EECS 307: Lab Handout 2
(FALL 2012)

I- "Audio" Transmission of a Single Tone

In this part you will modulate a low-frequency audio tone via AM, and transmit it with a carrier also in the audio range. The audio signal will be transmitted from a speaker and will be received with a microphone. You will program the transmitter (connected to one PC) to generate the AM signal, and the receiver (connected to another PC) to demodulate the signal. To ensure that the received signal is sufficiently strong to demodulate, you will need to bring the microphone close to the speaker.

Illustrate the modulation and demodulation steps in the frequency domain, assuming that the message is a single tone at 500 kHz, and the carrier fc = 2 kHz.

Build the AM transmitter:

1. Open a new terminal and type: **gnuradio-companion**. This will launch GNU radio companion.

2. Start a new file in GNU Radio Companion (GRC). Double click on the **Variable** block to set the sampling rate to **48 kHz**. The variable name is **samp_rate**.

3. Create a source signal block by selecting Signal **Source** listed under [**Sources**] in the Blocks sidebar on the right and dragging it onto the workspace. Double click the block to edit its properties. Change its frequency to **500 Hz**. This is within the audible frequency range. Change its amplitude to **0.5**. In addition, change the output type from **Complex** to **Float** and hit OK.

4. Add a DC component to the original signal by creating two blocks: an **Add** block (under **Operators**) and a **Constant Source** block (under **Sources)**. Change the type of both blocks to **Float**, and set the constant for the constant source to **1**. Connect the constant and original sources to the **Add** block inputs.

5. Create another signal source block, then create a **Multiply** block (under **Operators**) and set their output type to **Float**.

6. Modify the new signal source to have a frequency of 2 kHz (audible) and connect its output to an input of the modulation multiply block. This will be the carrier. Also connect the output of the adder block to the other input of the modulation multiply block.

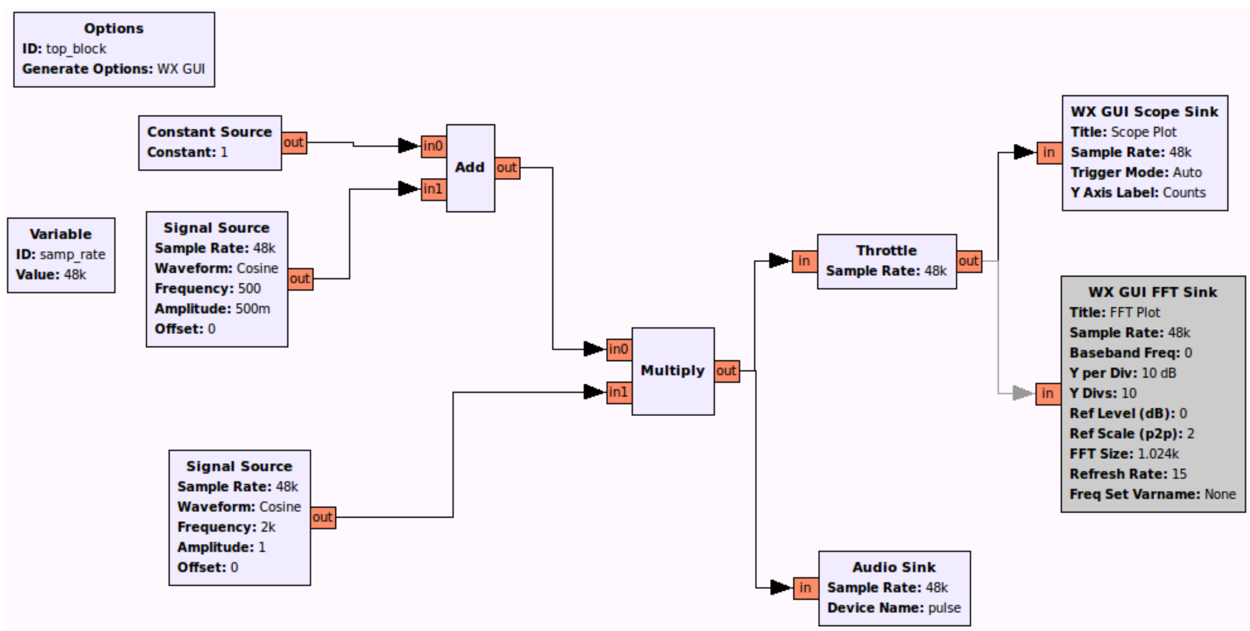
7. To view the signal, first attach the output of the multiply block to a **Throttle** (in the **Misc** category). Create a **WX GUI Scope sink** and **WX GUI FFT sink** block (under **WX GUI**

