

EENG 373: DIGITAL COMMUNICATIONS

EXPERIMENT NO. 3

BASEBAND DIGITAL TRANSMISSION

Objective

This experiment investigates how a digital data stream can be encoded into a sequence of pulses for transmission through a baseband analog channel. Specifically, you will examine:

- various line coding methods used for digital baseband modulation in data communication applications;
- spectral properties of those lines codes, i.e. their power spectral densities;
- two important causes of signal distortion in a data communication channel: additive gaussian noise and filtering effects due to the bandwidth of the channel;

Laboratory Procedure

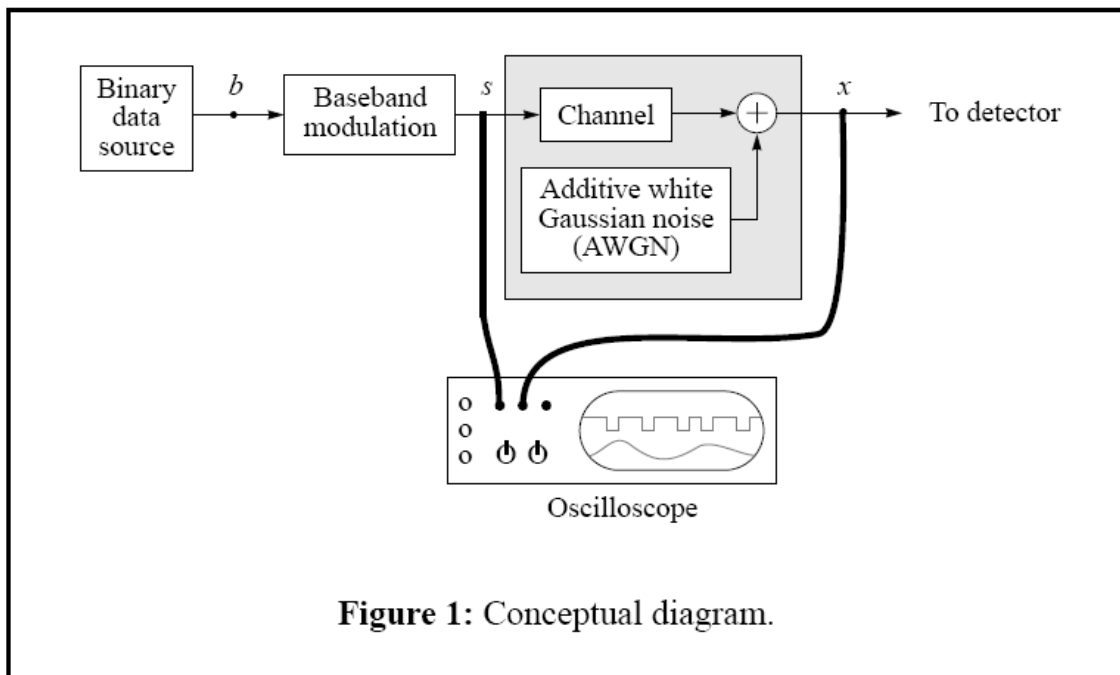


Figure 1: Conceptual diagram.

1. Line codes for binary signaling over baseband channels

Binary digits produced by a data source can be serially encoded using a variety of signaling formats called *line codes* for transmission over an analog baseband channel. Their structure and spectral properties will be examined in this section. You will be using the modul custom MATLAB command:

```
output = modul(<binary_sequence>,<line_code_name>,<Fd>,<Fs>);
```

where F_d is the data rate in symbols per second (bauds) and F_s is the simulation sampling frequency. The action of the command is to take the vector `binary_sequence` and modulate it according to the specified line code. The following codes are supported: `'unipolar_nrz'`, `'bipolar_nrz'`, `'bipolar_rz'`, `'ami'` and `'manchester'`.

Many of the custom MATLAB commands (`scope`, `modul` etc.) require the F_d and F_s parameters.

