Modeling some data communications functions using Microsoft Excel 5.0.

Thomas C. McDermott, N5EG
Texas Packet Radio Society

Abstract

Recent enhancements to the Microsoft Excel[1] spreadsheet program, version 5.0, provide some interesting features that may be of interest to those designing or analyzing data modems. This paper looks at the following examples: 1) bit error rate of a modem vs. Eb/No in additive white gaussian noise (AWGN), 2) phase-locked loop response vs. loop filter parameters, and 3) modem eye patterns vs. channel response, and shows how each can be modeled with Excel.

Bit error rate in AWGN

To determine the theoretical bit error rate of a modem utilizing a certain modulation, it is necessary to know how often the noise voltage will exceed the signal voltage, given that the signal level (Eb), and the noise spectral density (No) is known. A common way to analyze this is to plot the bit error rate versus the Eb/No. In order to do this, an assumption about the statistical noise properties must be made. One assumption that is easy to model is that of additive white gaussian noise (AWGN) that has a uniform power spectral density (PSD) and a gaussian amplitude distribution. For an AWGN signal with zero-mean (no DC offset) and an R.M.S. voltage of 1 volt, the equation that expresses the probability density P(x) versus the voltage, x, is given by:

$$P(x) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{x^2}{2}\right] \qquad (1)$$

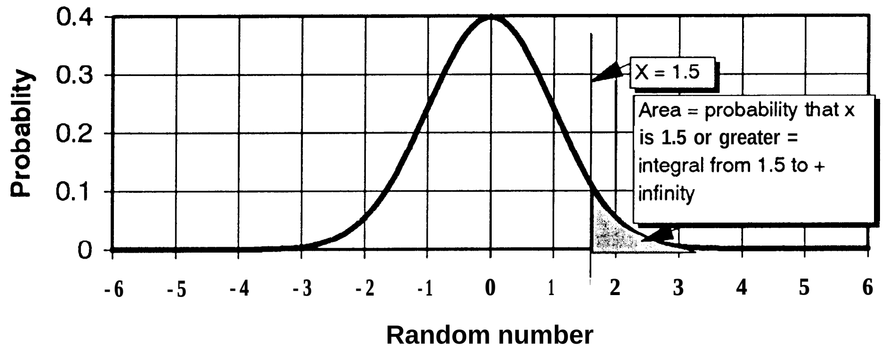
Then, the probability that the voltage x exceeds some value is the cumulative probability density of x versus the voltage. The probability that an error voltage will exceed the signal voltage is thus the cumulative density that the noise voltage is of the opposite magnitude and equal to or greater than the signal voltage. Assume that the signal voltage is n, then the probability of an error, versus the signal voltage, n is:

$$Error(x) = \int_{n}^{+\infty} \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{x^2}{2}\right] dx \qquad (2)$$

---

[1] Excel, Excel 5.0, and VBA are trademarks of Microsoft, Inc.

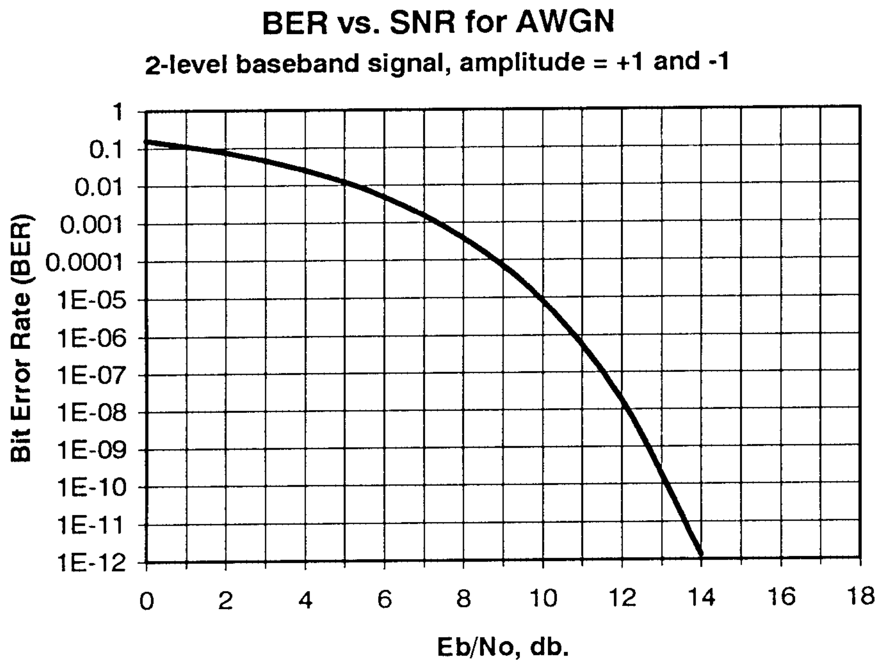
This equation is interpreted below, which shows the probability density of x (gaussian noise vloltage) and the shaded part of the curve shows the cumulative probability of the noise from +1.5 to + infinity.