Widgets). In the properties of both of these blocks, change the type to **Float**. Also notice that the sampling rate is set to **samp_rate.** Connect the output of the **Throttle** to both the scope sink and FFT sink.

8. Save your file and execute it. Make sure you have no errors and that the time- and frequency-domain plots show the AM wave of interest. To include those plots in your report, click ALT+PrtScn to take a snapshot of the graphs.

9. To send the modulated signal to the speakers (physical transmitter), add an **Audio Sink** block (under **Sinks**) and connect it to the modulation multiply block. This will play the generated signal through the speakers. Double click on the **Audio Sink** and make sure its **Sample Rate** is set to **samp_rate** and its **Device Name** is set to **Pulse**. Your block diagram should look similar to the figure below.

10. Save the file and run it *with the speaker volume turned low*. You will hear a high frequency audio signal. Once you verify the transmitter is working, stop running the file.



Build the AM receiver:

1. On a different computer (connected to a microphone), start GNU Radio companion and create a new file. Set the sampling rate to 48 kHz and the variable name to **samp_rate**

2. Create an **Audio Source block** (under **Sources**).

3. Create a **Band Pass Filter** (under **Filters**). Set its type to **Float → Float (Decim).** The sample rate should be set to **samp_rate**. Set the low cutoff frequency to **1.2 kHz**, the

high cutoff frequency to **2.8 kHz**, and the transition band to **100 Hz**. The purpose of this filter is to eliminate other signals and noise outside the frequency range of the desired signal. **What is the bandwidth of the modulated signal?**

4. To view the signal, first attach the output of the bandpass filter to a **Throttle** (in the **Misc** category). Create a **WX GUI Scope sink** and **WX GUI FFT sink** block (under **WX GUI Widgets**). In the properties of both of these blocks, change the type to **Float**. Also ensure that the sampling rate is set to **samp_rate.** Connect the output of the throttle to both the scope sink and FFT sink.

5. Save the receiver file and execute it. Run the transmitter on the other computer and place the microphone next to the speaker. At the receiver, check the time- and frequency-domain plots to make sure that what you are receiving is actually the modulated signal. For the time signal, unclick **Auto Range** and adjust the time-scale and magnitude so that your plots show the modulated signal. For the frequency-domain signal, click **Average** to see the frequencies of interest. Save your plots.

PLEASE turn off the volume or the transmitter whenever you are not using them.

6. Once you have verified that the modulated signal is received by the microphone/receiver, connect the output of the bandpass filter to an envelope detector (similar to the one constructed in Lab 1) to obtain the original signal.

7. Create a **Null** source block (under **Sources**) and a **Max** block (under **Operators**). Change the type of the **Null** source and **Max** to **Float**.

8. Connect the **Null** source to one of the inputs of the **Max** block, and the output of the band pass filter block to the other input. This will serve as a diode, since if the input is positive, it remains unchanged at the output, and otherwise, the output is zero.

