

# EE3209 - Communications Laboratory

## Lab Manual

California State University, Los Angeles  
College of Engineering, Computer Science, and Technology  
Department of Electrical and Computer Engineering



**Instructor: Khashayar Olia**



# Lab 3: Frequency-Division Multiplexing

## Objective:

In this laboratory exercise you will investigate sending multiple messages on a single carrier by Frequency-Division Multiplexing (FDM). Each individual message is modulated onto a separate subcarrier then the modulated subcarriers are summed before being sent to the USRP. At the receiver, the individual messages are separated by filtering, and then demodulation process is applied. The purpose of this exercise is to:

- Provide additional practice in programming the USRP
- Introduce the concept of Frequency-Division Multiplexing (FDM)
- Explore the concept of intermediate-frequency filtering in the receiver

## Background:

One of the important processing operation in analog communication is multiplexing where a number of distinct signals are combined into a single composite signal to transmit over the channel. The multiplexing technique is accomplished by separating the signals in either frequency or time. Here we discuss and experiment the multiplexing in frequency which is known as Frequency-Division Multiplexing (FDM). In FDM, the channel bandwidth is divided into numbers of non-overlapping adjacent frequency slots and each message signal is assigned a frequency slot, this can happen when the bandwidth of the communication medium is greater than individual signals to be transmitted through the channel. So FDM transmits data in parallel over narrow frequency slices compared to the overall bandwidth and applies linear modulation in parallel to a number of subcarriers. FDM is widely used in telemetry, TV signal transmission, and, before the fiber optics, it was the standard transmission method for long-distance telephone signals [1]. The orthogonal version of FDM technique (OFDM) in which the subcarriers are orthogonal to others is used in DSL and in fourth-generation (4G) cellular telephone systems.

A simple Block Diagram of an FDM system is shown in Figure 1 where the incoming messages are passed into a Low Pass filter (LPF) to remove high frequency components then the modulation is applied to the filtered signals to shift the frequency ranges of the signals so as to occupy mutually exclusive frequency



intervals. The Band Pass Filters (BPFs) are used after the modulators to restrict the band of each modulated wave to its prescribed range. The resulting outputs are next combined in parallel to form the input to the communication channel.

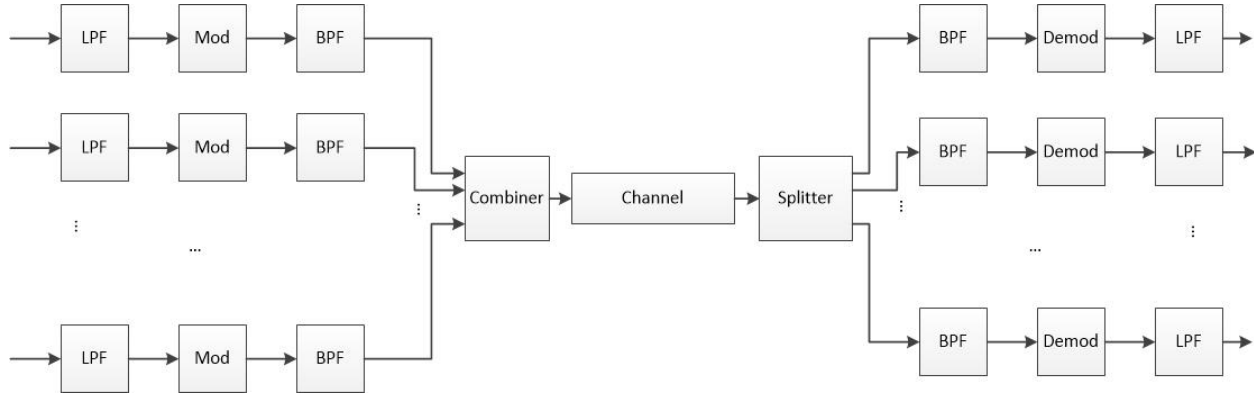


Figure 1: FDM Architecture

Figure 2 shows the message combiner results in frequency domain:

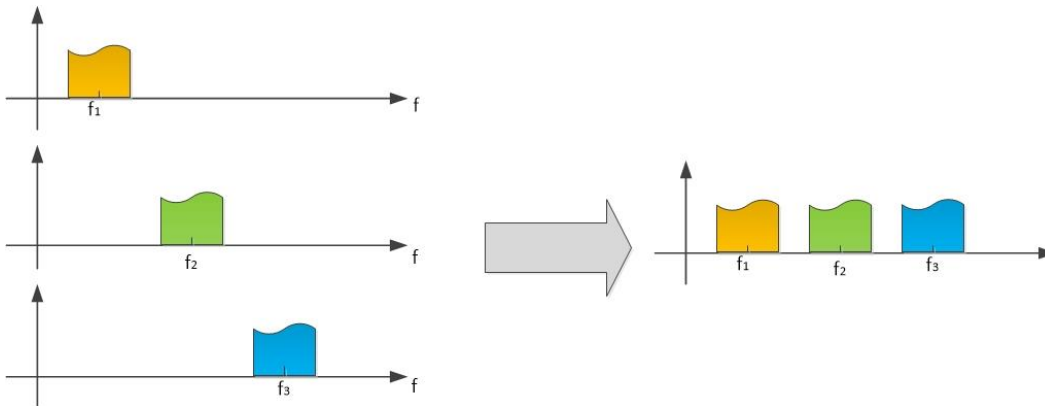


Figure 2: Combining the Messages in Frequency Domain

At the receiver, the BPFs are used to separate the message signals on a frequency-occupancy basis then the demodulators reconstruct the signals. The demodulated signals are passed into a LPF to finally, the original message signals are being recovered.

