LAB 2

Amplitude Modulation

Prerequisite: Lab 1 – Introduction to the USRP

2.1 Objective

This laboratory exercise has two objectives. The first is to gain a firsthand experience in actually programming the USRP to act as a transmitter and a receiver. The second is to investigate classical analog amplitude modulation and the envelope detector.

2.2 Background

Amplitude Modulation

Amplitude modulation (AM) is one of the oldest of the modulation methods. It is still in use today in a variety of systems, including, of course, AM broadcast radio. In digital form it is the most common method for transmitting data over optical fiber.

If $m(t)$ is a baseband "message" signal with a peak value m_p and $A\cos(2\pi f_c t)$ is a "carrier" signal at carrier frequency f_c, then we can write the AM signal $g(t)$ as

$$g(t) = A\left[1 + \mu \frac{m(t)}{m_p}\right]\cos(2\pi f_c t), \tag{1}$$

where the parameter μ is called the "modulation index" and takes values in the range $0 \le \mu \le 1$ (0 to 100%) in normal operation. For the special case in which $m(t) = m_p \cos(2\pi f_m t)$, we can write

$$\begin{aligned} g(t) &= A\left[1 + \mu\cos(2\pi f_m t)\right]\cos(2\pi f_c t) \\ &= A\cos(2\pi f_c t) + \frac{A}{2}\mu\cos\left[2\pi(f_c - f_m)t\right] + \frac{A}{2}\mu\cos\left[2\pi(f_c + f_m)t\right]. \end{aligned} \tag{2}$$

In the above expression the first term is the carrier, and the second and third terms are the lower and upper sidebands, respectively. Figure 1 is a plot of a 20 kHz carrier modulated by a 1 kHz sinusoid at 50% and 100% modulation.

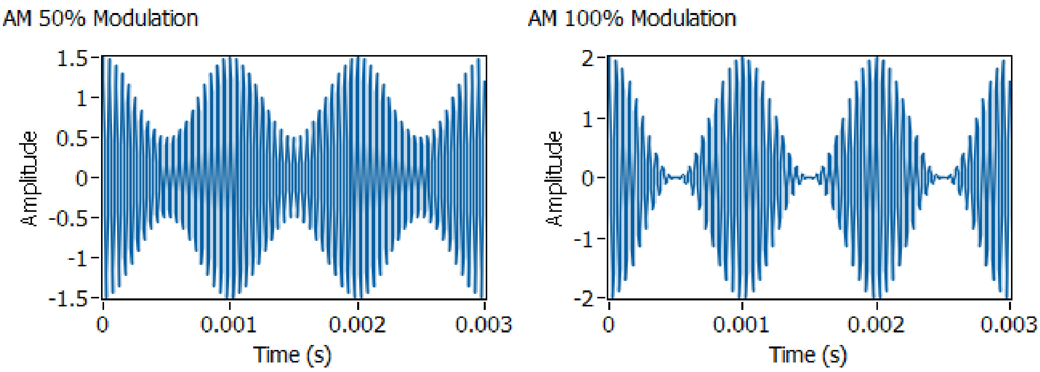


Figure 1. Amplitude Modulated Signals

When the AM signal arrives at the receiver, it has the form

$$r(t) = D\left[1 + \mu \frac{m(t)}{m_p}\right]\cos(2\pi f_c t + \theta), \tag{3}$$

where the carrier amplitude D is usually much smaller than the amplitude A of the transmitted carrier and the angle θ represents the difference in phase between the transmitter and receiver carrier oscillators. We will follow a common practice and offset the receiver's oscillator frequency f_0 from the transmitter's carrier frequency f_c. This provides the signal

$$r_1(t) = D\left[1 + \mu \frac{m(t)}{m_p}\right]\cos(2\pi f_{IF} t + \theta), \tag{4}$$

where the so-called "intermediate" frequency is given by $f_{IF} = f_c - f_0$. The signal $r_1(t)$ can be passed through a bandpass filter to remove interference from unwanted signals on frequencies near f_c. Usually the signal $r_1(t)$ is amplified as well.

Demodulation of the signal $r_1(t)$ is most effectively carried out by an envelope detector. An envelope detector can be implemented as a rectifier followed by a lowpass filter. The envelope $A(t)$ of $r_1(t)$ is given by

$$A(t) = D\left[1 + \mu \frac{m(t)}{m_p}\right] = D + \frac{D\mu}{m_p} m(t). \tag{5}$$

Setting up the USRP

Transmitter

LabVIEW interacts with the USRP transmitter by means of four functions located on the block diagram's palette under Hardware Interfaces→NI-USRP→Tx. Figure 2 shows the basic transmitter structure. This structure is the starting point for all of the laboratory exercises in this series.