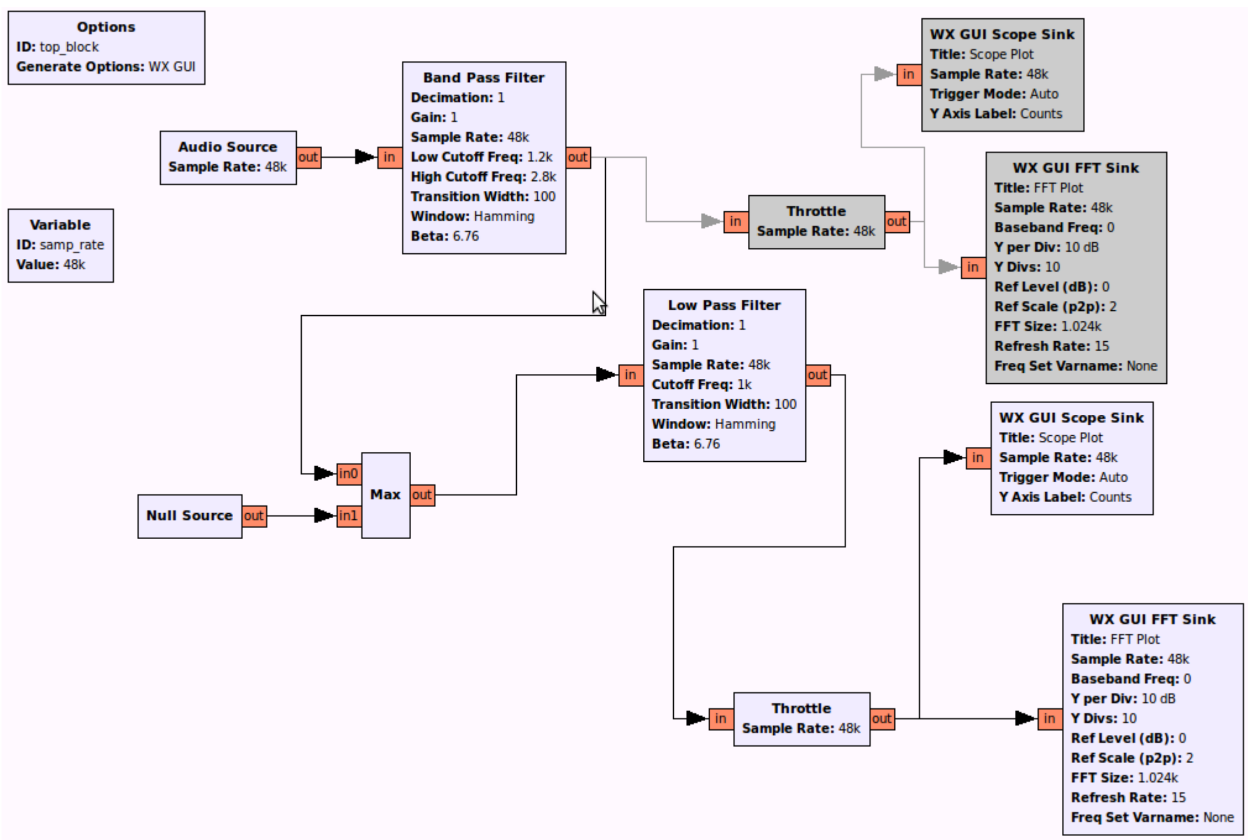
9. Add a **Low Pass Filter** block (under **Filters**). Open the low pass filter properties. Change the FIR Type of the filter to **Float → Float (Decimating)**, then set the cutoff frequency to **1 kHz** and the transition width to **100 Hz**.

10. To view the signal, first attach the output of the low pass filter to a **Throttle** (in the **Misc** category). Create a **WX GUI Scope sink** and **WX GUI FFT sink** block (under **WX GUI Widgets**). In the properties of both of these blocks, change the type to **Float**. Also notice that the sampling rate is set to **samp_rate.** Connect the output of the throttle to both the scope sink and FFT sink.

11. Disable (by selecting and hitting “d”) the **Throttle** block, scope sink, and FFT sink for the modulated signal (the blocks created in step 4). Your block diagram should look

similar to the figure below.

12. Save your file and run it. Now run the transmitter. In the spectrum of the received demodulated signal, you will see the frequency of interest and a DC component. The latter can be filtered out by subtracting the average of the signal or by using a bandpass filter. The original signal is then recovered. Save the plots.



II- Transmitting a Single-tone Signal through USRP1

In this section you will use the USRP1 along with its daughterboard (RF front end) to transmit and receive signals using electromagnetic waves. Each group will receive two USRPs. One has a Basic TX daughterboard, which you will connect to a PC and use as a transmitter. The other comes with a Basic RX daughterboard, which you will connect to a neighboring computer and use as a receiver. The two USRPs should be brought close together since the transmitted power is low.