1.1 First, create this binary sequence:

```
>>b = [1 0 1 0 0 0 1 1 1 1];
```

Then, generate the coded signal representing `b` using the `modul` command with a data rate of $F_d = 1$ kb/s, a sampling rate $F_s = 10$ kb/s and using the unipolar NRZ line code:

```
>> Fs = 10000;  
>> Fd = 1000;  
>> T = 1/Fd;  
>> s = modul(b,'unipolar_nrz', Fd, Fs);
```

Display the resulting signal:

```
>> scope(s,'one-shot',10*T);
```

Repeat these steps above for the following line codes:

- bipolar NRZ;
- bipolar RZ;
- manchester;
- AMI.

Result: (Print the graphs)

1.2 Power spectral densities of the line codes

We will now examine the power spectral densities of the line codes. Generate a random binary sequence of 1000 bits:

```
>> b = round(rand(1,1000));
```

Display the PSD function¹ of this sequence after it has been encoded using unipolar NRZ:

```
>> [Ss,f] = psd(modul(b,'unipolar_nrz', Fd, Fs),10000,Fs,'mean');
```

```
>> plot(f,Ss);
```

The resulting plot is what you would observe by plugging a spectrum analyzer operating in linear scale at point s in the conceptual diagram of Figure 1.

To locate peaks and nulls, it is sometimes helpful to also examine the PSD in logarithmic scale:

```
>> plot(f,10*log10(Ss));
```

By observing the psd plot, identify the location on the frequency axis of the first and second spectral peaks (denoted by f_{p1} and f_{p2}) and the first and second spectral nulls (f_{n1} and f_{n2}). Please note that while the first spectral peak can be situated at $f=0$, we are only interested in spectral nulls for $f > 0$, i.e. a null at $f=0$ doesn't count. The location of the peaks and nulls should be entered in the table below. Repeat the operation for all the line codes studied in part 1.1 and record your results in Table 1.

Table 1: Spectral characteristics of line codes at 1kb/s data rate.

line code	f_{p1} [Hz]	f_{p2} [Hz]	f_{n1} [Hz]	f_{n2} [Hz]	W [Hz]
unipolar NRZ					
bipolar NRZ					
bipolar RZ					
manchester					
AMI					

Result: (Print the graphs and record the results in Table 1)

1. The number 10000 is passed as an argument to the psd command; it represents the number of samples in the sequence, i.e. 1000 bits \times F_d . If you use a shorter sequence length due to the student edition limitation, adjust this parameter accordingly.

1.3 Examine the relationship between the PSD and the data rate:

Let us now examine the relationship between the PSD and the data rate F_d . Use Manchester coding and produce a PSD for different values of F_d :

```
>> [Ss,f] = psd(modul(b,'manchester',<data_rate>, Fs),10000*Fs/<data_rate>,Fs,'mean');  
>> plot(f,Ss);
```

where `<data_rate>` is successively taken to be 250, 500 and 1000 b/s. Observe the displacement of the spectral nulls and peaks and relate it to the binary data rate.

Result: (Write the MATLAB Codes ,print the graphs and deduce the bandwidth values)

2. Simulating a baseband channel with noise

You will now simulate the characteristics of an ideal baseband communication channel including additive white Gaussian noise. The channel is modelled as an ideal low-pass filter whose output is summed with the output of a white Gaussian noise generator (see conceptual diagram of Figure 1). This channel model is implemented as the custom MATLAB function `bbchannel`:

```
out = bbchannel(<input>,<gain>,<noise_power>,<bandwidth>,<Fs>);
```

2.1 Create a 10 bit binary sequence `b` and encode it into an analog waveform using bipolar NRZ.

```
>>b = [0 1 0 0 1 0 0 1 1 0];  
>> s = modul(b,'bipolar_nrz', Fd, Fs);
```

From your observation in part 1, determine the transmission bandwidth W of `s`:

```
W = [ ] [Hz]
```