To plot the probability of the signal, n, being less than the noise voltage requires calculating the value of P(x) in equation 2 for all values of n. This is a tedious task, since equation 2 does not have an explicit solution. However, Excel 5,O provides a function, called ERFC that provides the above probability. ERFCO is defined so that:

$$Error(x) = \frac{1}{2} ERFC\left(\frac{x}{\sqrt{2}}\right) \qquad (3)$$

Now, to plot the theoretical bit error rate (BER) of a signal vs. the Eb/No, all that is required is to plot Error(x) vs. x. The value x is generated as a ratio, and convenient decibel ratios are chosen for display (0 dB, 0.5 dB, 1 dB, etc.). This is plotted below, and is the theoretical BER vs. Eb/No (the value of x) for coherently-demodulated 2PSK modulation.



Different modulation formats can be compared by substituting into equation 3 the different formulas expressing x, the signal voltage.

Phase-locked loop (PLL) modeling

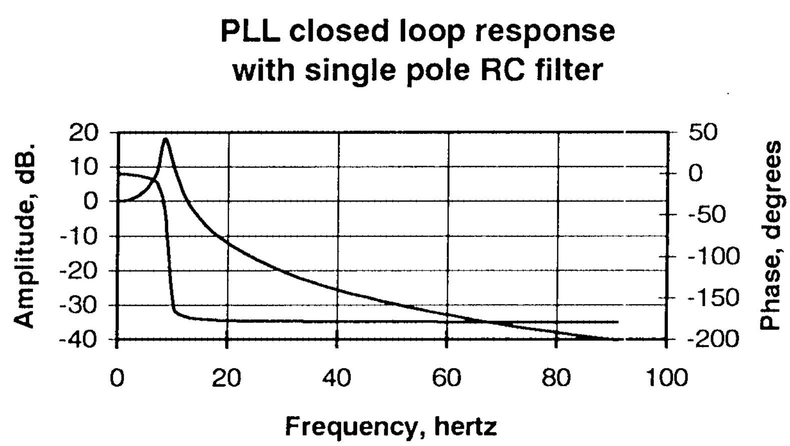
A simple phase locked loop consists of several components: a phase detector, a loop filter, and a voltage-controlled oscillator. In order to analyze the loop performance, it is necessary to express the phase detector gain, the loop filter gain, frequency, and phase response, and the VCO response. In general, the loop filter and the VCO response are complex (that is, they contain real and imaginary parts) and thus the equations must be computed using complex algebra. Two expressions of interest in the PLL design are the open-loop response, and the closed-loop response. The closed-loop response of a PLL is given by:

$$H_{closed}(s) = \frac{G(s)K_v K_f}{s + G(s)K_v K_f} \tag{4}$$

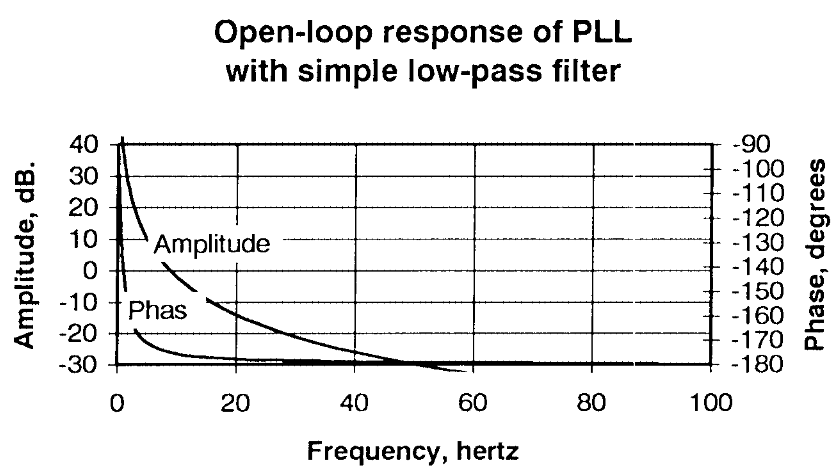
Where G(s) is the response of the loop filter, s is the Laplace variable (equal to $j\omega$ for sinusoidal analysis), Kv is the VCO control voltage gain, and Kf is the phase detector gain. Similarly, the open-loop response is given by:

$$H_{open}(s) = \frac{G(s)K_v K_f}{s} \tag{5}$$

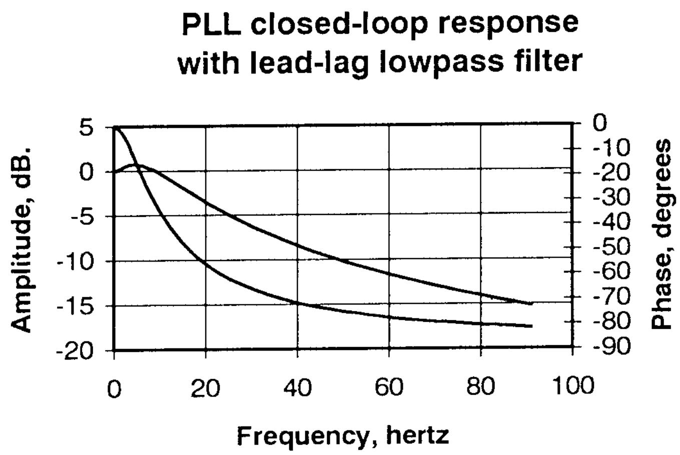
To plot the magnitude and phase of H(s) versus the frequency, the use of complex algebra is required. Excel 5.0 supports complex numbers, and operators to add, subtract, multiply, divide, and to find the magnitude and phase of a complex number. These operations are not always as convenient as manipulating real numbers in Excel, so several steps are required in the computation. First, an array of numbers is set up as the frequency, f. Then the value of s is computed. s is equal to $j\omega$, or $j2\pi f$. j is equal to the square-root of negative one (an imaginary number). In Excel, s is equal to =COMPLEX(0,2*PI()*B9) where B9 happens to be the cell containing the frequency in radians per second (for this example). Thus, the real part is zero, and the imaginary part is 2*PI*f, and s obviously is j*2*PI*f. G(s), the loop-filter frequency response can be computed, and in the particular spread-sheet, each column is a different frequency, and each row is a partial product, such as s, G(s), and finally H(s). Then the magnitude and phase of H(s) are computed as additional rows, which can then be plotted versus frequency. The diagram below shows the closed-loop response of one such computation, a PLL with a simple low-pass loop filter. It can be seen that the loop response is very poorly damped, and the loop is near instability, with a gain peak near 9 hertz.



The open-loop response of this PLL is computed almost identically, and is shown below.



From the open-loop response it can be seen that the amplitude curve crosses zero-dB. at 9 hertz, and that the loop has about 6-degrees of phase margin, clearly an invitation to disaster with this loop. Fortunately, it is easy to make more complex lead-lag filters for the loop-filter, G(s), and perform any additional calculations in Excel. It is also easy to make the pole frequencies of G(s) adjustable, so they can be altered, and the PLL response plotted accordingly. The diagram below shows the closed-loop response of a PLL with a lead-lag loop filter. This PLL is much more likely to operate properly.



