Lab 5
Baseband PAM
EE 445S

Enoc Balderas   Daniel Diamont

April 17, 2019

Date Performed: March 1, 2019
Instructor: Professor Evans

1 Introduction

In this lab, we investigated aspects of pulse amplitude modulation (PAM) for baseband transmission and reception. The lab was split up into the following three parts: pulse shaping, symbol clock recovery, and demodulation.

1.1 Pulse Shaping

Pulse shaping refers in this case to applying a band-limited pulse to an impulse train of symbol amplitudes before sending the signal to the DAC. In this lab, we explored the effects of using differently parametrized raised cosine pulses on the time domain representation of the signal, as well as the effects on quantization. Specifically, since the raised cosine pulse shape with $\beta = 1$ has most of its power in the time domain inside of of the symbol period, then it follows that using the raised cosine to shape successive symbol amplitudes causes less inter symbol interference (ISI). In contrast, using a raised cosine with $\beta = 0.125$, which has more power leakage across time with respect to the previous example, then it follows that there will be more inter symbol interference. We can see the difference in ISI by comparing the eye diagrams in Figure 3 and Figure 4. It is clear that we would be able correctly quantize with a higher probability data using a raised cosine with $\beta = 1$ as opposed to data using raised cosine with $\beta = 0.125$.

1.2 SCR

1.3 Demodulation

2 Methods

3 Results

3.1 Pulse Shaping

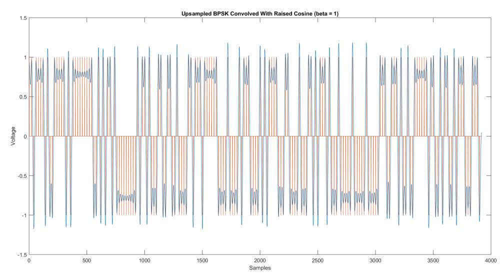


Figure 1: Pulse shape BPSK $\beta = 1$.

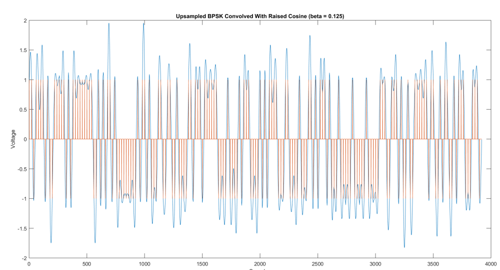


Figure 2: Pulse shape BPSK $\beta = 0.125$.

1. Explain the major differences between the two filters with respect to their A) Magnitude responses B) Impulse responses

   Answer: A) The main difference between the magnitude response is that the $\beta = 0.125$ filter has a much lower bandwidth than the $\beta = 1$ filter.

   B) The main difference between the impulse response is that the $\beta = 0.125$ filter side lobes decay at a slower rate than the $\beta = 1$ filter side lobes.

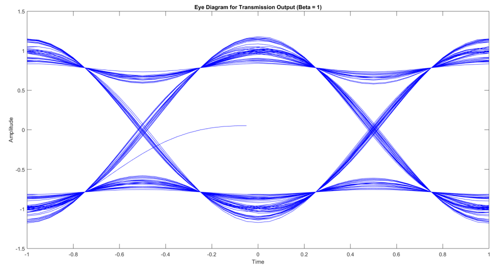


Figure 3: Eye diagram $\beta = 1$

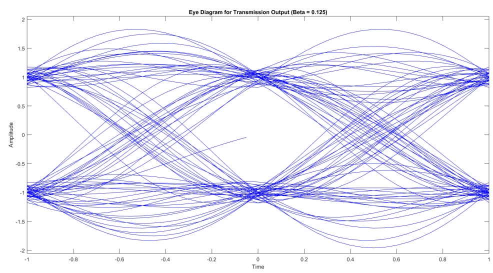


Figure 4: Eye diagram $\beta = 0.125$

2. What is the width of the impulse response for the $\beta = 0.125$ case?

   Answer:

3. How would you obtain this number theoretically? (Hint: Look at the fsym and truncation limit you set)

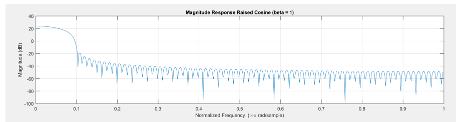
   Answer:

3.2 Symbol Clock Recovery

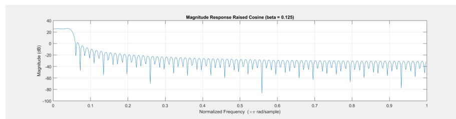
Prefilter:

b0	1
b1	0
b2	-1
-a1	1.96004
-a2	-0.984414

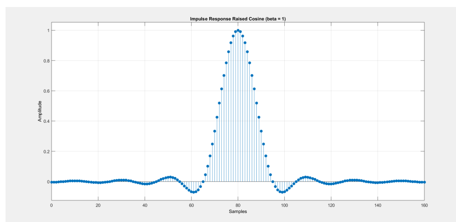
Bandpass:



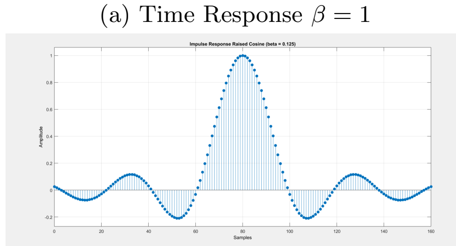
(a) Magnitude Response $\beta = 1$



(b) Magnitude Response $\beta = 0.125$

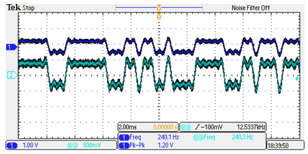


(a) Time Response $\beta = 1$

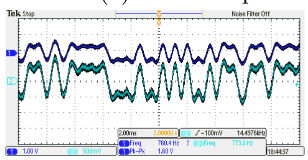


(b) Time Response $\beta = 0.125$

b0	1
b1	0
b2	-1
-a1	1.87293
-a2	-0.969067



(a) Time Response $\beta = 1$



(b) Time Response $\beta = 0.125$

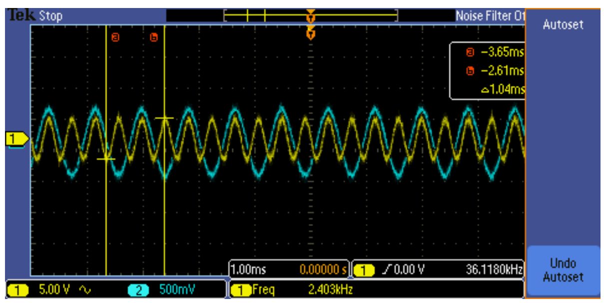


Figure 8: Dotting sequence

Code:

```
if (counter == 0) {
symbol = SSRG_update(&SSRG_state); // pseudo random m-sequence
x[0] = data[symbol]; // read the table

/* dotting sequence
symbol = symbol ^ 1;
x[0] = data[symbol]; // read the table
*/
}

// perform impulse modulation based on the FIR filter, B[N]

y  = 0;

for (i = 0; i < 8; i++) {
y +=  x[i]*B[counter + 20*i]; // perform the dot-product
}
```

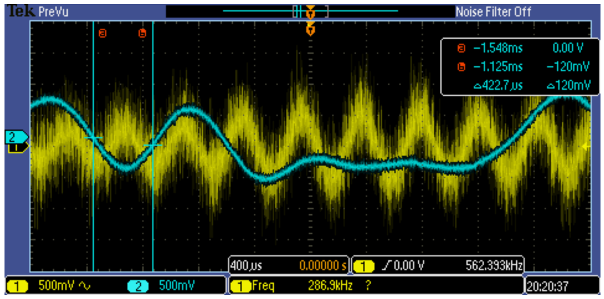


Figure 9: 2-PAM SCR

```
if (counter == (samplesPerSymbol - 1)) {
counter = -1;

/* shift x[] in preparation for the next symbol */

for (i = 9; i > 0; i--) {
x[i] = x[i - 1];           // setup x[] for the next input
}
}

counter++;

output = y;
scr = clock_recover(y);
```

