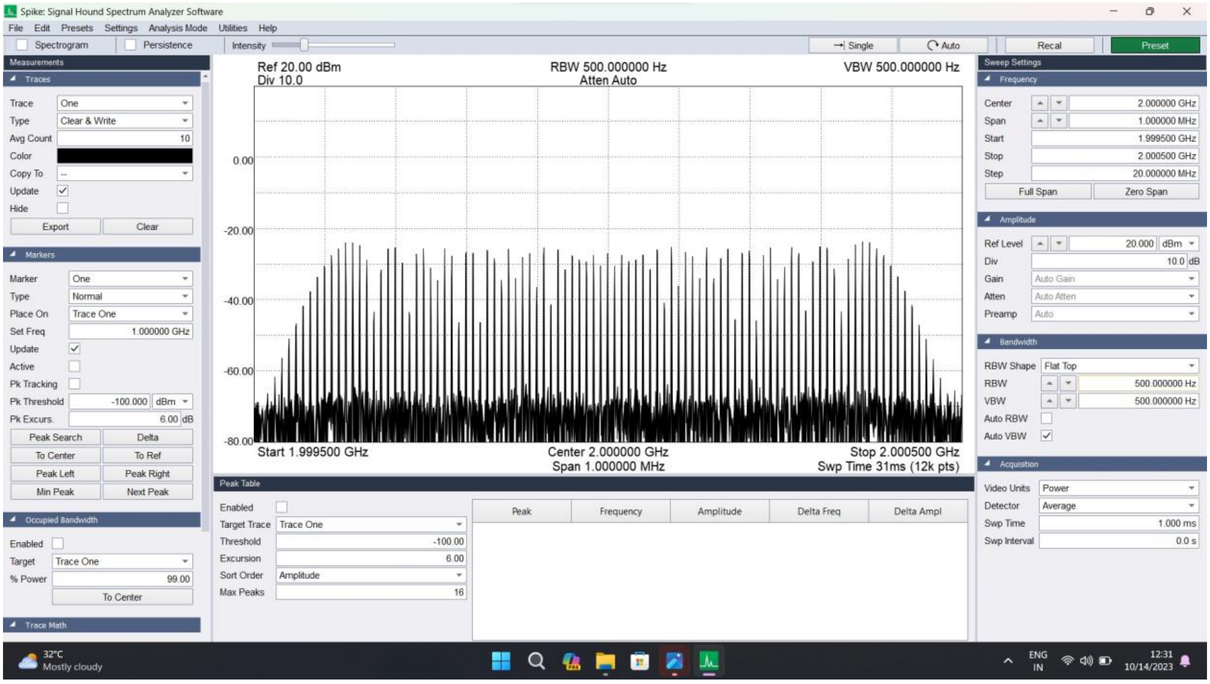
F) REAL TIME NBFM AND WBFM MODULATED SIGNAL FROM THE SPECTRUM ANALYSER:



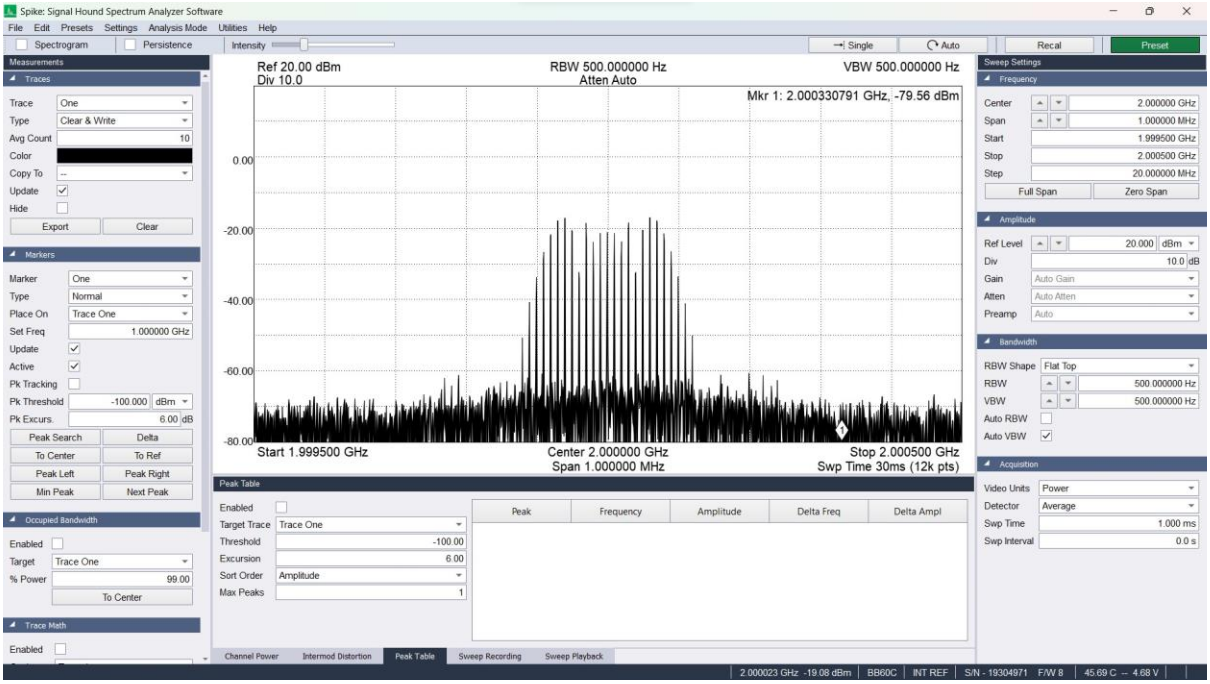
This is the NBFM spectrum at frequency deviation of 10khz