A simple form of FDM signal can be written as Eq. 1:

$$s(t) = \begin{cases} \sum_{i=0}^{N_s-1} m_i(t) [\cos(2\pi f_i(t - t_s))] & t_s \leq t \leq t_s + T \\ 0 & \text{Otherwise} \end{cases}$$

Eq. 1



Where it starts at  $t = t_s$ ,  $m_i(t)$  is the message signal,  $N_s$  is the number of subcarriers,  $T$  is the symbol duration, and  $f_i = f_0 + \frac{i}{T}$  is the carrier frequency.

Figure 3 shows time domain representation of three sub-carriers from a FDM signal in which all sub-carriers have the same phase and amplitude, however in practice the amplitudes and phases may be modulated differently for each sub-carrier.

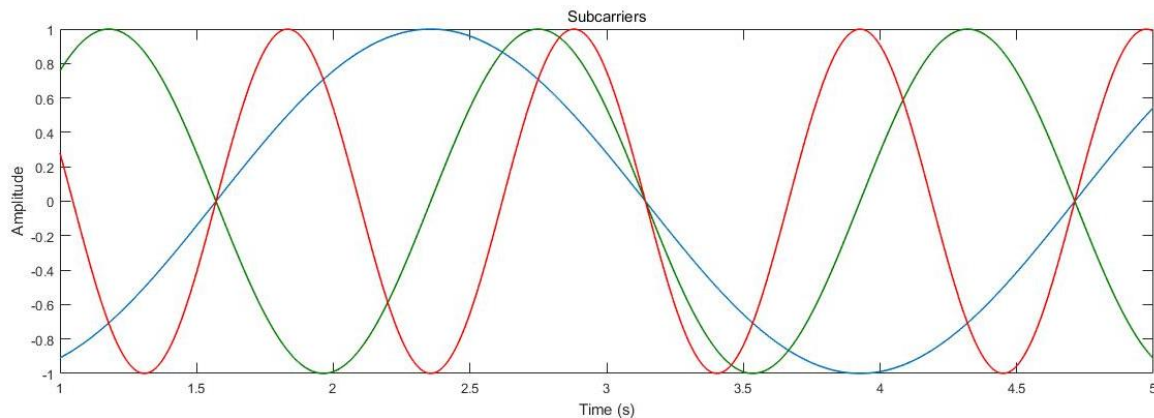


Figure 3: Example of FDM subcarriers

Figure 4 shows the sum of subcarriers as the final Baseband (BB) message:

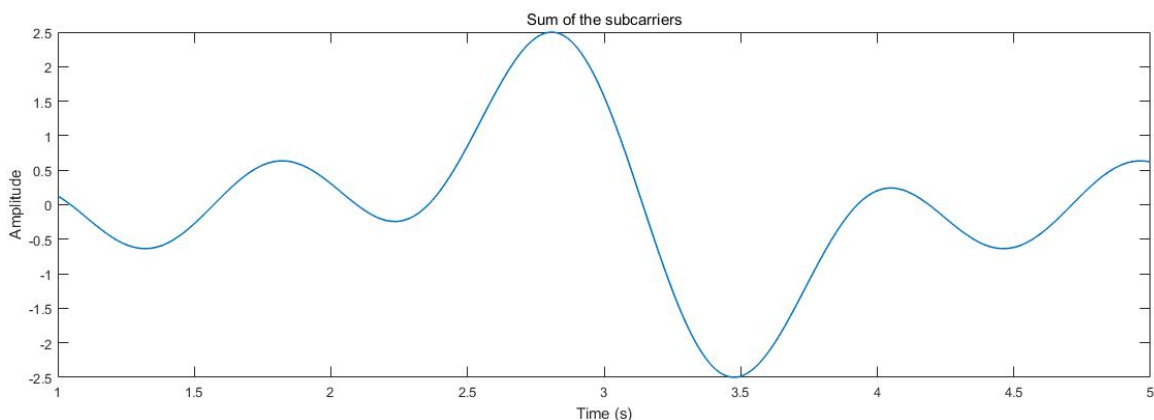


Figure 4: Final FDM signal

In this lab, we use FDM and Amplitude Modulation (AM) technique to modulate two signals. It is not necessary to use AM to modulate the subcarriers but we are using AM in this lab exercise because it is familiar from previous lab, and also because it is easy to demodulate. Suppose  $m_1(t)$  and  $m_2(t)$  are



message signals, and let  $f_1(t)$  and  $f_2(t)$  be corresponding subcarrier frequencies so we can form the modulated subcarrier signals:

$$s_1(t) = A_1 \left[ 1 + \mu_1 \frac{m_1(t)}{m_{1p}(t)} \right] \cos(2\pi f_1 t)$$

$$s_2(t) = A_2 \left[ 1 + \mu_2 \frac{m_2(t)}{m_{2p}(t)} \right] \cos(2\pi f_2 t)$$