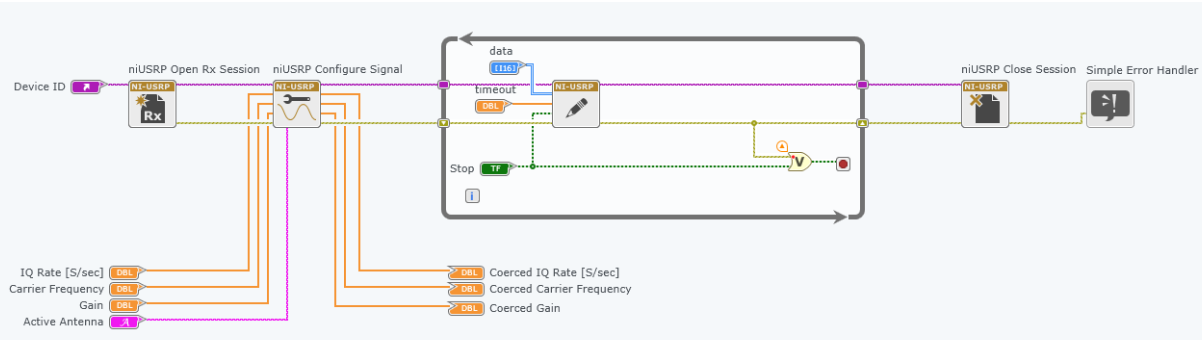


Figure 2. Transmitter Template

Open Tx Session initiates the transmitter session and generates a session handle and an error cluster that are propagated through all four functions. When you use this function, you must add a control called "device names" that you will use to inform LabVIEW of the IP address or resource name of the USRP.

Configure Signal is used to set parameter values in the USRP. Attach four controls and three indicators to this function as shown in the figure. To get started, set the IQ rate to 200 kSa/s (the lowest possible rate), the carrier frequency to 915.1 MHz, the gain to 0 dB, and the active antenna to TX1. When the function runs, the USRP will return the actual values of these parameters. These values will be displayed by the indicators you connected. Normally the actual parameter values will match the desired values, but if one or more of the desired values is outside the capability of the USRP, the nearest acceptable parameter value will be chosen, rather than returning an error.

Write Tx Data writes the baseband signal to the USRP for transmission. Placing this function in a *while* loop allows a block of baseband signal samples to be sent over and over until the "stop" button is pressed. Note that the *while* loop is programmed to terminate if an error is detected. Baseband signal samples can be provided to the *Write Tx Data* as either an array of complex numbers or as a complex waveform data type. The Configure ribbon at the top of LabVIEW Communications allows you to choose the data type. If the baseband signal is expressed as

$$\tilde{g}(nT_x) = g_I(nT_x) + jg_Q(nT_x), \tag{6}$$

then the signal transmitted by the USRP is

$$g(t) = Ag_I(t)\cos(2\pi f_c) - Ag_Q(t)\sin(2\pi f_c t). \tag{7}$$

In this expression, the constant A is set by the "gain" parameter and f_c is the carrier frequency. The sampling interval T_x is the reciprocal of the "IQ rate." Note that the signal $g(t)$ produced by the USRP is a continuous-time signal; the discrete-to-continuous conversion is done inside the USRP.

Observe that the baseband signal $\tilde{g}(nT_x)$ is actually two baseband signals. By long-standing tradition, the real part $g_I(nT_x)$ is called the "in-phase" component of the baseband signal and the imaginary part $g_Q(nT_x)$ is called the "quadrature" component of the baseband signal. The AM signal that we will generate in this lab project uses only the in-phase component, with

$$g_I(t) = A\left[1 + \mu\frac{m(t)}{m_p}\right], \tag{8}$$

and

$$g_Q(t) = 0. \tag{9}$$

We will explore other modulation methods in subsequent lab projects that use both components.

<u>Close Session</u> terminates transmitter operation once the *while* loop ends. Note that the function should be terminated using the STOP button rather than with "Abort Execution" on the toolbar. This is so that the Close Session function will be sure to run and will correctly close out the data structures that the function uses.

