

## **SIGNALS AND SYSTEMS LABORATORY 13:**

### **Digital Communication**

#### **INTRODUCTION**

*Digital Communication* refers to the transmission of binary, or digital, information over analog channels. In this laboratory you will gain an appreciation of the two main factors that limit reliable communication, namely *bandwidth* and *noise*.

#### **COMMUNICATION SYSTEMS**

The use of man-made devices to enable long distance communication has a history extending at least as far back as the written word, even to the use of petroglyphs on canyon walls. The use of electrical devices to aid communication dates back to the demonstration of a viable telegraph system by Samuel Morse in 1838, although others may have had similar ideas a few years earlier. The technology for communication has developed explosively since that time. The way we think about communication and information theory today dates from two historic papers by Claude Shannon published in the *Bell System Technical Journal* in July and October of 1948. These papers, and some supplementary material, were published in book form in 1949 as *The Mathematical Theory of Communication*, University of Illinois Press. This book is well written and, unlike many of the thousands of papers which followed, very readable.

Shannon introduced several ideas. One important idea was to break the communication problem down into two stages called *source coding* and *channel coding*. For present purposes let us decompose the communication process even further into the following sequence of operations.

I. *Source Coding*. A natural source (thoughts, text, music, pictures, line drawings) is coded into binary form. The number of bits required (for a given accuracy) is called the *information content* of the message. Analog sources cannot be represented exactly with a finite number of bits. One must agree on a standard for error. For example, the hifi audio standard is 16 bits per sample. This produces a maximum quantizing error of  $2^{-15}$  times the maximum amplitude of the signal. Other natural sources may involve more clever ways of coding. Shannon studied sources that produce information continuously in time, and considered the problem of finding the most efficient representation based on the statistical structure of the source. This results in the fewest possible bits per second (on average), which is called the *entropy* of the source. Programs that do *file compression* are source coders. The Morse code (invented by Samuel Morse) is a source coder. It is a variable-length coder that uses a short code for frequently occurring letters and a long code for infrequently used letters. One can say that the intent of a source coder is to remove redundancy and find an efficient representation of the message. The output of a source coder is a sequence of binary digits, zeros and ones.

II. *Channel Coding*. A channel coder takes a binary sequence and produces a signal which is appropriate for a given communication channel. In general it will add redundancy to defeat errors due to noise. The interleaved Reed-Solomon code that is used in the compact disk medium is a channel code. Provided it is not done too often, one can wipe out individual bits, and even whole blocks of bits, and the original information can still be restored because of the constraints the channel code imposes on its output. This cannot be done without cost; the channel code increases the total number of bits by a fixed percentage. The *rate* of a channel code is the ratio of the number of input bits to the number of output bits and will be less than one. The output sequence is still binary. Shannon studied the problem of characterizing the highest possible rate among all channel coders, which can lead to decoding with arbitrarily small probability of error. The solution to this problem characterizes the channel *capacity* in bits per second.

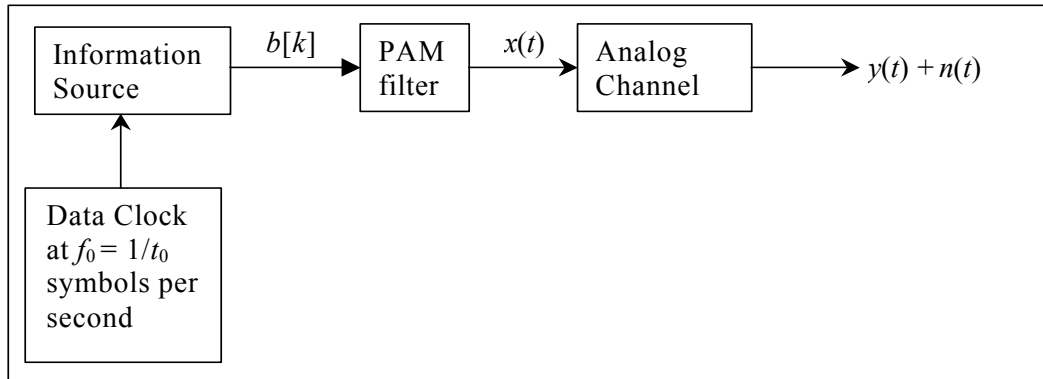
III. *Coding for an analog channel*. If we must use an analog channel to convey the information to a destination, then the binary input must be converted to a real valued continuous time signal before it is transmitted. There are very many ways to do this. As an example we shall consider a pulse amplitude modulation (PAM) system. Here one transmits a pulse  $p(t)$  for a *binary one* and a pulse  $-p(t)$  for a *binary zero*. The pulse shape may be designed to match the characteristics of the channel.