Eq. 2

Where the  $m_{ip}$  and  $\mu_i$  are maximum value of the signals and modulation indices respectively. For simplicity, we assume the phase is zero ( $\varphi = 0$ ).

Note that the subcarrier frequencies  $f_1(t)$  and  $f_2(t)$  must be spaced sufficiently apart in frequency so that the spectra of  $m_1(t)$  and  $m_2(t)$  do not overlap. For FDM, the modulated signals  $s_1(t)$  and  $s_2(t)$  are combined and the result is the In-Phase (Real) component of the Baseband signal:

$$s_I(t) = s_1(t) + s_2(t)$$

Eq. 3

As in Lab 2, we will let the Quadrature (Imaginary) component,  $s_Q(t)$  equal zero, so that the signal sent to the USRP is:

$$\tilde{s}(t) = s_I(t) + js_Q(t) = s_I(t) + j \underbrace{[0]}_{s_Q(t)} = \underbrace{s_1(t) + s_2(t)}_{s_I(t)}$$

Eq. 4

The USRP transmitted signal is illustrated by Eq. 5:

$$\begin{aligned} \tilde{s}(t) &= As_I(t) \cos(2\pi f_c t) - jAs_Q(t) \sin(2\pi f_c t) \Rightarrow \\ &= A \underbrace{[s_1(t) + s_2(t)]}_{s_I(t)} \cos(2\pi f_c t) - jA \underbrace{[0]}_{s_Q(t)} \cos(2\pi f_c t) \Rightarrow \\ \tilde{s}(t) &= A[s_1(t) + s_2(t)] \cos(2\pi f_c t) \end{aligned}$$

Eq. 5

where A is set by the **carrier level** parameter and  $f_c$  is the USRP **carrier frequency**. Note that  $s_1(t)$  and  $s_2(t)$  are sinusoid signals.

On the receiving side, the USRP receiver generates the output as follow:



$$\tilde{r}(t) = A_r[s_1(t) + s_2(t)]e^{j\theta} \Rightarrow$$

$$\tilde{r}(t) = A_r[s_1(t) + s_2(t)]\cos(2\pi f_c t + \theta) + jA_r[s_1(t) + s_2(t)]\sin(2\pi f_c t + \theta)$$

Eq. 6

where  $\theta$  is the phase difference between the transmitter and receiver oscillator signals. This complex-valued signal can be sent to a bank of two bandpass filters centered on the subcarrier frequencies  $f_1$  and  $f_2$ . The filter outputs are the individual signals given in Eq. 2. These can now be demodulated using envelope detectors as in Lab 2. Figure 5 and Figure 6 shows the high level FDM architecture used in this lab experiment:

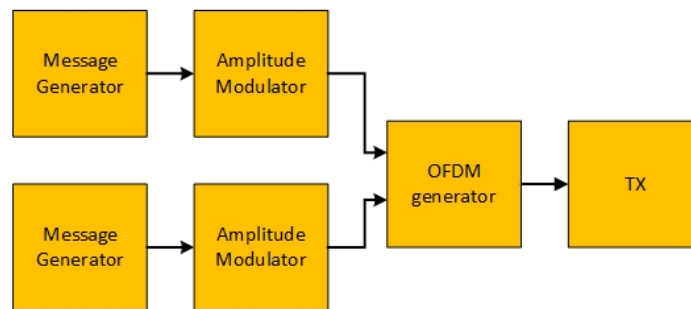


Figure 5: OFDM TX Architecture

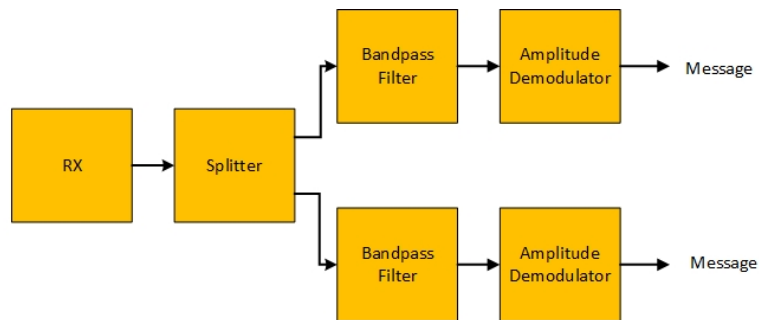


Figure 6: OFDM RX Architecture

## Pre-Lab:

**LabVIEW Note:** In order to be able to run all the VIs and use all the LabVIEW icons, run LabVIEW 2015 (32bit) version for this lab.