Eye pattern versus modem channel response

A more complicated example, but one which shows the power of Excel is to compute the eye pattern that would be seen at a receiver given knowledge of the frequency and phase response of the channel. We'll assume a real frequency response, but this is not necessary (it does make this example less difficult). The key to this is provided by two features of Excel: the ability to compute the Inverse Fast Fourier Transform (IFFT), and the ability to write functions in Visual Basic Application (VBA) language that comes embedded within Excel 5.0. A function that is needed is the Multiply-Accumulate operation, which forms the kernel of convolution and correlation integrals.

Given the frequency and phase response of the channel, the impulse response of the channel is given by the Inverse Discrete Fourier Transform (IDFT) or the IFFT of the channel frequency response. If the frequency response has no phase variation, then the impulse response will contain only a real component. Once the impulse response is known, the time-domain response of the channel to a data signal can be computed by linear convolution of the data bits with the impulse response. The convolution function is given by:

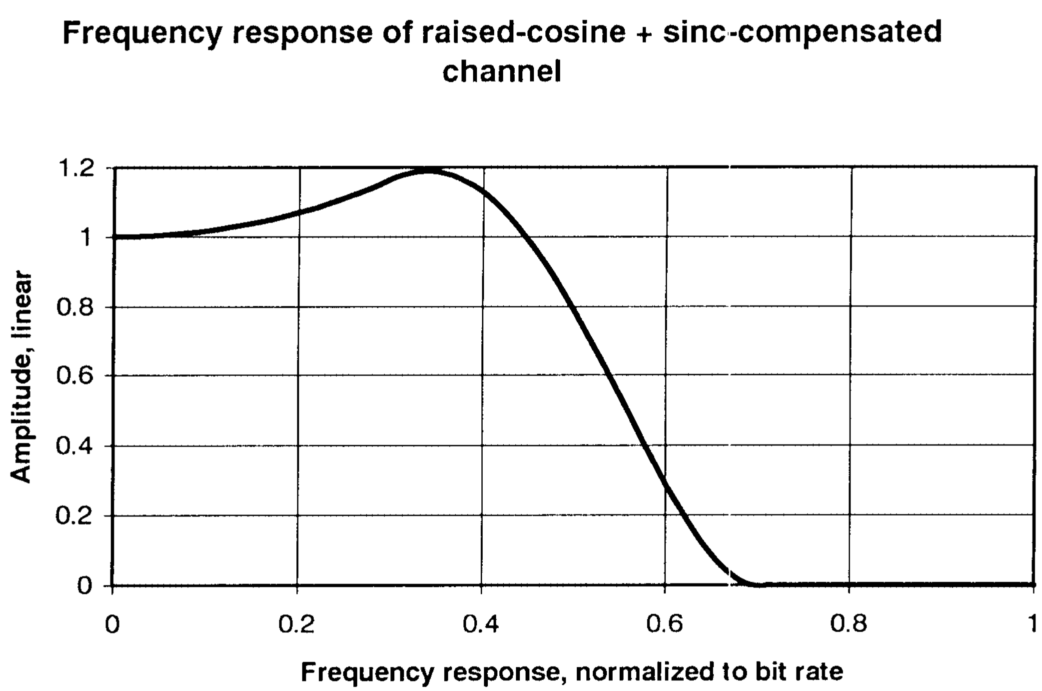
$$y(t) = \int_{-\infty}^{+\infty} h(\tau)x(t-\tau)d\tau \qquad (6)$$

Where y(t) is the output of the convolution, h(t) is the channel impulse response, and x(t) is the input signal to the channel. It can be recognized that when converted to discrete-time, this is just exactly the equation of a finite-impulse-response (FIR) digital filter (in fact, this is how the FIR filter is derived). The Multiply-Accumulate operation performs the multiplication of h and x for all values of tau, and sums them. This operation is then repeated for the next value of t. Thus, an array of Multiply-Accumulate (MAC) functions can perform a linear convolution (or

FIR filtering) of the signal. The MAC function written operates just like a built-in Excel function, and it can be copied and pasted in ranges.

