# ***Lab #3 – PSK & Baseband Pulse Shaping***

In this lab you will construct digitally modulated waveform in MATLAB. The type of modulation used in this lab will be Phase Shift Keying (PSK), and you will use MATLAB to analyze the bandwidth of these signals. You will also investigate the effect of baseband pulse shaping on the signal bandwidth.

## ***Section 1: BPSK with Rectangular Baseband Pulse***

In this section of the lab, you are to construct a BPSK waveform with the following parameters:

Bit Rate = 1000 bits per second

Carrier Frequency = 10 kHz

Pulse Shaping = Rectangular

Assuming a rectangular pulse shape, **predict** the baseband and modulated frequency spectrum for this signal (ie. obtain a closed-form mathematical expression). Predict the bandwidth for both the baseband and modulated signals, with bandwidth defined as the frequency of the first null.

Using MATLAB, generate a random binary data pattern with polar signaling (ie. the signal can be either +1 or -1). You can use the built-in function ***randsrc*** for this purpose. This binary pattern will then be used to construct the baseband digital signal, with the baseband pulse shape being rectangular. You may wish to use the built-in function ***rectpulse*** to build the baseband digital signal.

Generate the frequency spectrum for the baseband signal in MATLAB (see the Appendix for hints on how to do this). Since you will be taking the Fourier transform of a pseudo-random binary pattern, the number of bits in the baseband digital must be statistically significant. In other words, when using the ***randsrc*** function to generate the binary signal, make sure you have more than a few bits.

Using the same baseband digital signal, modulate a sinusoidal carrier using the Binary Phase Shift Keying (BPSK) method. Generate the frequency spectrum for the modulated signal. Check that the spectrums of the baseband and modulated signal are in agreement with your prediction. Measure the attenuation of the first sidelobe relative to the main lobe. What are the issues of these sidelobes, in the context of a wireless communication system?

## ***Section 2: BPSK with Raised Cosine Pulse Shaping***

Use the same bit rate and carrier frequency as section 1, **predict** the bandwidth required for both the baseband and modulated signals (BPSK), but now using a raised cosine pulse with a rolloff factor of 0.5.

Using MATLAB, construct a baseband digital signal using raised cosine pulses. Again, use ***randsrc*** to generate a random binary pattern with polar signaling. You can then use the built-in function ***rcosflt*** to create a raised cosine filter, then shape the binary data with this filter. Plot the baseband signal with raised cosine pulse shaping in the *time domain* over several bit intervals. Next, generate the frequency spectrum of the baseband signal; does the bandwidth of this spectrum match your prediction?

Using the raised-cosine shaped baseband signal, modulate a sinusoidal carrier using BPSK. Generate the frequency spectrum for the modulated signal. Does the bandwidth of this spectrum agree with your prediction? How does the sidelobe of this spectrum compare with that using rectangular pulse shaping? How is this advantageous in wireless communication?

Next, compare the BPSK waveform for rectangular pulse shaping versus raised cosine pulse shaping. Specifically, compare the envelope variation between the two cases. Putting this signal through a real amplifier, which signal may be more suitable?

## ***Section 3: QPSK with Raised Cosine Pulse Shaping***

Using the same bit rate and carrier frequency as before, **predict** the bandwidth required for the modulated signal if Quadrature Phase Shift Keying (QPSK) is used. Recall that QPSK can be represented mathematically as

$$s(t) = \cos\left(\omega_c t + \theta(t)\right)$$

where $\theta(t)$ takes on discrete values such as $\{ 0, \pi/2, \pi, 3\pi/2 \}$ for QPSK. The amplitude of the carrier has been normalized to 1 for convenience. Note that QPSK carries 2 bits per symbol. In your bandwidth prediction, assume that raised cosine pulse shaping with a rolloff factor of 0.5 is used.

Using MATLAB, construct a QPSK signal with raised cosine pulse shaping. Generate the frequency spectrum for the modulated signal. Does the bandwidth of this spectrum agree with your prediction? Note: Some thinking and / or work will be needed to convert the shaped baseband pulses into a QPSK signal. Document your approach clearly in the lab report, with supporting mathematical equations.

For your lab report, hand in a copy of your commented MATLAB code, with plots of the various frequency spectrums and time domain waveforms for each section. Answer the questions posed in each section. Add further comments and thoughts of your own as part of your research.

**Hand in a lab report by no later than the beginning of the next lab.**

## Appendix – Frequency Domain Analysis with MATLAB

There are several ways to perform frequency analysis in MATLAB, and they all revolve around the Discrete Fourier Transform (DFT). Recall that the DFT of a discrete time sequence x[n] is given by

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi \cdot k \cdot n / N}$$

In MATLAB, the built-in function ***fft*** can be used to perform a DFT. Note that the result of the DFT, X[k], is just a bunch of samples in the frequency domain, and you will need to relate them back to their analog frequencies for this lab.

Another approach for performing frequency domain analysis in MATLAB is via the use of *spectral estimators* (type *Spectral Analysis* in MATLAB product help for more detail). The basis for spectral estimators is still the DFT, with different windowing and algorithms depending on your needs. Since you are dealing with well-defined signals in this lab, I would suggest you explore the *Periodogram* spectral estimator. The following commands can be used to invoke this spectral estimator:

```
specScope = spectrum.periodogram;
specOpts=psdopts(specScope);
set(specOpts, 'Fs', fs, 'SpectrumType','twosided','CenterDC',true);
figure(1)
psd(specScope, x_n, specOpts);
```