Table:

pattern	frequency	amplitude
dotted sequence	2381 Hz	1.04 V
dotted sequence	2323 Hz	1.46 V

LPF:

b0	1
b1	1
b2	0
-a1	0.509525
-a2	0

Code:

```
  if (counter == 0) {
symbol = SSRG_update(&SSRG_state); // a faster version of rand() % 2
```

```
x[0] = data[symbol]; // read the table
}

  // perform impulse modulation based on the FIR filter, B[N]
  y  = 0;

  for (i = 0; i < 8; i++) {
y +=  x[i]*B[counter + 20*i]; // perform the dot-product
}

  if (counter == (samplesPerSymbol - 1)) {
    counter = -1;

/* shift x[] in preparation for the next symbol */
  for (i = 9; i > 0; i--) {
x[i] = x[i - 1];          // setup x[] for the next input
}
  }

  counter++;

output = y*cosine[counter & 3];

//receiver
demod = 2*output*cosine[counter & 3];

//LPF
biquad_x[0][0] = demod;

CodecDataOut.Channel[LEFT]  = y; // setup the LEFT  value
CodecDataOut.Channel[RIGHT] = G[0] * biquad(0, biquad_x[0][0]); // setup the RIGHT value
```

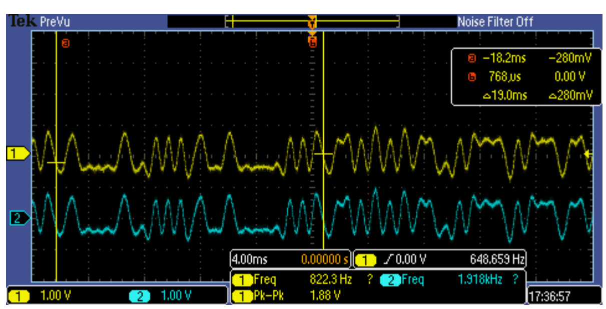


Figure 10: transmit and recieve

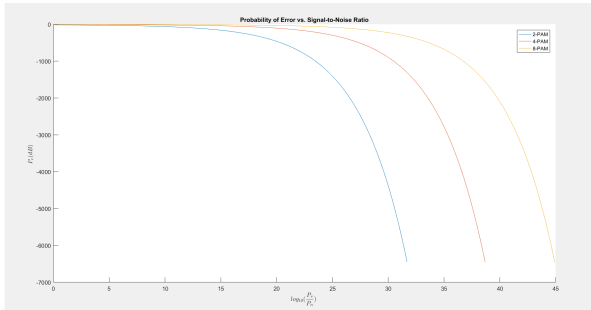


Figure 11: PES

4 Discussion

5 Answers to questions

1. How could pulse shaping be implemented using only a single "filter" (not a bank of filters). Practically, why would this be undesirable?

   Answer: Pulse shaping could be implemented by convolving with a single FIR filter. This is undesirable because for baseband transmission we usually take our symbol amplitude vector and upsample (by some fator L) to the DAC frequency before pulse shaping. Because of this upsampling, the amplitude vector contains a lot of extra zeros, which would cause (L - 1) multiplications by 0 for every original symbol amplitude. By using a filter bank, we avoid these fruitless multiplications by 0, and obtain a factor of L performance improvement.

2. In the clock recovery system, discuss the need for a Prefiltering. For a symbol rate of 2 kHz, what would be the output if the prefilter attenuated all frequencies greater than 900 Hz? Would it be possible to recover the transmitter's symbol frequency (using the same squaring operation and post-filters as in the lab)? If not, give a short reason why.

   Answer: We prefilter to obtain a cosine with a center frequency of $fsym/2$ without high frequency noise. What we are interested in is capturing the phase (θ) and frequency information (fsym) of the cosine to feed this into a Costas or Phase Lock Loop to determine the phase offset for sampling. Any other frequency content is unnecessary.

   If the output of the prefilter attenuated all frequencies greater than 900 Hz, we would not be able to recover the transmitter's symbol frequency because $fsym/2$ in this case equals 1 KHz, which would be attenuated by our hypothetical prefilter.