Build the transmitter:

1. After connecting the transmitting USRP, start a new file in GNU Radio Companion (GRC). Double click on the Variable block to set the sampling rate to 1 MHz where the variable name is **samp_rate**.

2. Create a source signal block by selecting **Signal Source** listed under [**Sources**] in the Blocks sidebar on the right and dragging it onto the workspace. Double click the block to edit its properties. Change its frequency to **100 kHz** and its amplitude to **0.5**. In addition, change the output type from **Complex** to **Float** and hit OK.

   Note: In the previous section, we modulated the message by creating a carrier signal and multiplying it with the message. In this part, the message is modulated within the USRP hardware.

3. The USRP uses the complex baseband representation of signals (with in-phase and quadrature phase components). To represent our signal as a complex baseband signal add a **Float to Complex** block (under **Type Conversions**). Also add a **Null** source (under **Sources**) and change its output type to **Float**. Connect the output of the signal block to the real input of the **Float to Complex** block. Connect the output of the **Null** source to the imaginary input of the **Float to Complex** block. This way we create a complex number with a zero-valued imaginary part. **What is the in-phase component of this signal? What is its quadrature component?**

4. To view the signal, first attach the output of the **Float to Complex** block to a **Throttle (**in the **Misc category).** Create a **WX GUI Scope sink** and **WX GUI FFT sink** block **(**under **WX GUI Widgets).** In the properties of both of these blocks, change the type to **Complex**. Also notice that the sampling rate is set to **samp_rate.** Connect the output of the **Throttle** to both the scope sink and FFT sink.

5. Save your file. Run it and verify that the signal is not yet modulated. Save your plot.

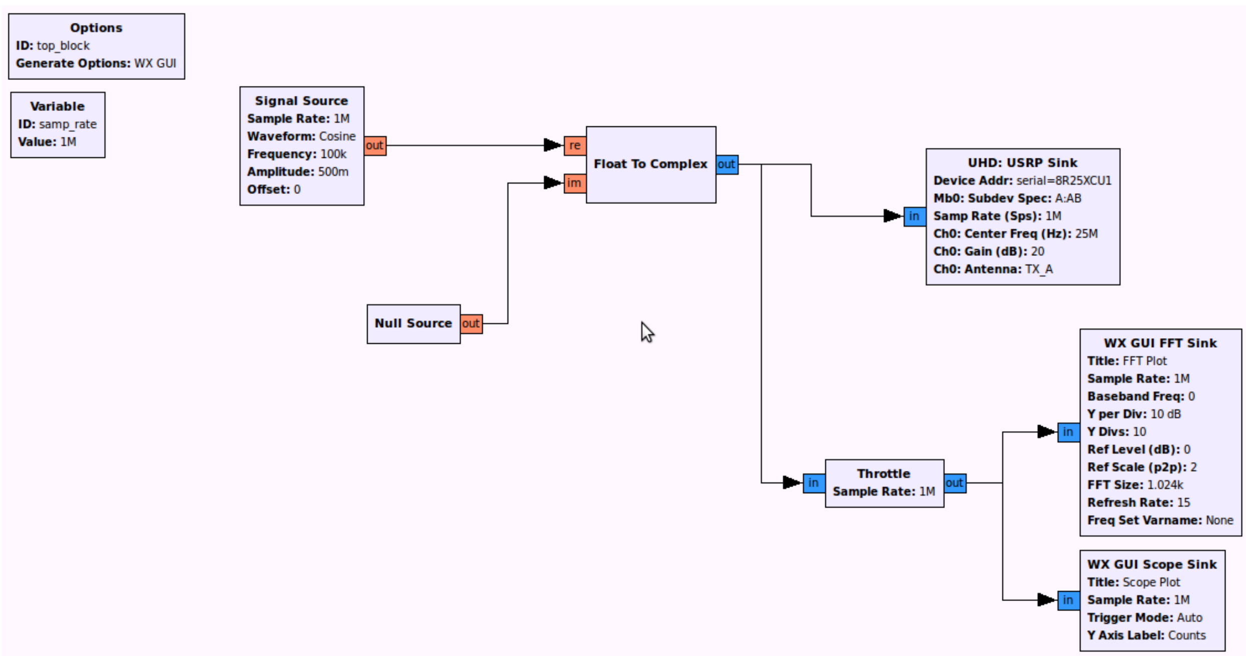
6. Add a **UHD: USRP Sink** block (under **UHD**). This will send your message to the USRP board. Double click on the **UHD: USRP Sink** block to edit its properties.

