LAB 5

Double-Sideband Suppressed-Carrier

Prerequisite: Lab 2 – Amplitude Modulation

5.1 Objective

This laboratory exercise introduces suppressed-carrier modulation. A simple scheme for phase and frequency synchronization is introduced in implementing the demodulator.

5.2 Background

Double-Sideband Suppressed-Carrier

Amplitude modulation is inherently inefficient because the largest part of the transmitted power is contained in the carrier. In suppressed-carrier schemes the carrier is simply not transmitted. There are two common suppressed-carrier techniques in use, double-sideband suppressed-carrier (DSB-SC) and single-sideband (SSB). Double-sideband suppressed-carrier modulation is identical to AM, except that the carrier is omitted.

If $m(t)$ is a baseband "message" signal and $\cos(2\pi f_c t)$ is a "carrier" signal at carrier frequency f_c, then we can write the DSB-SC signal $g(t)$ as

$$g(t) = Am(t)\cos(2\pi f_c t). \tag{1}$$

For the special case in which $m(t) = m_p \cos(2\pi f_m t)$, we can write

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

The two terms in Eq. (2) represent the lower and upper sidebands, respectively. There is no carrier term at frequency f_c. Figure 1 is a plot of a 20 kHz carrier modulated by a 1 kHz sinusoid using DSB-SC modulation.

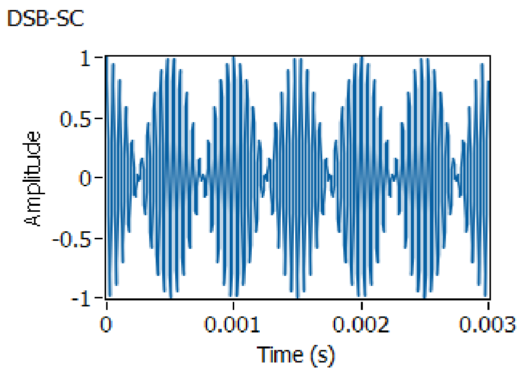


Figure 1. Double-Sideband Suppressed-Carrier Modulation

When the DSB-SC signal arrives at the receiver, it has the form

$$r(t) = Dm(t)\cos(2\pi f_c t + \theta), \tag{3}$$

where D is a constant, usually much smaller than A, and the angle θ represents the difference in phase between the transmitter and receiver carrier oscillators. If the receiver's carrier oscillator (the "local" oscillator) is set to the same frequency as the transmitter's carrier oscillator, the USRP will generate the two demodulated signals

$$\begin{aligned} r_I(t) &= \frac{D}{2} m(t)\cos(\theta), \text{ and} \\ r_Q(t) &= \frac{D}{2} m(t)\sin(\theta). \end{aligned} \tag{4}$$

The *Fetch Rx Data* provides these demodulated signals as a single complex-valued signal $\tilde{r}(t)$ given by

$$\begin{aligned} \tilde{r}(t) &= \frac{D}{2} m(t)\cos(\theta) + j\frac{D}{2} m(t)\sin(\theta) \\ &= \frac{D}{2} m(t) e^{j\theta}. \end{aligned} \tag{5}$$

It is tempting to suppose that the message $m(t)$ can be extracted from $\tilde{r}(t)$ by taking the magnitude of the complex signal. Unfortunately, the magnitude of $\tilde{r}(t)$ is

$$|\tilde{r}(t)| = \frac{D_r}{2} |m(t)|, \tag{6}$$

where the absolute value represents unwanted distortion of the message signal. It is more productive to use the in-phase (real part) signal $r_I(t)$ given in Eq. (4). The $\cos(\theta)$ factor of

$r_I(t)$ represents a gain constant. Unfortunately, the value of this gain constant is not under user control, and might be small if θ turns out to have a value near $\pm\pi/2$. Moreover, if the receiver's oscillator and transmitter's oscillator differ slightly in frequency, then the phase error θ will change with time, causing $r_I(t)$ to fade in and out. The next section discusses how we can compensate for the $\cos(\theta)$ term.

Phase Synchronization

There are a number of techniques that can be used to eliminate the $\cos(\theta)$ phase-error term. The method we present here is simple and easy to implement in LabVIEW. The basic steps are

- Estimate θ
- Multiply $\tilde{r}(t)$ by $e^{-j\theta}$ to produce $\tilde{r}(t)e^{-j\theta} = \frac{D}{2}m(t)e^{j\theta}e^{-j\theta} = \frac{D}{2}m(t)e^{j0}$
- Take the real part: $\mathrm{Re}\left\{\frac{D}{2}m(t)e^{j0}\right\} = \frac{D}{2}m(t)\cos(0) = \frac{D}{2}m(t)$.