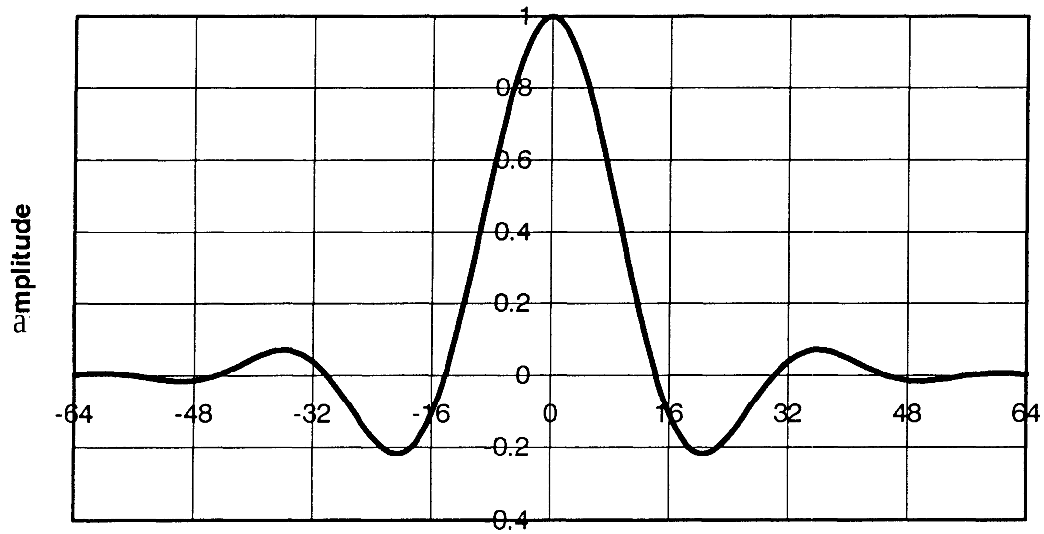
To generate the eye pattern, a filter with a suitable frequency response is chosen. This is converted into an array of frequency versus amplitude. This example uses a raised-cosine filter with an alpha factor of 0.4. Then the Inverse FFT is computed on the response and the resultant output circularly-shifted to produce the real impulse response. Next, a pseudo-random bit sequence is generated by writing another VBA function. This bit stream is stored in an array of cells. Then the channel impulse response is convolved with the pseudo-random bit stream, and the resultant time-domain signature of the ringing filter is produced as an array. Finally, many pieces of this time signal, each 3-bit times long, and each offset by one bit time are generated as a 2-dimensional matrix. All of the signals in the matrix are then plotted on top of one another, resulting in an eye diagram.

The diagram below show the frequency response of the channel for an alpha=0.4 sinc-compensated channel filter.



The impulse response of this channel is the computed using the IFFT, and the impulse response is shown below.