Templates for transmitter and receiver have been provided in the file **Lab3TxTemplate.vit** and **Lab3RXTemplate.vit**. The task is to add blocks as needed to produce FDM signal composed of two AM messages, pass it into the transmitter, and then reconstruct the messages at the receiver by using BPF and envelope detector (amplitude demodulation).

## Setting up the USRP

### Transmitter:

To complete the transmitter you will be asked to perform two tasks:

- A. Create a sub-vi that modulates a message  $s_i(t)$  using Amplitude Modulation.
- B. Update the transmitter template to combine the modulated messages to form the FDM signal.

Hierarchical nature of LabVIEW makes it a suitable and powerful software for complex programs. LabVIEW provides the opportunity to create a VI and then use it indefinite time on Block Diagrams of other VIs. This feature is called modularity which helps to debug and manage changes in program very quickly. Modularity offers the capability of creating smaller sections of code known as sub-VIs, which have Front Panels and Block Diagrams, and call them within another VI [2]. Sub-VIs are similar to subroutine, function and method in text-based programming languages. Here in this lab, we use this very helpful feature of LabVIEW to create our own AM subprogram, then use this subprogram in the main transmitter program. To do this task, follow the following steps to create your AM transmitter. The first task is to create the subprogram and the second task is to update the transmitter:

- A. Creating a sub-vi **AMonSubcarrier.vi** to generate the signals  $s_1(t)$  and  $s_2(t)$  as shown in Eq. 2.

The template for the sub-vi is provided in the file **AMonSubcarrierTemp.vit**, use it as the starting point:

- I. In order to generate envelope of a signal, we used **MathScript Node** in Lab 2. Use the



AMonSubcarrierTemp.vit and place a MathScript Node in the Block Diagram (Mathematics > Script & Formulas > MathScript and add a MathScript Node). The MathScript Node needs the following four inputs to generate the Baseband signal output:

1.  $m(t)$ : the message waveform data values
2.  $m_p(t)$ : the maximum of the message waveform data values
3.  $\mu$ : modulation index
4.  $A$ : carrier level

So, define these four inputs.

- II. Unlike Lab 2, you need to have a real signal at this step and the imaginary part of the signal will be generated later at the input of the transmitter. In addition, there is no need to scale down the signal due to constraint imposed by DAC in USRP because we will use “1” as amplitude. So, simply construct the envelope illustrated in Eq. 7 using the four inputs from previous step inside the MathScript Node:

$$E = A \left[ 1 + \mu \frac{m(t)}{m_p(t)} \right]$$

Eq. 7

- III. The message signals are provided from the transmitter VI through message generators, and are transferred to this sub-vi as cluster. In order to get the data values of the generated signal  $\{m_i\}$  from the cluster-type input, use **Get Waveform Components** icon in your sub-vi (Functions > Programming > Waveform > Get Wfm Comps).
- IV. Make proper wire connection between **Message waveform in** cluster input and Get Waveform Components from previous step.
- V. Use the second output of the **Quick Scale** block (Functions > Signal Processing > Signal Operation > Quick Scale) to calculate the maximum absolute value of the generated signals  $\{m_p\}$ .





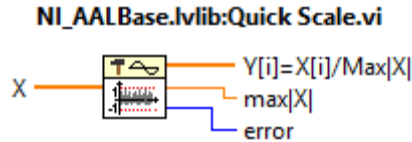


Figure 7: Quick Scale

- VI. Go to Front Panel and use two **Numeric Controls** (Controls > Modern > Numeric > Numeric Control) for the carrier level and modulation index and name them appropriately.
- VII. Go back to the Block Diagram. In order to increase the readability of your code, right click on the Numeric Control icons you just added and unselect the **View As Icon** option.

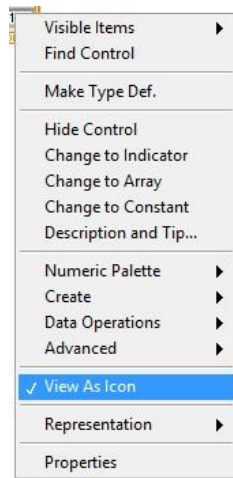


Figure 8: View As Icon option

- VIII. Wire  $\{m_i\}$ ,  $\{m_p\}$ ,  $\mu$  and  $A$  properly.
- IX. Now, the envelope is ready and the signal  $s_1(t)$  needs to be generated. As Eq. 2 suggests, you need to multiply a cosine waveform to the output of the MathScript Node block to generate  $s_1(t)$ . You can use the **Sine Waveform** icon (Functions > Signal Processing > Waveform Generation > Sine Wfm).
- X. For sampling info input of the Sine Waveform icon, use **Sampling info for subcarrier** cluster. This cluster carries the sample frequency from transmitter.
- XI. Now, you need to define the subcarrier frequency  $f_c$ . Go to Front Panel and use Numeric Controls for subcarrier frequency and wire it to the Sine Waveform properly.