<u>Receiver</u>
LabVIEW interacts with the USRP receiver by means of six functions located on the block diagram in Hardware Interfaces→NI-USRP→Rx. Figure 3 shows the basic receiver structure. This structure is the starting point for all of the laboratory exercises in this series.

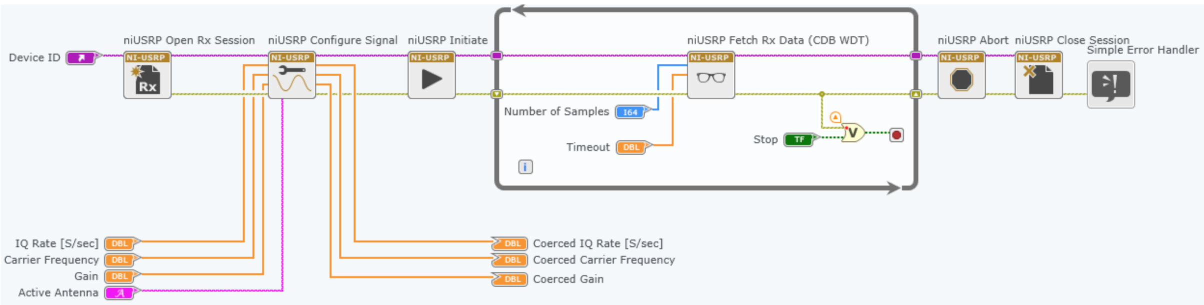


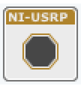
Figure 3. Receiver Template

Open Rx Session initiates the receiver session and generates a session handle and an error cluster that are propagated through all six functions. You must add a control called "device names" that you will use to inform LabVIEW of the IP address or resource name of the USRP.

Configure Signal has the same function as the corresponding function in the transmitter. Attach four controls and three indicators to this function as shown in the figure. This time, set the IQ rate to 1 MSa/s, the carrier frequency to 915.0 MHz, the gain to 0 dB, and the active antenna to RX2. When the function runs, the USRP will return the actual values of these parameters.

Initiate sends the parameter values you selected to the receiver and starts it running.

Fetch Rx Data retrieves the message samples received by the USRP. Placing this function in a *while* loop allows message samples to be retrieved one block at a time until the "stop" button is pressed. Note that the *while* loop is programmed to terminate if an error is detected. A "number of samples" control allows you to set the number of samples that will be retrieved with each pass through the *while* loop. In later lab projects in this series, we will not use the *while* loop, and will fetch only a single block of data from the receiver. *Fetch Rx Data* can provide message samples to the user as either an array of complex numbers or as a complex waveform data type. A pull-down tab allows you to choose the data type for the message samples.



Abort stops the acquisition of data once the *while* loop ends.

Close Session terminates receiver operation. As noted above, use the STOP button to terminate execution so that *Close Session* will be sure to run.

File Hierarchy

To access the files and functions specific to this course double click the *Communication Systems.lvproject.* After the project is loaded, you should see the Files Pane located on the top left of the screen, just below the ribbon. Files and functions can be accessed by expanding the folders and double clicking the file you would like. You can also drag and drop files from this pane to the block diagram to use it.

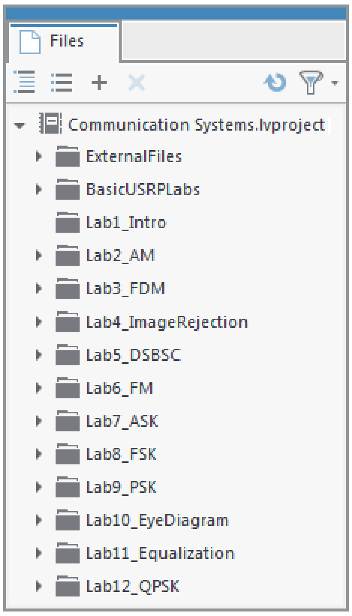


Figure 4. Files Pane in LabVIEW Communications

2.3 Pre-Lab

Transmitter

1. A template for the transmitter has been provided in the file *Lab2TxTemplate.gvi*. This template contains the four interface functions described in the Background section above along with a "message generator" (Basic Multitone) that is set to produce a message signal consisting of three tones. The three tones are initially set to 1, 2, and 3 kHz, but these frequencies can be changed using front-panel controls. Your task is to add blocks as needed to produce an AM signal, and then to pass the AM signal into the *while* loop to the Write Tx Data function. The modulation index is to be user-settable in the range $0 \le \mu \le 1$, and a front-panel control has been provided.