7. To get the device address, open a terminal and type **uhd_find_devices.** Use the serial number that you get as the device address, for example **serial = 8R25XCU1**.

8. The number of motherboards **Num Mboards** should be set to 1 and the **Mb0: Subdev Spec** should be set to **A:AB**. There are two types of daughterboard slots on your USRP board, slot A and slot B. The first **A** in **A:AB** denotes which slot you are using. Make sure your daughterboard is connected to slot **A**.

9. Set your **Sample Rate** to **samp_rate**. Set the **Center Freq** to **25 MHz**. This is how you specify the carrier frequency at which you modulate your signal. Set the **Gain** to 20 dB. Set your **Antenna** to **TX_A**.

10. Your block diagram should be similar to the figure below. Save your file and run just to make sure there are no errors. Once you have verified that the transmitter is working, stop the file.



Build the Receiver (very simple):

1. After connecting the receiving USRP to a neighboring computer, start a new file in GNU Radio Companion (GRC). Double click on the Variable block to set the sampling rate to **1 MHz** and variable name is **samp_rate**.

2. To capture the transmitted signal using USRP, add a **UHD: USRP Source** block (under **UHD**). Double click on the **UHD: USRP Sink** block to edit its properties.

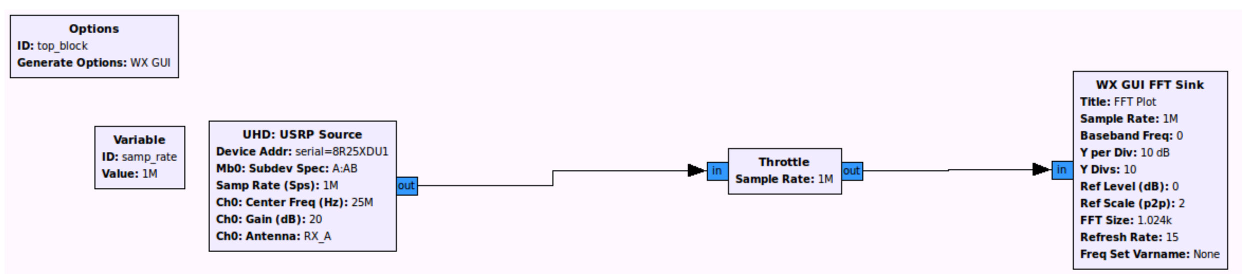
3. To get the device address, open a terminal and type **uhd_find_devices**. Use the serial number that you get as the device address, for example **serial = 8R25XCU1**.

4. The number of motherboards **Num Mboards** should be set to 1 and the **Mb0: Subdev Spec** should be set to **A:AB**. There are two types of daughterboard slots on your USRP board, slot A and slot B. The first **A** in **A:AB** denotes which slot you are using. Make sure your daughterboard is connected to slot **A**.

5. Set your **Sample Rate** to **samp_rate** and the **Center Freq** to **25 MHz**. This is how you specify the carrier frequency at which your signal is modulated. Set the **Gain** to 20dB. Set your **Antenna** to **RX_A**.

6. To view the signal, first attach the output of the **Float to Complex** block to a **Throttle (**in the **Misc** category**).** Create a **WX GUI FFT sink** block **(**under **WX GUI Widgets).** In the properties of both of these blocks, change the type to **Complex**. Also check that the sampling rate is set to **samp_rate.** Connect the output of the **Throttle** to the FFT sink.

7. Your block diagram should be similar to the figure below. Save your file and run it. In the FFT graph, set **Average** to **ON** in **Trace Options**. Now turn on the transmitter and start transmitting your 100 kHz signal. Verify that it is the received frequency. Save your plot.