Estimating θ requires several steps. Note first that the phase angle of $\tilde{r}(t)$ will jump by $\pm\pi$ whenever $m(t)$ changes sign. To eliminate these phase jumps, start by squaring $\tilde{r}(t)$:

$$\tilde{r}^2(t) = \frac{D^2}{4}m^2(t)e^{j2\theta}. \tag{7}$$

Since the squared message $m^2(t)$ never changes sign, the phase jumps are eliminated. The angle 2θ can be extracted using a Complex to Polar function from the Data Types→Numeric→Complex palette. It turns out to be helpful at this point to smooth variations in 2θ caused by noise. The *median filter* from the ExternalFiles folder does a good job. The default values can be accepted for the "left rank" and "right rank" parameters. Next the Unwrap Phase function from the Analysis→Signal Processing→Conditioning palette will remove jumps of $\pm 2\pi$. Finally, dividing by two gives the desired estimate of the phase error θ. The block diagram in Figure 2 shows the entire phase synchronization process.

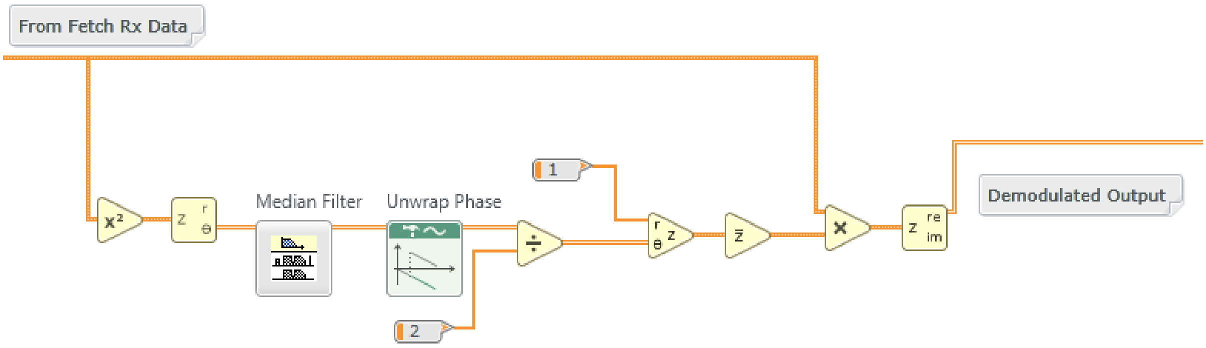


Figure 2. Phase Synchronization

5.3 Pre-Lab

Transmitter

1. A template for the transmitter has been provided in the file Lab5TxTemplate.gvi. This template contains the four interface functions along with a "message generator" 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 a DSB-SC signal, and then to pass the DSB-SC signal into the *while* loop to the Write Tx Data block.

Hint: The DSB-SC 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 zeroes. Then combine the two into a single complex array $\tilde{g}(nT) = g_I(nT) + jg_Q(nT)$.

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. In this template the message generator has been provided with a "seed." This causes the initial phase angles of the sinusoids to be the same every time you run the program. As a result, the same message will be generated every time, which is useful to aid debugging. To restore random behavior, set the seed to -1.

b. There is a practical constraint imposed by the D/A converters in the USRP: Scale the signals you generate so that the peak value of $\left|\tilde{g}(nT)\right|$ does not exceed +/- 1. (Check out the *Quick Scale 1Dfunction* in the ExternalFiles folder.)

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

Receiver

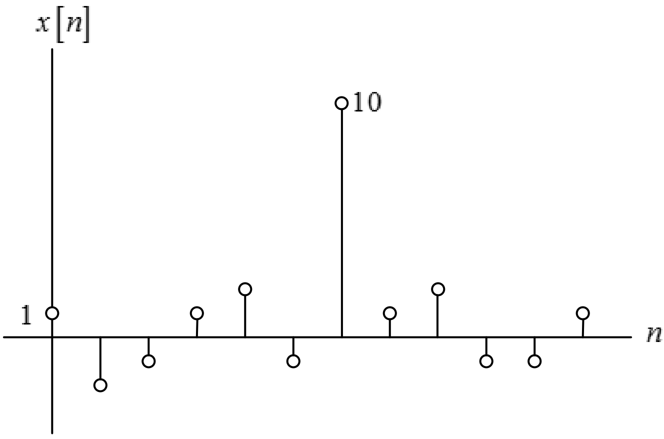
2. A template for the receiver has been provided in the file Lab5RxTemplate.gvi. This template contains the six interface functions along with a waveform graph on which to display your demodulated output signal.