IV. *The channel.* Analog baseband channels are usually modeled as a lowpass filter with additive white Gaussian noise. The channel is characterized by its bandwidth and noise power.

The rest of the blocks simply undo what has been done, and consist of a receiver which makes a best guess as to what the binary input sequence to the channel might have been, followed by a decoder for the channel coder (which corrects errors), followed again by a decoder for the source coder.

### A SIMPLE COMMUNICATION SYSTEM MODEL

Our model for an analog channel is shown in Figure One. It consists of an additive noise source followed by a lowpass filter. This filter might not be ideal. Its magnitude might not be constant and its phase might not be linear.



**Figure One** A digital communication system

The PAM transmitter which produces the continuous time channel input  $x(t)$  is described by the following relations in time and frequency.

$$(1) \quad p(t) \xleftrightarrow{\text{Fourier}} P(j\omega)$$

This signal is the pulse used by the PAM filter to respond to one bit of input. If we let  $b[k] = 1$  for a *binary one*, and  $b[k] = -1$  for a *binary zero* then the output of the PAM filter is

$$(2) \quad x(t) = \sum_k b[k]p(t - kt_0) \xleftrightarrow{\text{Fourier}} X(j\omega) = B(e^{j\omega t_0})P(j\omega)$$

This is one of the most important *mixed* transforms in Fourier analysis. The channel adds noise to this input and then passes the result through a lowpass filter  $H(j\omega)$ . In the time domain this is convolution.

$$(3) \quad y(t) = (h * (x + n))(t) = \int h(t - \tau)(x(\tau) + n(\tau))d\tau \longleftrightarrow Y(j\omega) = H(j\omega)[X(j\omega) + N(j\omega)].$$

Figure Two depicts an example. It was produced using our homebrew MATLAB function ‘[data\\_ch.m](#)’, found on the class web page under ‘Functions for Lab 13’. This function simulates a particular data channel. Pieces of the input signal  $x(t)$  and the output signal  $y(t)$  are shown on the top two panels. If we put the output into a spectrum analyzer, we would get a display like the one on the lower left which shows that the channel is indeed a lowpass filter. The graph on the lower right is an *eye diagram*. This is easily obtained in the laboratory and we will describe the experiment that produced it in the next section.

data channel in and out signals; 0.085333 seconds, 128 symbols, clock period = 16 samples