Hint: The AM signal you generate will be $g_I(nT)$. For $g_Q(nT)$ set up an array the same length as $g_I(nT)$ containing all zeros. Then combine the two into a single complex array $\tilde{g}(nT) = g_I(nT) + jg_Q(nT)$. You can use the MathScript node to implement the AM Modulation formula easier if you know .m file syntax

Notes:

a. The message generator creates a signal that is the sum of a set of sinusoids of equal amplitude. You can choose the number of sinusoids to include in the set, you can choose their frequencies, and you can choose their common amplitude. The initial phase angles of the sinusoids are chosen at random, however, and will be different every time you run the program. This will make the message signal look somewhat different every time you run the program.

b. There is one practical constraint imposed by the D/A converters in the USRP: Scale the signals you generate so that the peak value of $|\tilde{g}(nT)|$ does not exceed +/-1. (+/-1 usually refers to full scale on the DAC in a device. Any value higher will result in clipping.)

c. Save your transmitter in a file whose name includes the letters "AMTx" and your initials (e.g., *AMTx_BAB.gvi*).

Receiver

2. A template for the receiver has been provided in the file *Lab2RxTemplate.gvi*. This template contains the six interface functions described in the Background section above along with a waveform graph on which to display your demodulated output signal.

   Pass the complex array returned by *Fetch Rx Data* through a bandpass filter. Filters can be found in the Functions Palette of your block diagram under Analysis>>Signal Processing>>Filters. Use a fifth-order Chebyshev Filter with a high cutoff frequency of 105 kHz and a low cutoff frequency of 95 kHz. The default passband ripple of 0.1 dB is acceptable.

3. To extract the envelope, take the absolute value of the Chebyshev bandpass filter and pass the result through a lowpass filter. This acts as a full-wave rectifier. Then, to extract the message, pass the signal through the lowpass filter. For the lowpass filter, use a second-order Butterworth Filter with a cutoff frequency of 5 kHz. As was the case for the bandpass filter, the sampling frequency input to the lowpass filter should be the "actual IQ rate" obtained from the *Configure Signal*. The output of your lowpass filter should be connected to the Baseband Output graph.

   Save your receiver in a file whose name includes the letters "AMRx" and your initials (e.g., *AMRx_BAB.gvi*).

Questions

1. Suppose the message $m(t)$ is given by

$$m(t) = \cos(2\pi 1000t) + \cos(2\pi 2000t) + \cos(2\pi 3000t). \tag{10}$$

   Find and plot the power spectrum of $r_1(t)$ given by Eq. (4). Leave your answer in terms of D and μ.

2. For the message of Eq. (10), find and plot the power spectrum of $A(t)$ given by Eq. (10). Leave your answer in terms of D and μ.

3. Qualitatively, based on your answer to Question 1, what happens to the power in the carrier as the modulation index μ is varied? What happens to the power in the sidebands?

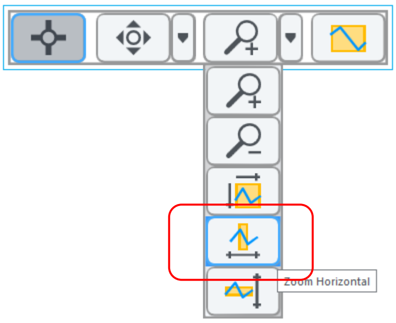
2.4 Lab Procedure

1. Connect a loopback cable and attenuator between the TX 1 and RX 2 connectors. Connect the USRP to your computer and plug in the power to the USRP. Run LabVIEW and open the transmitter and receiver functions that you created in the prelab.

2. Ensure that the transmitter is set up to use
   Carrier Frequency: 915.1 MHz
   IQ Rate: 200 kHz
   Gain: 0 dB
   Active Antenna: TX1
   Message Length: 200,000 samples gives a good block of data to work with.
   Modulation Index: Start with 1.0.
   Start Frequency, Delta Frequency, Number of Tones: Three tones seems to work well, but keep the highest frequency below 5 kHz.

3. Ensure that the receiver is set up to use
   Carrier Frequency: 915 MHz
   IQ Rate: 400 kHz
   Gain: Not critical. 0 dB
   Active Antenna: RX2
   Number of Samples: Same value as the transmitted message length.