   Complete the program to demodulate the complex array returned by the Fetch Rx Data function and display the result. Include the phase synchronization of Figure 2. Also, to help in debugging, include a graph to display the phase error θ vs. time.

   Save your receiver in a file whose name includes the letters "DSBSCRx" and your initials.

Questions

1. The graph below shows a sequence of samples having values $x = [1 \quad -2 \quad -1 \quad 1 \quad 2 \quad -1 \quad 10 \quad 1 \quad 2 \quad -1 \quad -1 \quad 1]$ Assume that values of $x[n]$ at index values not shown are zero. Note that there is a single outlying sample at index $n = 6$.



Suppose we have a simple *median filter* than produces an output $y[n]$ given by

$$y[n] = \text{median}\{x[n-2],\ldots,x[n+2]\}, \quad \mathrm{n} = 0,\ldots,11.$$

Find $y[n]$ for the sample sequence $x[n]$ shown.

2. Suppose the receiver's carrier oscillator differs in frequency from the transmitter's oscillator by a small offset Δf. Modify Eqs. (4) and (5) to include the frequency offset.

5.4 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 that you created in the prelab.

2. Ensure that the transmitter is set up to use

   Carrier Frequency: 915 MHz

   IQ Rate: Not critical. 200 kHz

   Gain: Not critical. 0 dB

   Active Antenna: TX1

   Message Length: 200,000 samples gives a good block of data to work with.

   Start Frequency, Delta Frequency, Number of Tones: Not critical, but keep the highest frequency below 5 kHz. Three tones seems to work well.

3. Ensure that the receiver is set up to use

   Carrier Frequency: 915 MHz

   IQ Rate: 200 kHz

   Gain: Not critical. 0 dB

   Active Antenna: RX2

   Number of Samples: Same value as the transmitted message length.

   Run the transmitter, then run the receiver. After a few seconds, stop the receiver using the STOP button, then stop the transmitter (using the STOP button). Use the horizontal zoom feature on the graph palette to expand the "message" waveform in the transmitter and the "demodulated output" waveform in the receiver. Both waveforms should be identical, except for scaling.[5]

4. Modify your receiver to compute $r_I(t)$ without phase synchronization and $r_I(t)$ with phase synchronization. Plot both outputs on the same graph. Run the receiver several times and observe the outputs. Can you see the effect of the $\cos(\theta)$ term on the unsynchronized output?

5. Try using your AM receiver from Lab 2 to demodulate the DSB-SC signal. Note that you will need to offset the transmitter frequency to 915.1 MHz. Run the transmitter and receiver. Take a screenshot of both the transmitted message and the demodulated output. Be sure to expand the time base so that the waveforms can be clearly seen. Was the envelope detector in the AM receiver able to correctly demodulate the DSB-SC signal?

6. The phase synchronizer can also correct for modest frequency offsets. Use the DSB-SC transmitter and receiver, and offset the frequency of the transmitter by 10 Hz. Run the transmitter and receiver. Take a screenshot of the transmitted message, the unsynchronized demodulated output, and the synchronized demodulated output. Be sure to expand the time

---

[5] The demodulated output may be inverted. This is a consequence of squaring the signal in the phase synchronization process. An error of $\pm 2\pi$ in the angle 2θ is no error at all, but when the angle is divided by two, the error becomes $\pm\pi$.

base so that the waveforms can be clearly seen. Verify that the synchronized demodulated output is correct, except possibly for being inverted.

Repeat for frequency offsets of 100 Hz and 1 kHz. Can your phase synchronizer handle the 1 kHz case?

5.5 Report

Prelab

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

Submit your answers to the Questions at the end of the Prelab section.

Lab

Submit the program you created to implement the DSB-SC transmitter and receiver. Also submit any additional 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.

Describe the effect on the demodulated signal of the $\cos(\theta)$ term, as described in Step 4.

Submit the graphs required in Step 5. Discuss whether DSB-SC can be properly demodulated using an envelope detector.

Submit the graphs required in Step 6. Comment on whether your phase synchronizer was able to compensate for frequency offsets of 100 Hz and 1 kHz.