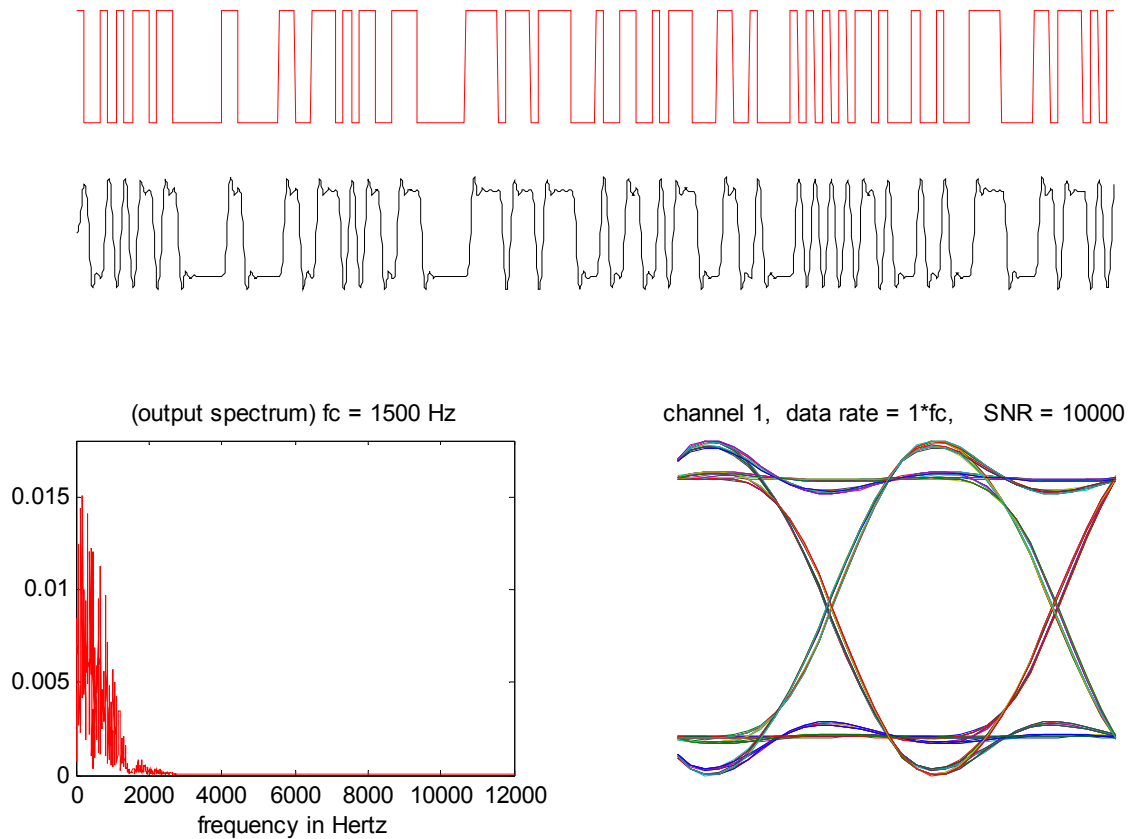
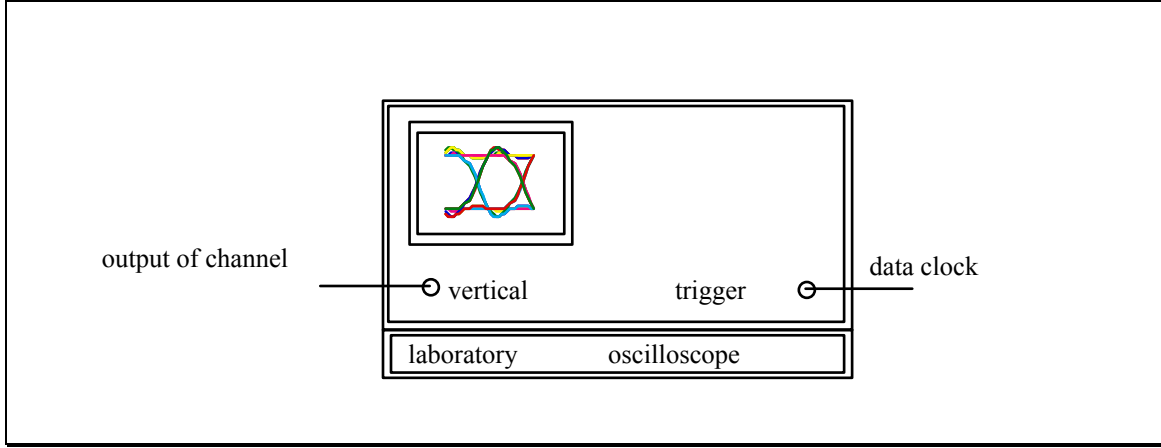


Figure Two Graphical output of 'data\_ch.m'

## EYE DIAGRAMS AND INTER-SYMBOL INTERFERENCE

The laboratory setup for producing an eye diagram is shown in Figure Three. It requires only an oscilloscope, assuming that the input signals are readily available. One connects the channel output  $y(t)$  to the vertical input of the scope and triggers on the data clock. Each trace is then begun at the same time within a bit (or, more generally, symbol) period. The horizontal time base generator frequency is then adjusted so that we get one to two bit periods across the screen. Thus the eye diagram contains many periods of the channel output overlayed. Its use is this. If there is a point within the eye diagram for which all traces corresponding to a *binary one* are well above all traces corresponding to a *binary zero*, then we can decode the signal  $y(t)$  and recover the binary sequence without error. Because of the shape of the collection of traces, this situation is called an *open eye*. We shall see however, that the eye can *close* if we attempt to transmit data at too high a rate, or if the additive noise power is too great. When the eye is closed there will be errors. This is an example of a situation where a simple laboratory experiment can get to the heart of the matter more quickly than theoretical analysis.



