Modeling Some Data Communications Functions using Microsoft Excel 5.0

T. C. McDermott
Texas Packet Radio Society
September 9, 1995

Features that aid modeling

Complex numbers:

- Make complex, add, subtract, multiply, divide,
- extract magnitude, phase parts
- extract real, imaginary parts

Fourier transform:

- FFT, IFFT (for integer powers of two only)
- Requires installation of the analysis toolpack

VBA - Visual Basic for Applications:

- Comes installed as a standard part of Excel
- Use to automate tiresome functions:
  - Multiply-Accumulate
  - Pseudo-random number generation

Features that aid modeling, cont'd.

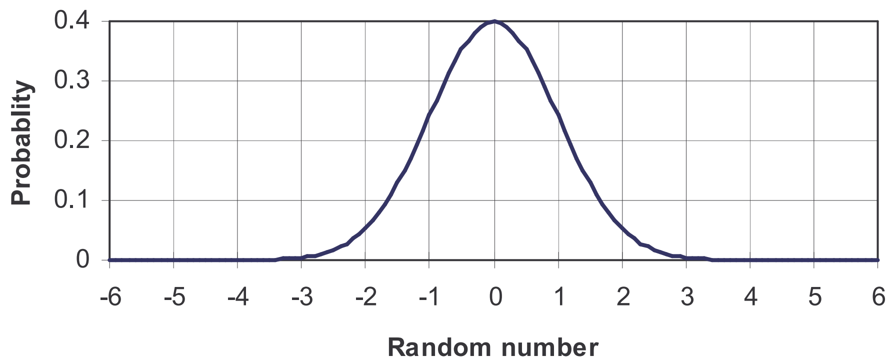
- Engineering and Statistical functions:
    - Gaussian, uniform, Poisson random number generators
    - Evaluation of Gaussian, Poisson PDF's.
    - Evaluation of Gaussian PDF integral
        - Complementary error function - erfc()
    - Trigonometric functions
        - Sin, Cos, Tan, Sinh, Cosh, Tanh

- Good graphing capabilities
    - Log, linear, multivariable, x-vs.-y, etc.

BER vs. AWGN

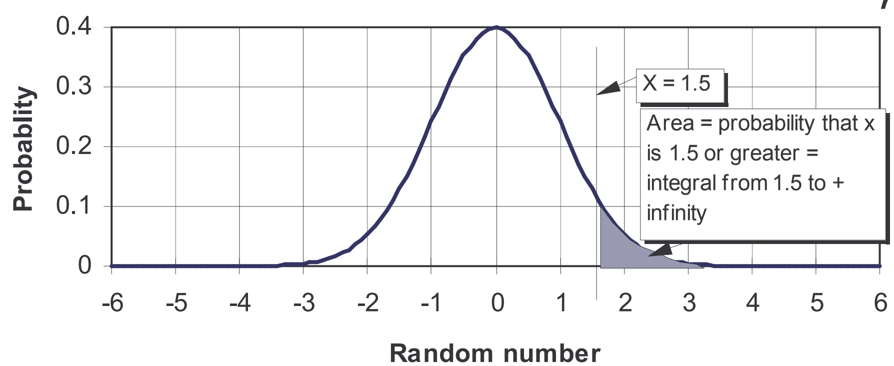
Probability density of AWGN noise:

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



Cumulative density of AWGN noise:

