UMass Dartmouth
COLLEGE OF ENGINEERING

Electrical and Computer
Engineering Department

ECE 403/591: Special Topcis in ECE: Software Defined Radio

Midterm Project - Baseband Digital Communication Transceiver Design

1 Objective

The objective of this project is to simulate the digital baseband communication system. Specifically, we will generate the transmitted baseband signal using different kinds of pulse shaping technologies. Then we will study the spectrum of transmitted signal to investigate the frequency leak and bandwidth efficiency of different pulse shaping. Meanwhile, we will decode the digital data set out from the transmitter.

2 Part 1: Transmitter Design (25 points)

In this part, we use Matlab to generate a baseband transmission signal, including on-off keying, 2-PAM, 4-PAM, 8-PAM, with different pulse shaping technologies.

First of all, we will allow a user to input anything from the keyboard, and the input will be translated into digital data using 7-bit ASCII code. These 0s and 1s are our data (in bit): $b[i]$. Next, the digital data will be encoded to symbols ($S[i]$), based on which baseband modulation scheme you use: on-off keying, 2-PAM or something else.

Using the encoded symbol sequence, we will use a pulse generator $p(t)$ to generate the transmitted baseband signal $y(t)$. The pulse generator $p(t)$ will generate a time-limited pulse, then we will have to delay this pulse at time delay τ to modulate the symbols, as shown in next equation:

$$y(t) = \sum_i S[i] \cdot p(t - iT - \tau) \tag{1}$$

Rectangular pulse is the simplest pulse shaping technology, but it has very bad frequency spectrum and very low bandwidth efficiency.

The Matlab script example given in this handout uses rectangular pulse shape and 2-PAM to generate the transmitted signal.

Modify the 2-PAM with rectangular pulse code so we can:

(1) Include a random initial time delay τ at transmitter side;

(2) Use on-off keying, 2-PAM, 4-PAM (-3:00, -1:01, 1:11, and 3:10), and 8-PAM (-7:000, -5:001, -3:011, -1:010, 1:110, 3:111, 5:101, and 7:100) as modulation scheme;

(3) Use Hamming pulse shape to generate the baseband transmitted signal, investigate its spectrum, compare it to the spectrum of using rectangular pulse;

(4) Use Hanning pulse shape to generate the baseband transmitted signal, investigate its spectrum, compare it to the spectrum of using rectangular pulse;

(5) Use truncated Sinc pulse shape (with 10 sidelobes, 5 sidelobes on each side) to generate the baseband transmitted signal, investigate its spectrum, compare it to the spectrum of using rectangular pulse.

Figure 1 and Figure 2 demonstrate the time domain signal and frequency domain spectrum of a baseband transmission using 2-PAM and rectangular pulse shape of text "*hello world*".

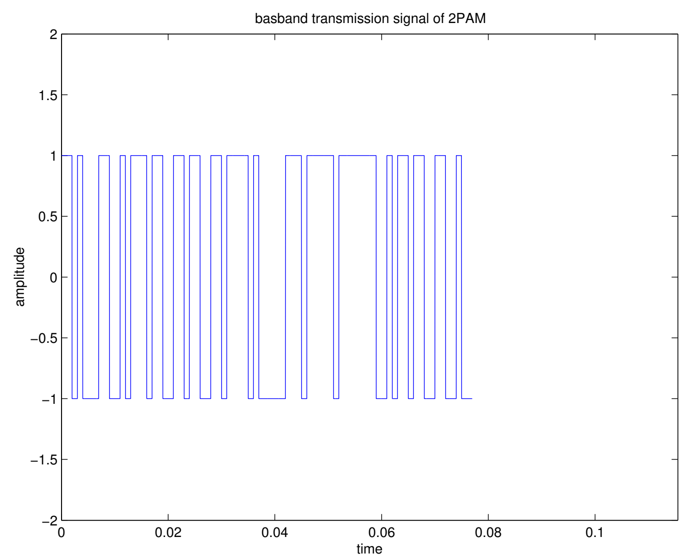

Figure 1: Transmitted baseband signal using 2-PAM and rectangular pulse shape

3 Part 2: Receiver Design (25 points)

When we have the received signal in time domain, we have all the parameters we need to build the ideal receiver.

Figure 3 illustrates the diagram of an ideal baseband receiver.

The first part of the receiver is a sampler. As we have studied in class already, when rectangular, Hamming or Hanning pulse shapes are used, the optimal sampling time is:

$$t_i = \tau + \delta + (i-1) * T + T/2 \tag{2}$$

Now, with all the parameters available, we can write a simple Matlab program to fetch the right samples of the received signal. Let's call these samples $r[i]$.