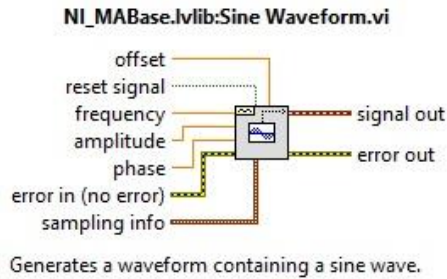


Figure 9: Sine Waveform Icon

- XII. Use  $90^\circ$  phase to convert it to the cosine block. To do so, you can use “Numeric Constant” or right click on the Sine Waveform block and select Create > Constant.

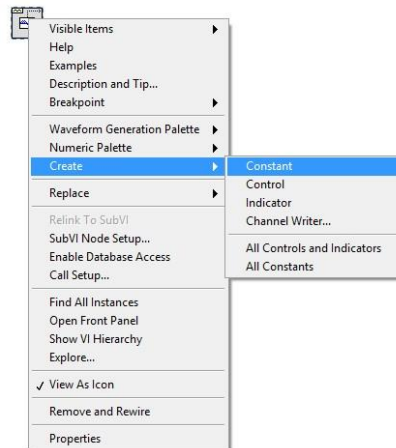


Figure 10: Create constant directly from the block

- XIII. Use another Get Waveform Components icon to get the waveform component of the cosine signal.
- XIV. In order to modulate the signal as Eq. 3, you need to multiply the cosine waveform by the output of MathScript Node, the Baseband signal. Use **Multiply** icon (Functions > Mathematics > Numeric > Multiply) for this purpose.
- XV. Go to the Front Panel and use a **Waveform Graph** (Controls > Modern > Graph > Waveform Graph) to show the modulated signal and call it **Output**. We’d like to have a numeric output for our sub-vi too. Since you’re working with an array and not a single element, so you need to have an array numeric output. In order to have an array numeric

output, use a **Numeric Indicator** (Controls > Modern > Numeric > Numeric Indicator) and **Array** (Controls > Modern > Array, Matrix and Cluster > Array) in the Front Panel. Then drag the Numeric Indicator object into the Array shell and call it Output. The array shell automatically resizes to accommodate the new object.

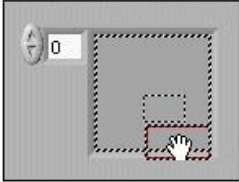


Figure 11: Generating Array Output

XVI. Toggle to the Block Diagram and wire both outputs properly. Now your sub-vi has the minimum required I/Os. The required inputs and outputs are summarized in Table 1.

Table 1

Inputs	Index	Descriptions
Message waveform	$m_i(t)$	waveform (double)
Modulation index	$\mu_i$	double
Carrier level	$A_i$	double
Sampling information for subcarrier	---	cluster
Subcarrier frequency	$f_i$	double
Output	Index	Descriptions
Modulated signal out	$g_i(t)$	array (double)

Once you define all the inputs and outputs, you need to make sure that the inputs and outputs of the sub-vi have corresponding controls and indicators, and are placed accordingly, then you need to assign ports to the inputs and outputs. We do this task through what is called the **Connector Pane**. LabVIEW provides default set of patterns that you can use to assign I/Os, so you just need to link the terminals to controls and indicators on the

Front Panel. To use the Connector Pane for your just created sub-vi, return to the Front



Panel and click on the icon located in the top-right corner of the window as illustrated in Figure 12.

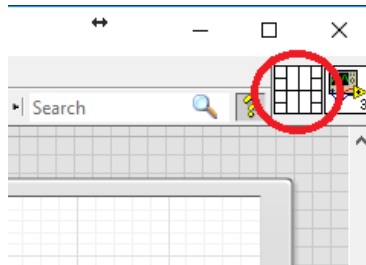


Figure 12: Connector Pane

In order to designate which terminals will take the inputs and which terminals will produce the outputs of the VI, hover your mouse over the Connector Pane icon where you can see the pointer turns into the **Wiring Tool**. Simply click on the terminal you want to assign to an input or output, and then click on the corresponding control or indicator on the Front Panel. Assign inputs and outputs to your Connector Pane as shown in Figure 13.

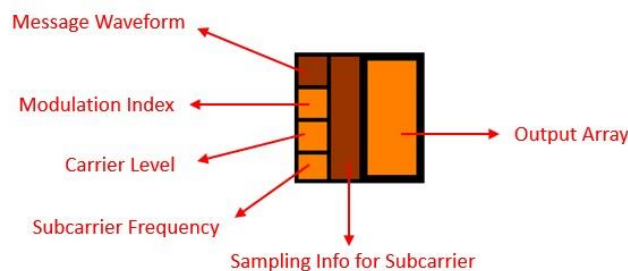


Figure 13: Filled Pattern of Connector Pane

If there are not enough terminals, you can right click the Connector Pane and select **Add Terminals** or change the **Patterns**. For the AM subcarrier VI, use Figure 14 pattern and assign inputs and outputs based on Figure 13. Now you have completed the first task and the sub-vi is ready to be imported to other VIs. If you need the Amplitude Modulator in any of the future labs, you can use this sub-vi in your main VI and do the modulation easily.