Impulse response of alpha=0.4 sine-compensated channel

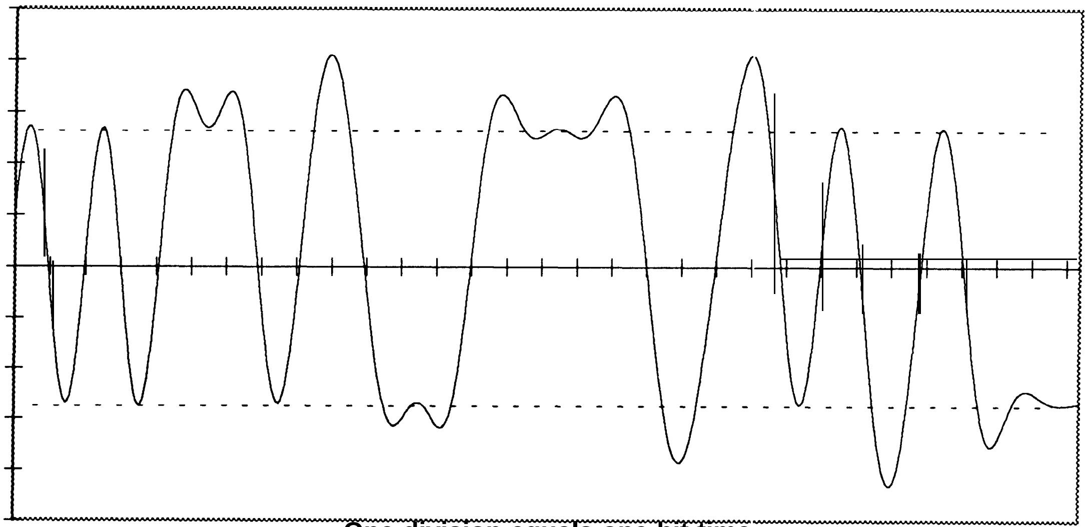


Time, 16 units equal one databit interval

Note that the zero-crossings of the sine-compensated impulse response do not cross through zero at the bit time (an uncompensated raised-cosine impulse response does cross through exactly zero at the bit time) Next, the convolution of the pseudo-random bit stream with the impulse response results in the time-domain waveform from the filter, shown below.

Time-domain response of x/xin(x) compensated raised-cosine filter

alpha = 0.4, sequence = 2'3 - 1 PRBS

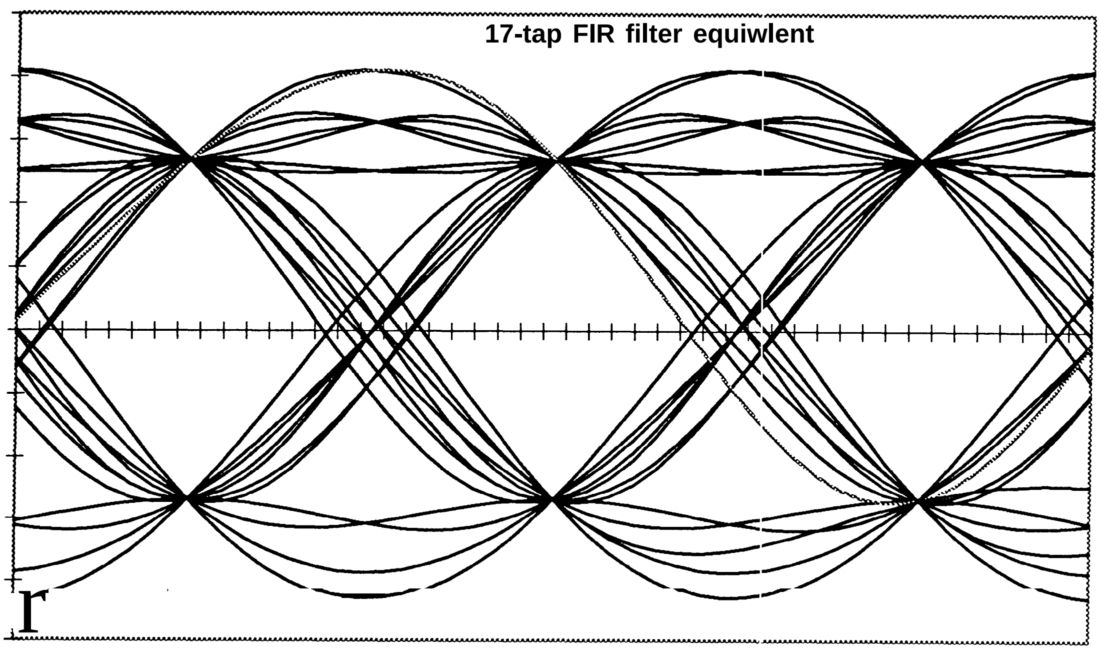


One division equals one bit time

And finally, the resultant eye-pattern from the above time-domain waveform is shown.

Eye Pattern - Raised cosine + x/sin(x) compensation

alpha = 0.4, sequence = 2'3 - 1 PRBS



Summary

It is possible to graphically solve many interesting problems in the design and analysis of data communications systems using spreadsheets. Three examples have been presented which illustrate the utility of this method.