III- Transmitting a Dual-Tone Signal through USRP1

In this section, you will use the USRP1 to transmit a dual-tone signal composed of the sum of two audio tones at different frequencies. Unlike what we did in Part II where we used the in-phase component only, we use Single Side Band Modulation, and therefore will have to use both the in-phase and quadrature components.

We will only modify the transmitter. The receiver will not change in this part.

Double Sideband Modulation (DSB)

We first transmit the signal using DSB. **Is the baseband envelope complex or real? Is the baseband envelope of a single side-band signal complex or real?**

1. Copy the transmitter that you built in part II to a new file. Delete the connection between

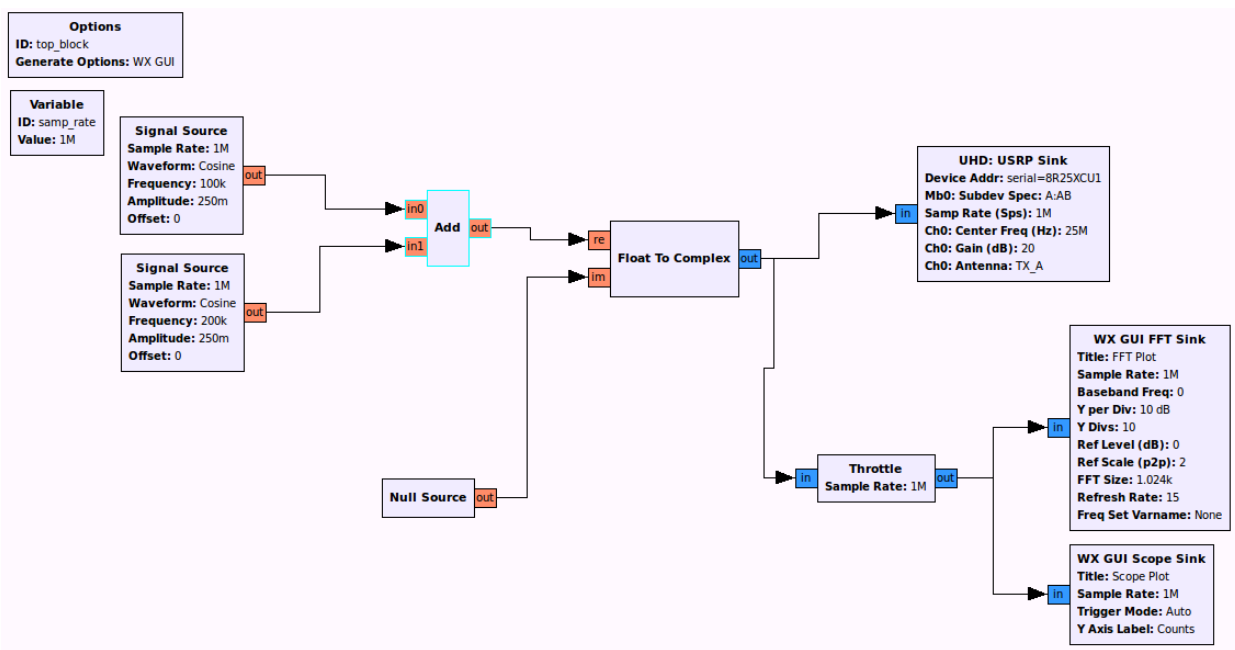
the **Signal Source** block and the **Float to Complex** block. Double click on the **Signal Source** block and set its amplitude to **0.25**. Keep the frequency equal to **100 kHz**

2. Create another source signal block (under **Sources**). Double click the block to edit its properties. Change its frequency to **200 kHz** and its amplitude **to 0.25**. In addition, change the output type from **Complex** to **Float** and hit OK.

3. Create an **Add** block (under **Operators**). Change its output type to **Float**. Connect the 100 kHz signal block to the first input of the adder and the 200 kHz signal block to the second input of the adder. This generates the dual-tone message signal. Connect the output of the **Add** block to the **Float to Complex** block. Everything else remains the same. Your block diagram should look like the figure below.

4. **What is the complex baseband envelope of the transmitted signal? Is the modulated signal Double Sideband or Single Sideband? Explain.**

5. Run the transmitter and receiver. Verify your answer by looking at the FFT sink at both the transmitter and receiver. Save your plots.