- XVII. Save your sub-vi in a file whose name includes the letters **AMSubcarrier** and your initials. Be sure to save your transmitter as a “vi” and not a “vit” (template). *Note: You need to*

*upload your program file with your lab report.*



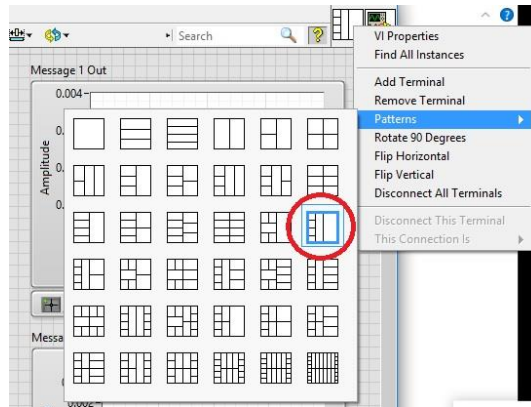


Figure 14: Pattern of Connector Pane

So, we completed the first task (A) to create a sub-vi. For the second task (B), you will use the transmitter template which has been provided in the file **Lab3TxTemplate.vit**. This template contains the four VIs for interfacing with the USRP along with two message generators that will generate waveforms  $m_1(t)$  and  $m_2(t)$ .

As you can see, the transmitter template contains two **Message Generator** blocks to generate two distinct messages. In order to show the generated messages before the modulation step, the message signals are combined into an array of two messages and then the result is shown on a single Waveform Graph icon called **Message**. Note that this Waveform graph shows the unmodulated messages.

B. Update the transmitter:

1. Use two instances of your **AMonSubcarrierYOURINITIALS.vi** (created as the first task) to generate the signals waveforms  $s_1(t)$  and  $s_2(t)$ . Follow the following steps for both sub-vi's to set up them appropriately:

- I. In TX template go to Function > Select a VI... then select your sub-vi. The **AMonSubcarrierYOURINITIALS.vi** is shown in Figure 15.

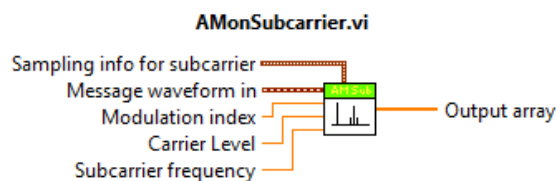


Figure 15: AM sub-vi



- II. For the first input of your sub-vi, **Sampling info for subcarrier**, use the output cluster of **Bundle By Name** icon. This icon is already located in the transmitter template and shown in Figure 16.

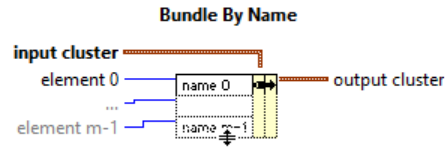


Figure 16: Bundle By Name VI

- III. Wire the **signal out** output of the Message Generator to the **Message waveform in** input of your sub-vi. Note that you have two Message Generator blocks.

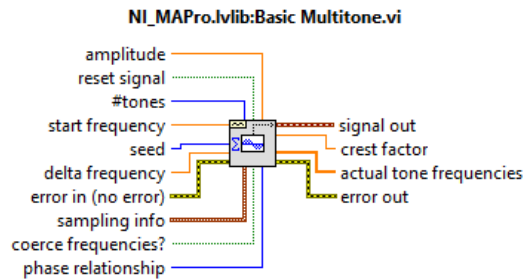


Figure 17: Basic Multitone Message Generator

- IV. Use a constant value of 1 for **Carrier level** input. (Functions > Programming > Numeric > Numeric Constant).
- V. Two Numeric Inputs are provided, Modulation index and Sub-carrier frequency. Wire them properly to your sub-vi.
2. As we discussed, in order to have a FDM signal, you need to combine the modulated messages. Use **Add** icon (Functions > Programming > Numeric > Add) to combine the output of two sub-vi's,  $s_1(t)$  and  $s_2(t)$ , together.
- Now, you have your FDM signals ready (output of the Add icon), but before transmitting them into the USRP and over the channel you need to make complex signals. As we discussed in Lab 2, the USRP is designed to transmit using a quadrature modulation approach. So in order to use the USRP to transmit an AM signal, it is necessary to represent the signal as a complex sequence, however

for Quadrature part, we only send zero. Here, you need to make Real and Imaginary vectors to transmit to USRP. So:

3. Normalize the peak value of the sum ( $s_1(t) + s_2(t)$ ) to one by using the first output of Quick Scale.
4. In order to form a complex array similar to Eq. 4, use **Re/Im to Complex** icon (Functions > Programming > Numeric > Re/Im to Complex). For the real input, use the normalized sum from step 3. For the imaginary input, you need to create another array of the same length as  $s_1(t) + s_2(t)$  but containing zero values. The following steps describe how to generate an array of zero with the same length as  $s_1(t) + s_2(t)$ :

- I. To generate the same length array, use **Array Size** icon (Functions > Programming > Array > Array Size). This icon gives you the size of the input array, feed this icon with  $s_1(t) + s_2(t)$  (output of the Add icon).