**Figure Three** Laboratory setup for obtaining an eye pattern

It is obvious from the eye diagram that the train of overlapping pulses which make up the channel output signal interfere due to interference from all previous pulses. If we sample the output synchronously (that is using a specified position within a bit period) then the resulting value will depend on all the previous symbols that were transmitted. This can be seen by combining equations (2) and (3). If we set the noise to zero (infinite signal-to-noise ratio), and sample at the symbol period  $t = mt_0$ , then

$$(4) \quad y(mt_0) = \sum_k b[k](h * p)(mt_0 - kt_0) \longleftrightarrow B(e^{j\omega t_0}) \sum_r HP(j\omega + jr\omega_0) = 1.$$

This output sample depends on all the symbols  $b[k]$  weighted by samples of the pulse  $(h * p)(t)$ . This is called *inter-symbol interference*. Ideally we would like  $y(mt_0)$  to depend only on one symbol, say  $b[m]$ . In this circumstance, it would be easy to recover the symbol. From equation (4),

$$(5) \quad y(mt_0) = b[m] \Leftrightarrow (h * p)(kt_0) = \delta[k], \text{ for all } k \Leftrightarrow \frac{\omega_0}{2\pi} \sum_r HP(j\omega + jr\omega_0) = 1$$

This situation is called *zero inter-symbol interference*. It is the situation that one might attempt to design by choosing the pulse  $p(t)$  to match the channel impulse response  $h(t)$ . But zero ISI is not always possible because of bandwidth limitations. How does bandwidth limit things?

## TWO FUNDAMENTAL LIMITATIONS, BANDWIDTH AND NOISE

Shannon produced a famous formula which describes what is possible for a power limited channel with a given bandwidth and signal-to-noise ratio. It gives the theoretical upper bound for channel capacity in bits per second. (This is the greatest data rate for which one can decode with arbitrarily small error probability, if an infinite amount of time and computational resources are available, assuming that we could figure out how to do it.) The Shannon inequality is

$$(6) \quad C \leq B \log_2 \left( 1 + \frac{P_s}{BP_n} \right),$$

where  $C$  is the capacity in bits per second,  $B$  is the channel bandwidth in Hz, and  $P_s$  and  $P_n$  are the signal and noise powers. The ratio  $P_s / P_n$  is called the power signal-to-noise ratio. We will experimentally demonstrate this limitation using a PAM transmitter and the eye diagram.

But first, let's look at a related problem. For a bandlimited channel, what is the greatest rate for which zero inter-symbol interference is possible? The answer comes out of the Shannon sampling theorem.

$$(7) \quad \text{If } (h * p)(kt_0) = \delta[k], \text{ for all } k, \text{ and } H(j\omega) = 0 \text{ for } |\omega| > 2\pi B, \text{ then } f_0 = \frac{1}{t_0} \leq 2B.$$

In other words, the maximum symbol rate, with zero inter-symbol interference, is *twice the channel bandwidth*. This rate is not automatically met. The limiting case is

$$(8) \quad (h * p)(t) = \text{sinc}(\pi t / t_0).$$

This is the *only* zero inter-symbol interference pulse with bandwidth  $f_0 = 2B$ . The symbol rate  $f_0 = 2B$  is called the *Nyquist rate* of the bandlimited channel. There are many zero ISI pulses when  $f_0 < 2B$ . There are no zero ISI pulses when  $f_0 > 2B$ . Keep in mind that the rate for zero inter-symbol interference and the channel capacity represent the limits for two distinct (but related) problems.

#### ***Assignment:***

There are two MATLAB functions for use in this assignment, which are located on the webpage. The first, called '[data\\_tx.m](#)', simulates a source and PAM transmitter. The other is called '[data\\_ch.m](#)'. It simulates the channel and produces the display shown in Figure Two. Investigate these using the 'help' facility.

##### 1. *The pulse shapes*

If the pulse selection variable 'pulseID' is negative, then only a single pulse is displayed. Type

```
»x=data_tx(.25,2,-1);xx=x(2:2049);% strip off clock period
»fs=24000;t=(1/fs)*[0:2047];subplot(2,1,1),plot(t,xx),subplot(2,1,2),spectrm(xx,fs)
```

One pulse is displayed in the time and frequency domains. The pulse width varies with the parameter 'rate', which is 0.25 in this example. Do this for all four pulse shapes and print the results. For future reference, keep in mind that if the rate is increased, then the pulse width will decrease and the pulse bandwidth will increase.