2.2 Transmit the waveform `s` over a baseband channel with unity gain and additive white Gaussian noise (AWGN) where the noise power is $0.01 W$ and the channel bandwidth is 4.9 kHz. Display the channel input and output:

```
>> x = bbchannel(s,1,0.01,4900);  
>> subplot(211), scope(s,'one-shot',0.01);  
>> subplot(212), scope(x,'one-shot',0.01);
```

Has the information been preserved by this channel? In other words, could you estimate visually the sequence that was transmitted with knowledge of the output only?

```
Yes or NO
```

2.3 To observe the effect of noise on the transmitted signal, you will gradually increase the channel noise power while keeping the bandwidth constant at 4.9 kHz.

```
>> clf; subplot(321), scope(s,'one-shot',0.01);  
>> title('Original signal');  
>> subplot(32<sub_win>), scope(bbchannel(s,1,<noise_power>,4900),'one-shot',0.01);  
>> title('noise power = <noise_power> W');
```

where you will use successively in lieu of noise_power values in the {0.1, 0.5, 1, 2, 5} set, and you will plot each signal in a different subwindow. Therefore, <sub_win> will take on values

in the set {2, 3, 4, 5, 6}.

Result: (Write the MATLAB Codes and print this graph)

At which noise power level does the channel output become undistinguishable from noise?

$\sigma_n^2 =$	[W]
----------------	-----

It is also possible to observe the effect of increasing the noise power by plotting the PSD of the channel output. First, produce a plot of the PSD of the modulated signal s:

```
>> b = round(rand(1,1000));  
>> s = modul(b,'bipolar_nrz', Fd, Fs);  
>> [Ss,f] = psd(s,length(s),Fs,'mean');  
>> clf, subplot(221), plot(f,Ss), a = axis;  
>> title('PSD of original signal');
```

Then, produce a series of PSDs corresponding to the channel output with noise power levels {0.01, 1, 5} and plot them in a separate subwindow. The <sub_win> variable will take on values in the set {2, 3, 4}.

```
>> [Sx,f] = psd(bbchannel(s,1,<noise_power>,4900),length(s), Fs,'mean');  
>> subplot(22<sub_win>), plot(f,Sx); axis(a);  
>> title('PSD of received signal; noise power = <noise_power>W');
```

Result: (Write the MATLAB Codes and print this graph)

2.4 We will now look at the effects of channel bandwidth on the transmitted signal. The channel we are studying in this experiment is an ideal low-pass filter and it introduces distortion by cutting off the higher portion of the signal's spectrum. To study this effect, set the noise power to zero and generate output waveforms for channel bandwidths of {4900, 3000, 2000, 1000, 500} Hz :

```
>> clf;  
>>b = [0 1 0 0 1 0 0 1 1 0];  
>> s = modul(b,'bipolar_nrz', Fd, Fs);
```

```
>> subplot(611), scope(s,'one-shot',10*T);  
>> subplot(612), scope(bbchannel(s,1,0,4900),'one-shot', 10*T);  
>> subplot(613), scope(bbchannel(s,1,0,3000),'one-shot', 10*T);  
  
and so on for the other bandwidth values ...
```

You should now have a series of 6 plots with the top plot being the original transmitted waveform and the subsequent plots showing increasing levels of distortion.

Result: (Write the MATLAB Codes and print this graph)

Questions

- Q1.** Assuming unit pulse amplitude and binary data rate $R_b = 1 \text{ kb/s}$, plot the waveforms representing the binary sequence $b = \{ 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1 \}$ using the following line codes:
- a) unipolar NRZ1 (on-off signaling);
 - b) binary antipodal signaling (bipolar NRZ);
 - c) Alternate Mark Inversion (AMI)
 - d) Manchester (split-phase);
- Q2.** Determine analytically and sketch the power spectral density functions for each of the above line codes. Use a bit rate of $F_d = 1 \text{ kb/s}$.
- Q3.** Determine the bandwidth for the line codes in question 1 as a function of F_d .