Figure 18: Array Size Icon

- II. In order to generate the array of zeros, use **Initialize Array** (Functions > Programming > Array > Initialize Array) and wire the **size** output of the Array Size block to the **dimension** input of the Initialize Array block. For **element** input, use constant value of zero.

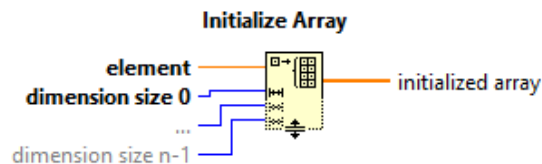
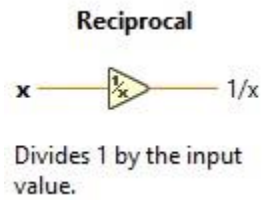


Figure 19: Initialize Array

5. Use Waveform Graph to show the real part of the signal as the Baseband signal.
6. Use **Build Waveform** icon (Functions > Programming > Waveform > Build Waveform) to build a waveform from the complex data values. Set the sampling time (**dt** component of the Get

Waveform block) equal to the reciprocal of the **Actual IQ rate** by using the **Reciprocal** icon (See Programming > Numeric > Reciprocal).



*Figure 20: Reciprocal Icon*

7. After building the complex waveform  $\tilde{s}(t)$ , pass it into the **While Loop**, and connect it to the **Write Tx Data vi**.
8. Save your transmitter in a file whose name includes the letters “MuxTx” and your initials. Be sure to save your transmitter as a “vi” and not a “vit” (template). *Note: You need to upload your program file with your lab report.*



## Receiver:

As we discussed earlier and based on the Figure 6, the receiver architecture consists of a bandpass filter and an amplitude demodulator. Similar to Lab 2, you must use a rectifier and a LPF to detect the envelope of the amplitude modulated signal.

A template for the receiver has been provided in the file **Lab3RxTemplate.vit**. This template contains six VIs for interfacing with the USRP along with two waveform graphs on which to display your demodulated output signals. The task for the receiver is to add as many blocks as needed to reconstruct the AM modulated messages. As it's mentioned in Lab 2, the required blocks are added in the While Loop by using the output of the **Fetch Rx Data**. The procedure to reconstruct the messages is the same as Amplitude Demodulation. Follow the following steps to create your receiver:

1. The Get Waveform Components block gives you the components of the received waveform. Use the output **Y** for signal processing.
2. Since we have two multiplexed signal, you need to design two separate amplitude demodulators. So, similar to what we did in Lab 2, use two **Chebyshev** bandpass filters (Functions > Signal Processing > Filters > Chebyshev) with the same bandwidth and different cut off frequencies.

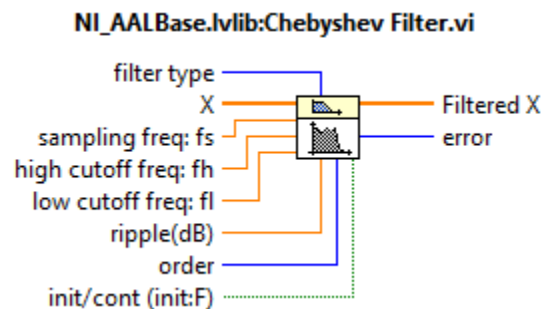


Figure 21: Chebyshev Filter

Set one of the filters to have the high cutoff frequency of 505 kHz and the low cutoff frequency of 495 kHz and set the other one to have the high cutoff frequency of 515 kHz and the low cutoff frequency of 505 kHz. For both filters set the order to 5 and the default passband ripple of 0.1 dB is acceptable. The sampling frequency input to both filters can be either the **actual IQ rate** obtained

from the **Configure Signal** block (outside of the while loop) or the reciprocal value of **dt** sampling time of the Get Waveform Components block.

3. Pass the output of each bandpass filter through an envelope detector which consists of an **Absolute Value** block followed by a LPF.

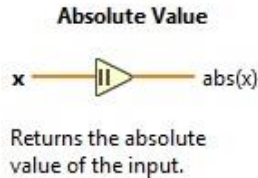


Figure 22: Absolute Value Icon

4. For LPFs, use two fifth order Chebyshev filters with the low cutoff frequency of 5000 Hz. Each LPF output must be wired to its own **Build Waveform block** in order to generate the waveform from sequence of data. Use the provided waveform graphs, **Message 1 Out and Message 2 Out** to plot the demodulated waveforms.
5. In order to use the Chebyshev filters to process a large data sequence consisting of smaller blocks, LabVIEW suggests to initialize the filter only when the VI runs for the first time. This requires to set the filter initialization input to FALSE only for the first loop, then set it to TRUE for the remaining process. In order to do that, you need to use the **First Call?** block. This block, which is shown in Figure 23, returns TRUE only the first time it's called after the **Run button** is pressed.