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

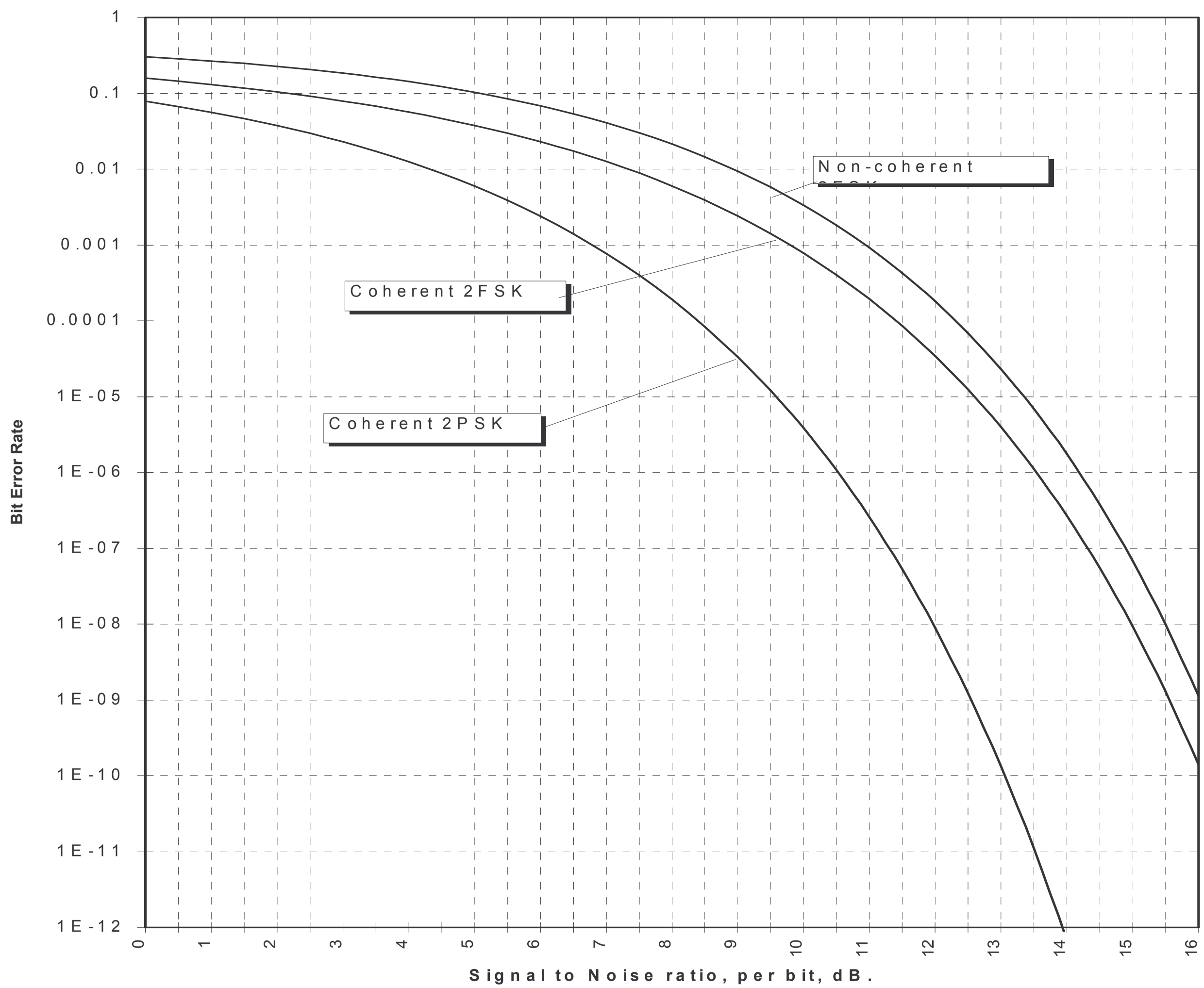


BER generation vs. AWGN

SNR / bit, db.	0	0.5	1	1.5	2	2.5	3
SNR/bit, ratio	1	1.122018	1.258925	1.412538	1.584893	1.778279	1.995262
2PSK	0.07865	0.067065	0.056282	0.046401	0.037506	0.029655	0.022878
Coherent 2 FSK	0.158655	0.144742	0.130927	0.117318	0.104029	0.09118	0.078896
Noncoherent FSK	0.303265	0.285316	0.266439	0.246741	0.226368	0.205505	0.184376

Converting dB. ratio to SNR ratio:	=10^(C3/10)
Converting x to P(x) for 2PSK	=0.5*ERFC(SQRT(B5))
Converting x to P(x) for C2FSK	=0.5*ERFC(SQRT(B5/2))
Converting x to P(x) for NC2FSK	=0.5*EXP(-B5/2)