### *Assignment:*

#### 2. *The Channels*

If the channel selection variable 'ID' is negative, then the 'plotZTP.m' function, from lab 9, is used to display the channel filter. (These simulations are all done in discrete time of course.) Type

```
»data_ch(x,1,-1);
```

You will get a Z-transform view of the filter. The simulations use a sampling frequency of  $f_s = 24,000$  Hz, and a channel bandwidth of  $f_c = 1,500$  Hz. Therefore the normalized cutoff frequency is  $\alpha = 2\pi f_c / f_s = \pi / 8$ . This should be apparent in the display. Do this for all four channels (ID = -1,-2,-3, and -4), and print the results.

#### 3. *Zero ISI*

Start with the ideal case. If we select an impulse for the pulse shape, and use channel two (the linear phase FIR filter is a good approximation to the ideal lowpass filter), run the data at the Nyquist rate (rate=2) and set the signal to noise ratio to infinity (snr=1e6), then we should see a zero inter-symbol interference eye pattern. Type

```
»x=data_tx(2,2,2);y=data_ch(x,1e6,2);
```

Make a print of the result. Hand annotate the plot, indicating why the eye pattern exhibits zero inter-symbol interference. Now use the Chebychev channel (ID=3), a sinusoidal pulse (pulseID=4), at a low rate (rate=.5). This is one-fourth the Nyquist rate. Zero ISI is for all intents and purposes always achievable at very low rates. Indicate on the plot where you would sample to get zero ISI.

#### 4. *Multi-level data*

One way to get a higher bit rate is to transmit with more than two levels. If the signal to noise ratio is sufficiently high, then one can separate all the levels and decode the output. The following example runs at half the Nyquist rate but uses four levels, so that each symbol yields two bits of information. Type

```
»x=data_tx(1,4,1);y=data_ch(x,1e6,1);
```

Make a print of this, and annotate the eye pattern showing how the data can be recovered. (There are other ways to increase the bit rate which are commonly used. With one full wave of a sinusoid per symbol period, one can use several amplitudes and phases.)

#### 5. *Data rate and SNR*

For a particular choice of pulse and channel, do several simulations varying the data rate and signal-to-noise ratio. The goal is to determine which combinations of rate and SNR will allow for reliable data transmission. Use a square pulse (pulseID=1) and the Butterworth channel (ID=1). Transmit binary data, i.e. set levels=2. Make a graph with rate on the horizontal axis and SNR on the vertical axis. Both variables are positive, so the first quadrant is all that is necessary. Experimentally determine the boundary for reliable communication. Let us agree to consider a (rate,SNR) pair usable if the eye pattern has an opening of roughly 25% of full scale. You need not make prints of everything you try, but include one for the limiting 'rate' when SNR = 1e6, and one for the limiting SNR when rate = 2. You might want to use the line

```
»rate=2;snr=10;data_ch(data_tx(rate,2,1),snr,1);
```

which is easy to edit and reuse.

**Assignment:**

6. *Phase Distortion*

Repeat the experiment in question (5), using channel 4 (ID=4). This is the same Butterworth filter, except that an allpass filter is added. The magnitude of the frequency response is exactly the same (so the bandwidth is clearly the same). But the phase is more nonlinear. Make the same two dimensional sketch of usable (rate,SNR) pairs. How severe is the degradation in performance? In practice, one can correct phase distortion (as well as amplitude distortion) by appending an *equalizer* to the channel. This forms part of the receiver. There are ways to design the equalizer in real time, using so-called *adaptive filter* algorithms.

7. *Chebyshev Channel*

Channel 3 (ID=3) is a Chebyshev filter and has very sharp cutoff. Does this make it better or worse than the Butterworth channel for data transmission? Run some simulations using (rate,SNR) pairs on the boundary of the Butterworth reliable region. Then answer the question. Use the same square pulse. If the performance degrades, what is the reason?

8. *Data detector design*

Write a MATLAB tool called 'data\_rx.m' which recovers the binary information from the signal y. Then exercise it:

```
»x=data_tx(2,2,1);y=data_ch(x,20,1);d=data_rx(y);
```

Note that the first elements in the row vectors **x** and **y** are not data, but the clock period in samples. Explain your method of determining the phase at which you sample the channel output. In other words, how do you determine where the center of the eye is? This determination should be done from the data only. Do not use any side information about the channel filter. Include a display component in 'data\_rx.m', which plots the channel output **y**, and overlays vertical markers at the center-of-the-eye positions. Include a print of this display in your report. The vector **d** should be a row vector of zeros and ones. A vertical marker at time **t1** can be added to the plot using

```
hold on;plot([t1,t1],[-1,1],'w');hold off;
```