**EE 445S Real-Time Digital Signal Processing Laboratory – Prof. Brian L. Evans**

**Lab 6 - Part 2 - QAM/QPSK Receiver**

This week, we will add code to simulate a QAM receiver to our QAM transmitter. In a real communication system, we would send the QAM signal over the channel to another DSP board. In this case, we simulate the transmitter and receiver on the same DSP board. This vastly simplifies receiver, since we don’t need to perform carrier recovery, symbol clock recovery, automatic gain control, etc.

**Part 1: Setup CCS Project (Same as last week):**

* Create a new project with the source files in C:/CD/code/chapter\_18/ccs/QPSK\_Tx
* Use the file “impulseModulatedQPSK\_ISRs.c”. (Exclude the file “impulseModulatedQPSK\_ISRs\_revA.c” from build for this task).
* Run the program and ensure that it works.

**Part 2: Modify the transmitter**

* Design your own raised cosine filter using *rcosine* in MATLAB:

Input symbol rate: 2400 Hz (fsym, Fd in MATLAB)

Output sample rate: 48 kHz (fs, Fs in MATLAB)

Type of filter: fir/normal

Alpha (the excess bandwidth factor): 0.8 (alpha, r in MATLAB)

Truncation limit: 4 symbol periods (delay in MATLAB)

* Export the coefficients into the source file “coeff.c” in the CCS project.
* **Note that the length of the filter is different from that in the original code. Modify the code accordingly.**
* Change the QAM/QPSK carrier frequency from 12KHz to 8KHz. You will have to modify the look-up tables for the new carrier wave. You will also have to modify modulation step where the look-up table is used. *Hint: The original table is length 4. The new table is length 6.*
* Run the program and ensure that it works. Check that the carrier frequency is correct.

**Part 3: Add your own receiver code *(Refer to Steven Tretter’s slide 13-18)***

* Add code to demodulate the in-phase component of the QAM signal:
  1. Multiply by the carrier and scale by 2 (*Why?).*
  2. Design an IIR biquad in MATLAB to extract the baseband signal. You might use a Butterworth filter. You will have to choose an appropriate cutoff frequency.
  3. Implement your biquad in C. Make sure to retain the full precision of the filter coefficients. Since this filter has feedback, low precision coefficients may lead to instabilities.

**Part 4: Test**

* Send your recovered in-phase signal to one channel of the codec. Send the original in-phase signal (prior to modulation by the carrier, but after the pulseshaping) to the second channel of the codec.
* Do they match? Provide an oscilloscope screenshot.
* Now demodulate the quadrature component of the QAM signal. Now you will have to invert the sign of the carrier (*Why?*).  Send your recovered quadrature component to one channel of the codec. Send the original, pulseshaped quadrature signal to the other channel of the codec.
* Do they match? Provide an oscilloscope screenshot.

**Question:**

1. Explain why we changed the carrier frequency to properly simulate the receiver.
2. Explain your choice of cutoff frequency for the demodulation filter.