Graphing the results

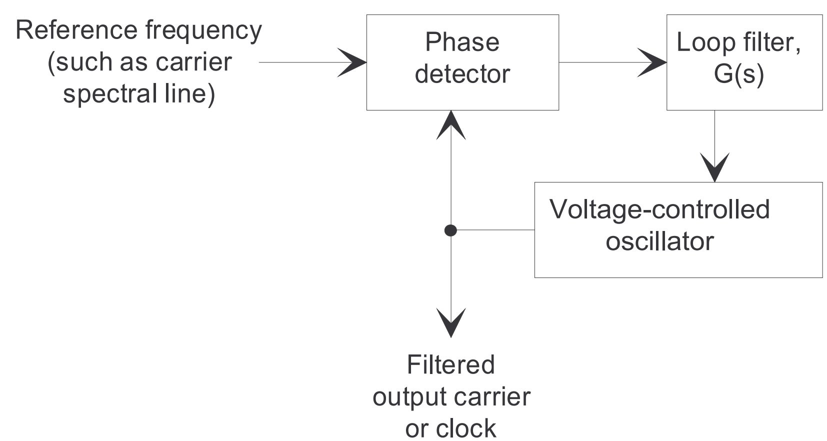


Phase-Locked Loop Analysis

Phase-Locked Loops are useful in clock- and carrier- recovery circuits for digital modems.

Open-loop and Closed-loop responses of the PLL are usually the two basic parameters of interest.

Need to solve for output phase divided by input phase for both cases.



Derivation of closed-loop gain

Write the loop equation: $Out = (In - Out)\ K_v\ G(s)\ \frac{K_f}{s}$

$$Out + Out\left(K_v\ G(s)\ \frac{K_f}{s}\right) = In\ K_v\ G(s)\ \frac{K_f}{s}$$

$$Out\left(1 + K_v\ G(s)\frac{K_f}{s}\right) = In\ K_v\ G(s)\frac{K_f}{s}$$

$$\frac{Out}{In} = \frac{K_v\ G(s)\frac{K_f}{s}}{\left(1 + K_v\ G(s)\frac{K_f}{s}\right)} \qquad H(s) = \frac{Out}{In} = \frac{G(s)\ K_v K_f}{s + G(s)\ K_v K_f}$$

Calculating the Closed-Loop Response vs. Frequency

Frequency, hertz	0.01	0.012	0.0144	0.01728	0.020736
s	6.2831853	7.5398223	9.04778684	0.1085734	0.1302881
G(s)	9.9984211	9.9977265	9.99672657	9.9952869	9.9932146
KvKfG(s)	628.21932	628.17568	628.112855	628.02240	627.89219
s+KvKfG(s)	628.21932	628.17568	628.112855	628.02240	627.89219
H(s)	1.0000012	1.0000017	1.00000258	1.0000037	1.0000053
mag H(s)	1.000001	1.000002	1.0000026	1.000004	1.000005
phase H(s)	-0.00573	-0.00688	-0.008251	-0.0099	-0.01188
mag H(s), dB.	1.09E-05	1.57E-05	2.254E-05	3.25E-05	4.67E-05

s =COMPLEX(0,2*PI()*B9)

G(s) =IMDIV($G$5*$G$6,IMSUM(B10,$G$6))

