EE 4440 – Comm Theory Lab 5

Line Codes

Purpose:

The purpose of this lab is to investigate the properties of various line codes. Specific parameters investigated will be wave shape, bandwidth, and transparency. Simple pulse shaping will be investigated as well.

Note: For this lab, no formal lab report is due. Be sure to answer all questions in the lab (questions are in **BOLD** typeface). Please turn in the answers to the questions with your sketches. Typed reports are required for this lab. Be sure to label all sketches and keep the sketches relatively neat and clean.

Equipment:

This lab will utilize two pieces of the lab equipment. The first piece is the Agilent 54622D mixed-signal oscilloscope used in the first four labs. The other piece of equipment is the computer equipped with LabVIEW and the Data Acquisition Cards (DAQs). The computer will serve as the generator of the lines codes and also provide a spectrum analyzer to observe the frequency content of the line codes.

Background:

Line codes are used everyday to transmit digital data streams. Line codes are a method of defining high and low bits in a digital data stream for transmission. Each line code contains inherent properties that may make it a better solution given a set of conditions. These properties often are defined into transmission bandwidth, power efficiency, error detection and correction, timing content, and transparency. The line codes implemented in the lab are Polar, Bipolar, On-Off, and Manchester. For more information on these specific line codes, refer to Chapter 7 in the book.

Basic binary encoding uses square waves. Basic binary waveforms are generated using a simple square wave. To properly generate the binary line codes and to provide flexibility in the wave shapes, a different approach is necessary. First, the binary bits are encoded in an impulse train representing their logical '1' or '0' value. This impulse is then convolved with a 'basis waveform' to create the actual line code shape. The line codes in this lab are generated using this method. The primary pulse wave shape used for the lab is a normal rectangular pulse. When convolved, this roughly creates a square wave. The spectrum of a square wave continues on infinitely and has a very large bandwidth. Filtering the data or applying a different wave shape can reduce this bandwidth reduced to a level acceptable for data transmission. Such filtering or wave shape application is known as pulse shaping. To demonstrate pulse shaping, the second wave shape used in the lab is a Gaussian (Normal) pulse. When convolved, the resulting line code uses significantly less bandwidth than the rectangular pulse version. Other wave shapes could be used to better satisfy bandwidth and inter-symbol-interference (ISI) conditions.

Procedure and Questions:

Notes:

- Connect the oscilloscope to the DAC 0 channel (yellow wire) of the DAQ board for this lab
- Connect the Analog Channel 7 input (red wire) of the DAQ board to the same location as the DAC 0 channel. This will provide the input for the spectrum analyzer.
- Be patient with LabVIEW. Changing the line code will take several seconds to propagate through the software. When changing the line code, ensure Acquire ON is selected and wait 25 seconds before obtaining data.
- To stop the LabVIEW module, press the ESC key. It tends to work better than the Stop System button. (*Note: This completely stops the module, it doesn't just pause the data flow like Acquire ON*).
- To use the LabVIEW capture option, you may need to click the Acquire ONCE button a couple times before the file dialog opens (if the spectrum analyzer is stopped).
- Pulse shaping for the lab is designed around a 1 MSample/second sampling rate and 50 kbits/second data rate, so changing either of these values may result in erratic pulse-shaped waveforms.
- LabVIEW exports the spectral data with a Δf value first, followed by the actual spectral information (in dB). To properly plot this data in Excel (or another spreadsheet program), you will need to generate the frequency column.
- To observe the line codes in the time domain, it is often useful to stop the oscilloscope and then press the 'Single' key until an acceptable binary sequence is displayed on the oscilloscope screen. This acquires a single trigger's worth of data and then puts the oscilloscope back into stopped mode.
- An oscilloscope time base of 200 µs/div provides the same FFT sampling rate as the LabVIEW spectrum analyzer for a more direct comparison.

Part 1: Line code shape and bandwidth

Open the LabVIEW module "Lab 5.vi" on the desktop. Ensure that the DAQ settings are set to 1 MSa/sec for both input and output sampling frequency. Be sure the bit rate is set to 50 kbits/sec and Shaping is not enabled. Adjust the Spectrum Analyzer to a data length of 1024, an FFT size of 4096, a Hann (Von Hann, Hanning) window, and 25 averages. Also ensure the coupling button is set for DC coupling.

i) Run the LabVIEW module.