Single Sideband (SSB) Modulation

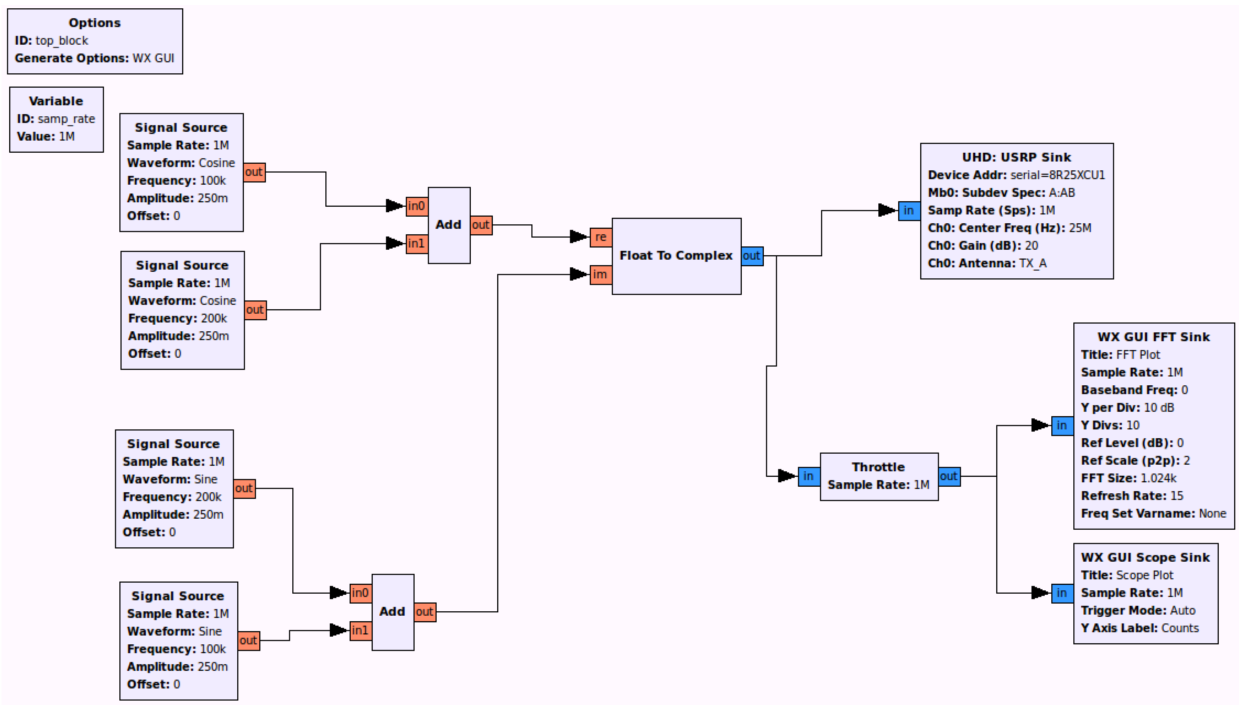
Again, the receiver will not change in this part. We will only modify the transmitter to transmit a Single Sideband signal. **Given the message** $m(t) = cos(2\pi.100k.t) + cos(2\pi.200k.t)$ **which modulates a carrier** $c(t) = \cos(2\pi.25M.t)$, **what is the complex envelope of the SSB signal?**

1. Delete the **Null** source and create two signal blocks (under **Sources**). Double click to edit

its properties. Change **Waveform** to **Sine** wave and set the frequencies to **100 kHz** and **200 kHz**. Change the output type to **Float** and set the amplitude to **0.25**.

2. Create an **Add** block. Connect the two quadrature baseband signals to the inputs of the adder. Connect the output of the adder to the imaginary part of the **Float to Complex** block. This generates the complex envelope. Your block diagram should look similar to the figure below.

3. Now we expect to get a single-sideband signal. Run your transmitter and receiver and verify the SSB signal by looking at the FFT sink at both the transmitter and receiver. Save your plots.



IV- Lab Report

For the lab report, please answer all of the questions in this handout. Please also attach and explain all graphs.