Run the transmitter, and then run the receiver. After a few seconds, stop the receiver using the STOP button, then stop the transmitter. Use the horizontal zoom feature on the graph palette to expand the "message" waveform in the transmitter and the "baseband output" waveform in the receiver.



Both waveforms (transmit and receive) should be identical, except for scaling and possible DC offset.

4. *Power Spectrum*
   Add the *FFT Power Spectrum and PSD* function (Analysis>>Signal Processing>>Measurement) to your receiver block diagram. Configure the function for "Power Spectrum" and "Continuous". To provide input to this power spectrum function, connect the "data" output from the niUSRP Fetch Rx Data function with the "signal" input on the power spectrum function Only one other input of the power spectrum function needs to be connected: Wire a Boolean constant set to True to the "dB On" input. Now connect the "Power Spectrum/PSD" output of your power spectrum function to a waveform graph. On the waveform graph, make the "graph palette" visible so that you can use the zoom feature, and change the label on the horizontal axis to "Frequency."

   Set the transmitter to generate a message consisting of three tones starting at 1 kHz with a 1 kHz spacing. Set the modulation index to $\mu = 1$. Run the transmitter and then the receiver. Stop the receiver and then stop the transmitter. Zoom in on the power spectrum so that you can clearly see the components in the vicinity of 100 kHz. Take a screen shot of your power spectrum graph.

   Compare the power spectrum with the spectrum you predicted in Prelab Question 1. How many dB below the carrier are your sideband components?

   Change the modulation index to $\mu = 0.5$ and capture a new power spectrum. Take another screenshot of your power spectrum graph. How many dB below the carrier are your sideband components now?

5. The constant D that represents the amplitude of the received carrier can be measured by passing the envelope of Eq. (5) through a lowpass filter. A measured value of D is often used in practical receivers to adjust the gain of the receiver's output, providing an "automatic gain control" feature.
   Add a lowpass filter to your receiver such that the filter output is proportional to D. Run the transmitter and receiver, and measure the value of D. Increase the gain of the receiver to 20 dB and repeat the measurement of D. Is the change in D consistent with a 20 dB change in receiver gain?

6. *Elegant receiver*
   The signal at the output of the bandpass filter has a real part that is given by Eq. (4). The imaginary part is given by

$$r_2(t) = A_r\left[1+\mu\frac{m(t)}{m_p}\right]\sin(2\pi f_{IF}t+\theta). \tag{11}$$

The complex signal at the output of the bandpass filter is therefore

$$\begin{aligned}\tilde{r}(t) &= r_1(t) + jr_2(t) \\ &= A_r\left[1+\mu\frac{m(t)}{m_p}\right]\cos(2\pi f_{IF}t+\theta) + jA_r\left[1+\mu\frac{m(t)}{m_p}\right]\sin(2\pi f_{IF}t+\theta) \\ &= A_r\left[1+\mu\frac{m(t)}{m_p}\right]\left[\cos(2\pi f_{IF}t+\theta) + j\sin(2\pi f_{IF}t+\theta)\right] \\ &= A_r\left[1+\mu\frac{m(t)}{m_p}\right]e^{j(2\pi f_{IF}t+\theta)}.\end{aligned} \tag{12}$$

The magnitude of $\tilde{r}(t)$ is $A_r\left[1+\mu\frac{m(t)}{m_p}\right]$, which is the desired demodulated output.

Connect the bandpass filter output directly to the absolute value block (bypass the *Complex to Re/Im* function). Connect the absolute value output directly to the Baseband Output graph (bypass the *Butterworth Filter*). Run the transmitter and receiver, and observe that the demodulated output is the same as it was in step 3. There is no need for the lowpass filter! (Note that since the modulation index μ is upper-bounded by one, the expression $A_r\left[1+\mu\frac{m(t)}{m_p}\right]$ is never negative, and is not affected by the absolute value).

2.5 Report

Prelab

Hand in documentation for the functions you created for the transmitter and receiver. Also include documentation for any sub-functions you may have created. To obtain documentation, print out legible screenshots of the front panel and block diagram.

Answer the questions in the *Questions* section at the end of the prelab instructions.

Lab

Submit the functions you created for the transmitter and receiver. Also submit any sub-functions you may have created. Be sure your files adhere to the naming convention described in the instructions above.
Resubmit documentation for any functions you modified during the lab.

Submit the spectrum graphs and answer all of the questions in Sections 4 and 5 of the Lab Procedure.