■ Set the oscilloscope time base to 200 µs/div.

■ Turn on the FFT function of the oscilloscope and adjust the display so the FFT span is 500 kHz and the center frequency is 250 kHz. This will set the oscilloscope's FFT function to display data in the same range as

LabVIEW.
■ Adjust the scaling and offset properties of the FFT if necessary.

ii) Start with the Polar RZ line code.
■ Adjust the horizontal time base on the oscilloscope to obtain at least 6 cycles (bits) of the waveform**.**
■ Stop the oscilloscope and sketch these six cycles. Be sure to label the corresponding voltages and bit pattern.

iii) Resume the oscilloscope and set the horizontal time base to 200 μs/div.
■ Observe the FFT spectrum on the oscilloscope and the output of the spectrum analyzer.
■ Sketch or capture the output of the spectrum analyzer. Be sure to label relative frequencies and amplitudes. To stop the spectrum analyzer, click the "Acquire ON" button to deselect it and then click the "Acquire ONCE" button to obtain one set of data.
■ From this stopped view, what is the estimated bandwidth of the Polar RZ line code (use –40 dB from the peak value)? If the value exceeds the display range, simply put the uppermost frequency.
■ What is the frequency of the first notch/zero crossing in the frequency spectrum?

iv) Repeat steps ii and iii for each line code (RZ and NRZ). Remember to wait for LabVIEW to propagate the new line code and be sure to change the Acquire setting back to 'Acquire ON' when changing line code types.

Part 2: Pulse shaping

Leave the settings in LabVIEW the same as in Part 1. Resume the LabVIEW module if it is not currently running.

i) Set the line code to Bipolar RZ (also known as AMI RZ).

ii) Observe the frequency content of the signal on both the oscilloscope and LabVIEW spectrum analyzer. (Make sure you are still using 200 μs/div for the time base on the oscilloscope). **Comment on any differences between the displays.**

iii) ■ Enable the pulse shaping function by toggling the "Shaping Enabled" switch. Allow LabVIEW time to adjust to the new display.
■ Freeze the spectrum analyzer display and sketch or capture the new spectrum. What is the estimated bandwidth of this pulse shaped line code (Use -20 dB from the peak amplitude as your bandwidth criterion)?

iv) **Sketch six cycles of the waveform on the oscilloscope display and label its bit pattern.**

v) Change the line code and observe the resulting spectrums with pulse shaping both enabled and disabled.
■ Sketch or capture the spectrum of at least one other line code and estimate the bandwidth of the signal.

■ **Sketch six cycles of the corresponding time domain signal and label its bit pattern.**

Part 3: Transparency

Ensure that pulse shaping is disabled initially. Again, leave the settings in LabVIEW the same as for the first section of the lab for now.

i) Set the line code to On-Off RZ to begin with.
ii) Change the binary string to a 4 character string.
iii) Observe spectrum and time domain signal on both the oscilloscope and spectrum analyzer for each line code (use the same sequence). Be sure to observe both the unshaped and shaped cases for each line code. Occasionally you may need to wait for the sequence of actual data characters to cycle through the visible line code. Ensure you are observing both the normal data and a long string of similar bits (0's) for each line code. **Which line code(s) provides the best transparency and why?**

Try different binary string values and different amplitudes with and without pulse shaping and observe the results. Any adjustments to the bit rate and sampling rate require the LabVIEW module to be stopped and restarted. Please note that the module does not give very satisfactory results for lower bit rates, so avoid decreasing the bit rate below 50 kbits/s. Also try to maintain the bit rate/sampling rate ratio (as far as integer relations, not an exact ratio since the DAQ boards do not function well above 1 MSa/sec) the same for best results.

For your lab report, answer the following additional questions:

i) Some transmission channels are AC-coupled. This means that DC components will not transmit well or at all along this type of channel. From your observations, which line codes would you expect to have a problem with this type of channel and why. Is decoding still possible with these line codes?

ii) Again from your observations, which line codes contain obvious timing content and why.

EE 4440 – Comm Theory – Prelab for Line Code Lab

The following questions are to be answered **before** coming to lab. The prelab will be due at the beginning of the lab period.

1) Given the choice of line codes for the lab (mentioned in the Background section) including the return to zero cases, which line codes will provide the best signal bandwidth and why?

2) Given your choices above, do these waveforms have any severe disadvantages?