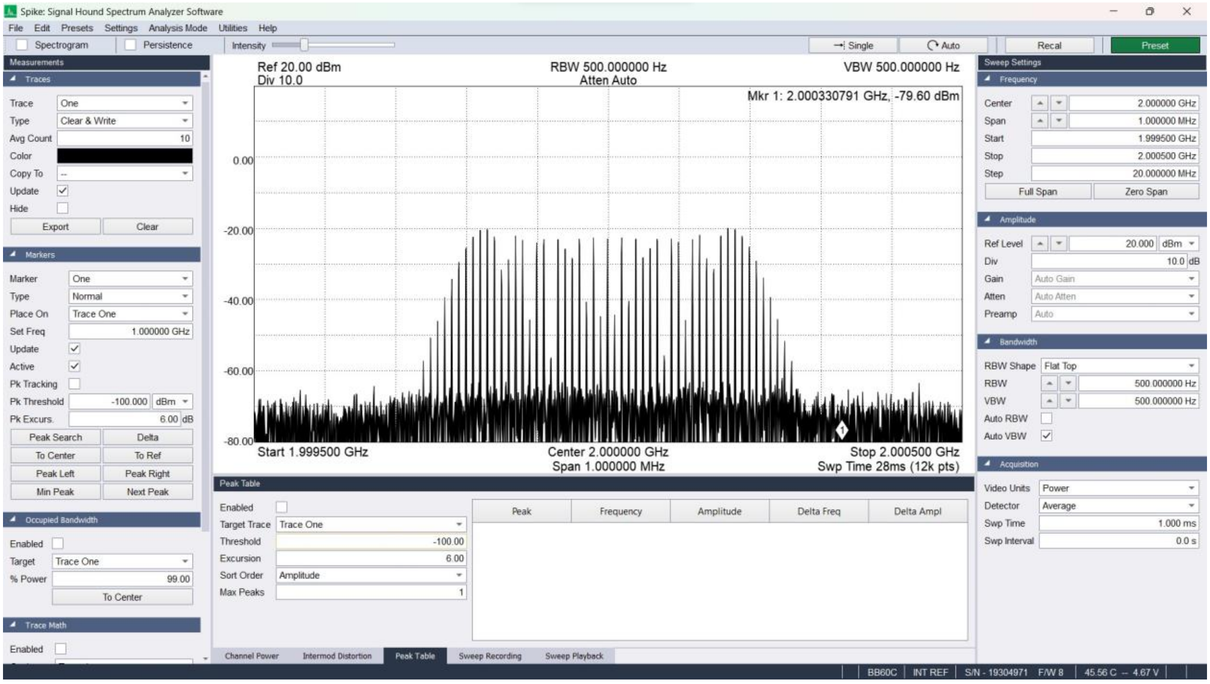
This is the NBFM spectrum at frequency deviation of 25khz



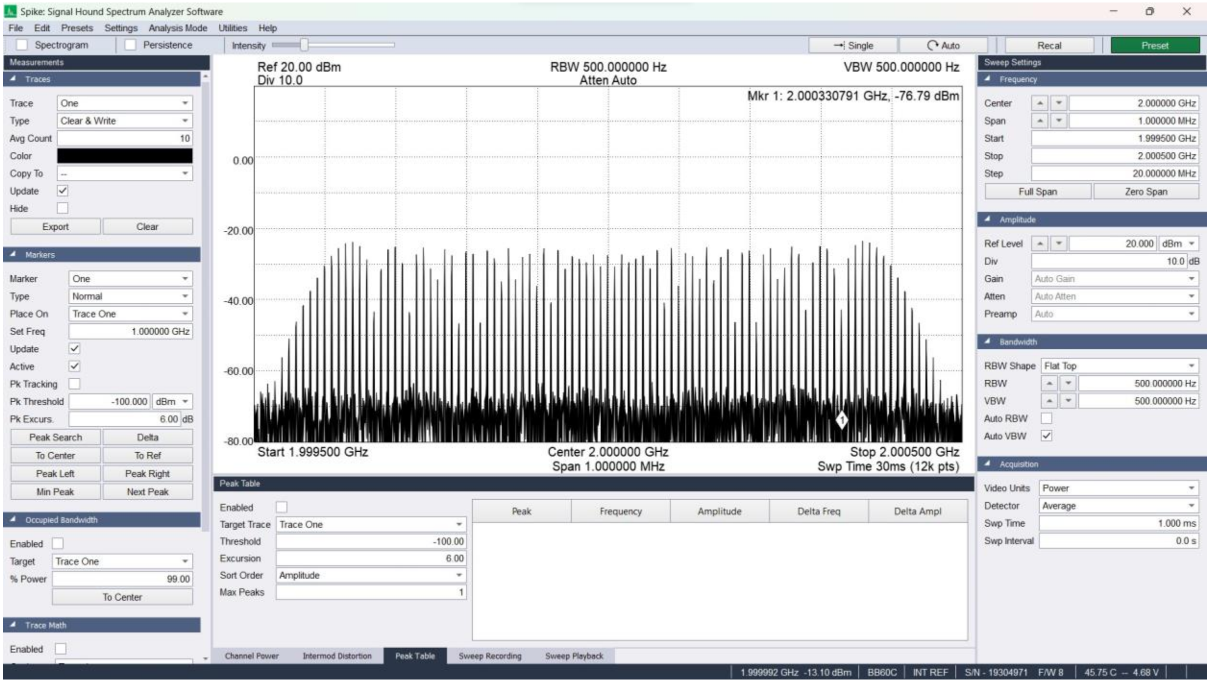
This is the NBFM spectrum at frequency deviation of 50khz



This is the WBFM spectrum at frequency deviation of 10khz



This is the WBFM spectrum at frequency deviation of 25khz



This is the WBFM spectrum at frequency deviation of 50khz