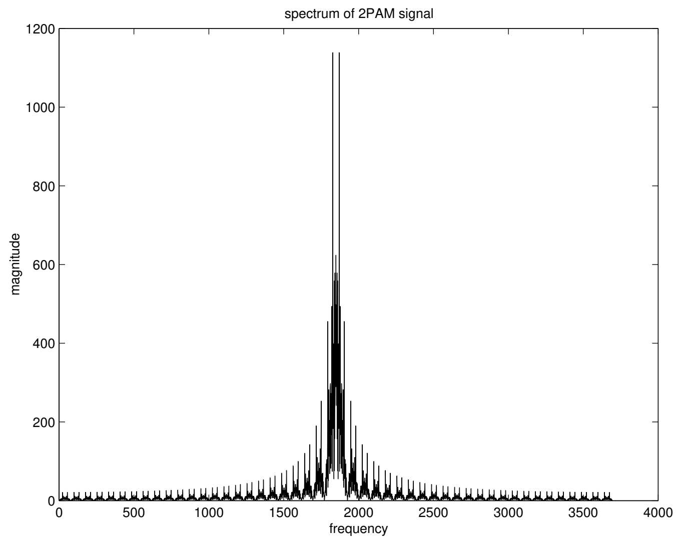

Figure 2: Spectrum of baseband signal using 2-PAM and rectangular pulse shape

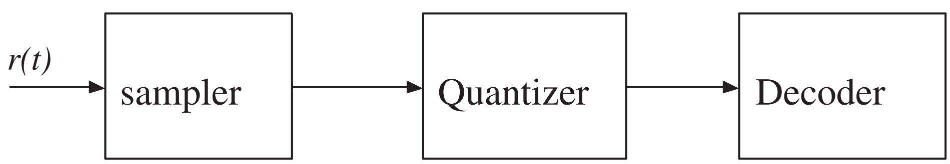

Figure 3: Ideal Baseband Receiver

The second part of the receiver is a quantizer. The quantizer can be divided into two steps. First, we need to compensate the transmission attenuation. That is, we need to perform the following operation:

$$r'[i] = r[i]/g \tag{3}$$

The second step is to convert each and every sample to its closest possible value. For example, in 2-PAM, only $+1$ and -1 are possible transmitted values. So, if one sample is 0.78, we compare it with $+1$ and -1 and find that $+1$ is closer to 0.78 than -1, so we convert this value to $+1$. Mathematically, this corresponds to:

$$d[i] = \begin{cases} 1 & if r'[i] > 0 \\ -1 & otherwise \end{cases} \tag{4}$$

After quantizer, we need a decoder to undo what the encoder did at transmitter side. That is, we need to convert all the received symbols back to data bits, 0s and 1s. This can be done by applying the reverse function of the encoder.

Finally, using the 0 and 1 sequence, cut them in group of 7 bits, convert each and every 7 bits to one ASCII code. (Hint: Matlab has functions to do this.)

In this part, you need to combine your transmitter and receiver to have a complete baseband communication system. You need to:

(1) Write a Matlab program to perform the quantizer and the decoder of the receiver.

(2) Implement an ideal baseband receiver, decode your transmitted data, and print the decoded message in Matlab command window.

4 Part 3: Decode the Given Data (30 points)

You are given a trans_signal.mat transmitted data file. It contains the transmitted baseband signal after transmission over a noisy channel. MAT is the standard file structure in Matlab to store data. Using *LOAD* command in Matlab, you can open the file and fetch the stored data into the computer's memory. In your Matlab command window, type in *LOAD trans_signal.mat*. Then type in *WHO*. You will see

Your variables are:
fs gain modulation pulseshape signal symbolrate tau delta

In these variables, signal is an array representing the received signal. Then, you can use PLOT command in Matlab to draw the received signal in time domain. Type in *plot(signal)*, you will see the signal in time domain:

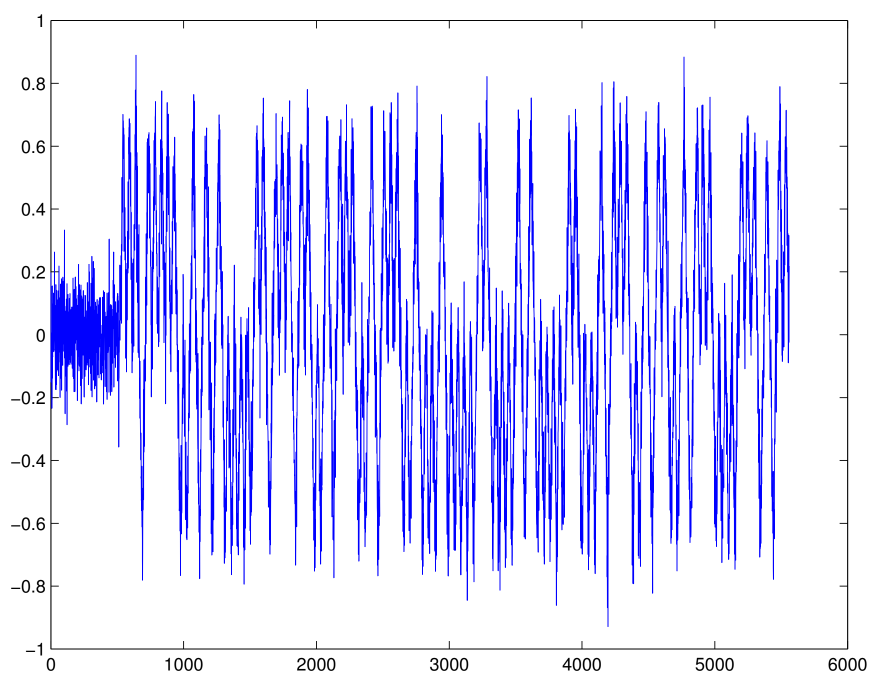

Figure 4: plotted received baseband signal

One MAT file can contain more than one data entry. The trans_signal.mat actually contains another important parameter we need. This parameter is the sampling frequency

f_s. Using $T_s = 1/f_s$, we can derive the sampling duration and based on this, we can plot the received signal in correct time scale.

Remember, the baseband receiver needs to know what the format the transmitter used to be able to decode the information successfully. The format includes:

(1) What's the modulation scheme employed at transmitter, e.g., is it on-off keying, 2-PAM, 4-PAM, 8-PAM or some other schemes;

(2) What's the symbol duration T and correspondingly, in our program, how many samples one symbol duration occupies?

(3) What's the pulse shape used at transmitter? Is it rectangular, Hamming, Hanning, sinc, or some other shape?

(4) What's the initial time delay τ and the transmission delay δ?

(5) What's the attenuation (or gain, g) of the transmission channel?

The given trans_signal.mat transmitted data file uses the following numbers for indicating modulations and pulse shaping:

(1) modulation scheme is defined by variable *modulation*:

- modulation=1: On-off keying
- modulation=2: 2-PAM
- modulation=4: 4-PAM (-3:00, -1:01, 1:11, and 3:10)
- modulation=8: 8-PAM (-7:000, -5:001, -3:011, -1:010, 1:110, 3:111, 5:101, and 7:100)

(2) variable *symbolrate* defines how many symbols we transmit per second. From this, we can derive what's the T and how many samples one symbol occupies.

(3) the pulse shape is defined by variable *pulseshape*:

- pulseshape=1: Rectangular
- pulseshape=2: Hamming
- pulseshape=3: Hanning
- pulseshape=4: Sinc (with 10 sidelobes, 5 sidelobes on each side)

(4) the initial time delay is defined by *tau*, while transmission delay is defined by *delta*. Both of them are in seconds. So you need to convert them into how many samples to get the correct sampling location later.

(5) the variable g represents the attenuation or gain factor the channel did on the signal. In other words, without noise, the amplitude is not $+1$, -1 but $+g$, $-g$ now.

In this task, you need to

(1) Find out the correct time scale, plot the received signal with correct index.

(2) Find out the correct sampling time.

(3) Decode the data.

5 Report (20 points)

Submit a team report. Name your file Midterm_LastName1_LastName2.pdf, Midterm_Transceiver_LastName1_Last
and Midterm_Decoder_LastName1_LastName2.m.

```
% ECE403/591
% Baseband transmission using 2-PAM
clear all;
close all;

fs = 48000; % sampling frequency
Ts = 1/fs; % sampling duration
symbolrate = 1000; % transmitted pulses/second should be an integer divisor of fs
sps = fs/symbolrate; % number of samples in one symbol

%% Transmitter setup
% define the basic rectangular pulse shape
pulse = ones(1,sps); % 1 pulse

data = [1 0 1 1 0 0 1 1 1 0 0 0 1 1 0 1 1 0];
[number, length0] = size(data);
symbol = 2*data-1;

for i = 1: length0
    for j=1:sps
        signal((i-1)*sps + j) = symbol(i) * pulse(j);
    end
end

index = 0: Ts: (length0*sps-1)*Ts;
freq=fft(signal);
freq=fftshift(freq);
%%
figure(1);
plot(index, signal);
axis([0 1.5*length0*sps*Ts -2 2]);
xlabel('time');
ylabel('amplitude');
title('basband transmission signal of 2PAM');
figure(2);
plot(abs(freq));
xlabel('frequency');
ylabel('magnitude');
title('spectrum of 2PAM signal');
```