Figure 23: First Call? Block

Since the Chebyshev filter needs FALSE value to go to initialization mode, you need to use the First Call? block along with a **Not** block to convert the TRUE to FALSE.

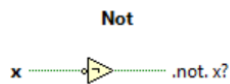


Figure 24: Logical Not Block

Wire the output of the Not to the initialize input of each Chebyshev filters to complete your code.

6. Save your receiver in a file whose name includes the letters “MuxRx” and your initials. Be sure to save your receiver as a “vi” and not a “vit” (template). **Note: You need to upload your program file with your lab report.**

### Lab Procedure:

1. Connect the loopback cable and attenuator between the TX 1 and RX 2 antenna connectors. Connect the USRP to your computer with an Ethernet cable and plug in the power to the radio. At this point LED’s **D** (firmware loaded) and **F** (power on) should be illuminated, as should the green light on the left side of the Ethernet connector. Run LabVIEW and open the transmitter and receiver VIs that you created in the prelab.
2. Ensure that the transmitter vi is set up to use:

Table 2: Transmitter Settings

Field	Setting
Device Name	192.168.10.2
IQ Rate	2 MS/s
Carrier Frequency	915 MHz
Gain	0 dB
Active Antenna	TX1
Message Length	Not critical, but 200,000 samples give a good block of data to work with.

3. Ensure that the message generator is set up to use Table 3 setting.
4. Ensure the receiver VI is set up to use Table 4 setting.

Note that there is no offset between the transmitter’s carrier frequency and the receiver’s carrier



frequency in this lab project.

Table 3: Transmitter Message Settings

Field	Message 1 Settings	Message 2 Settings
Start Frequency	10 Hz	100 Hz
Delta Frequency	100 Hz	100 Hz
Number of Tones	1	2
Modulation Index	1	1
Subcarrier Frequency	500 kHz	510 kHz

- Run the receiver VI, then run transmitter VI. LED C will illuminate on the USRP if the radio is receiving data. After a few seconds, stop the receiver and then stop the transmitter using the STOP button. Examine the Message Out graphs to ensure that the receiver is correctly demodulating and displaying the two message signals. (1 point)

Table 4: Receiver Settings

Field	Setting
Device Name	192.168.10.2
IQ Rate	2 MS/s
Carrier Frequency	915 MHz
Gain	0 dB
Active Antenna	RX2
Message Length	200,000

## Experiments:

Do the following experiments, plot the graphs in your report and describe the results.

- What would happen if one message is turned off? (1 point)
  - Turn off the second message



- II. Run your receiver
  - III. Run your transmitter
2. What would happen if one message has a much larger carrier level than the other? (1 point)
  - I. Decrease the carrier level of the second message to 0.1
  - II. Run your receiver
  - III. Run your transmitter
3. Cross-Talk: In telecommunications, crosstalk occurs when the signal from an adjacent channel causes distortion or interference. You may have experienced this when using a telephone and hearing someone else's conversation faintly in the back ground. (1 point)
  - I. Set the transmitter to use of only one tone for both messages. (Using more than one tone will make it difficult to observe the different messages.)
  - II. Set the first subcarrier frequency equals to 505 kHz, 508 kHz and 510 kHz while keep the second subcarrier frequency as 510 kHz.
  - III. Run your receiver
  - IV. Run your transmitter
4. Is crosstalk observable if the order of the bandpass filters is reduced? (1 point)
  - I. Set the first subcarrier frequency to 500 kHz and the second subcarrier frequency to 510 kHz.
  - II. Reduce the order of the second bandpass filter to 1
  - III. Run your receiver
  - IV. Run your transmitter
  - V. Set the order of the second bandpass filter to 5
  - VI. Reduce the order of the first bandpass filters to 1
  - VII. Run your receiver
  - VIII. Run your transmitter



5. What about the order of the LPFs? (1 point)

I. Repeat the last step for LPFs.

### **Report (4 points):**

The reports must be prepared in PDF or Microsoft Word format and should follow the standard of laboratory report. The report shall consist of:

1. A title page, including the name and number of the lab
2. The object, purpose and goal of the experiment shall be on the second page
3. A quick description on how the experiment is being done in 1-2 paragraphs
4. Calculations
5. Graphs
6. Pictures
7. Results
8. Conclusion
9. Tx and Rx VI files

All reports are due one week after the receiver experiment is completed and each team must submit only one report along with all the results and necessary files in a zip format file to the Moodle page.

### **References**

- [1] A. Abrol and E. Hamke, *Introduction to Communication Systems Using National Instruments Universal Software Radio Peripheral Lab Manuals*, The University of New Mexico, 2014.
- [2] "Tutorial: SubVIs," National Instruments, 01 April 2015. [Online]. Available: <http://www.ni.com/white-paper/7593/en/>. [Accessed 24 February 2018].