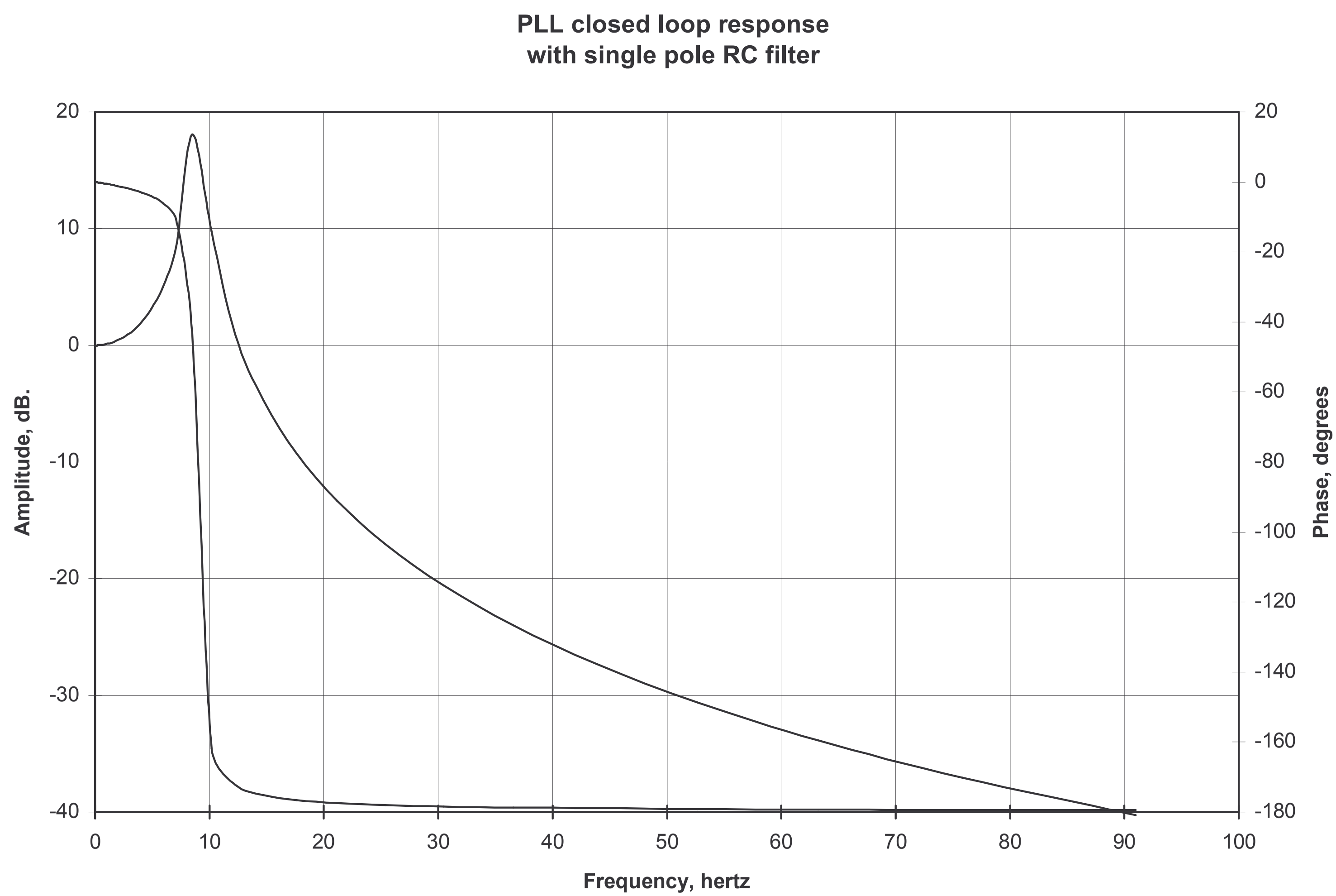
KvKfG(s) =IMPRODUCT($C$5,$C$6,B11)

s+KvKgG(s) =IMSUM(B10,B12)

H(s) =IMDIV(B12,B13)

mag H(s) =IMABS(B14)

Graphing the result



Deriving the Open-Loop Response

Write the transfer function: $$Out = In\, K_v\, G(s)\, \frac{K_f}{s}$$

$$\frac{Out}{In} = K_v\, G(s)\, \frac{K_f}{s}$$

Solve for transfer gain: $$\frac{Out}{In} = \frac{K_v\, G(s) K_f}{s}$$

Computing the Transfer Response

Open Loop Respons	-125.6438	-125.6351	-125.6225	-125.6044	-125.5784
Open loop mag, dB	79.99931	78.41539	76.831328	75.24708	73.66255
Open loop phase, de	-90.72	-90.8639	-91.03669	-91.244	-91.4927

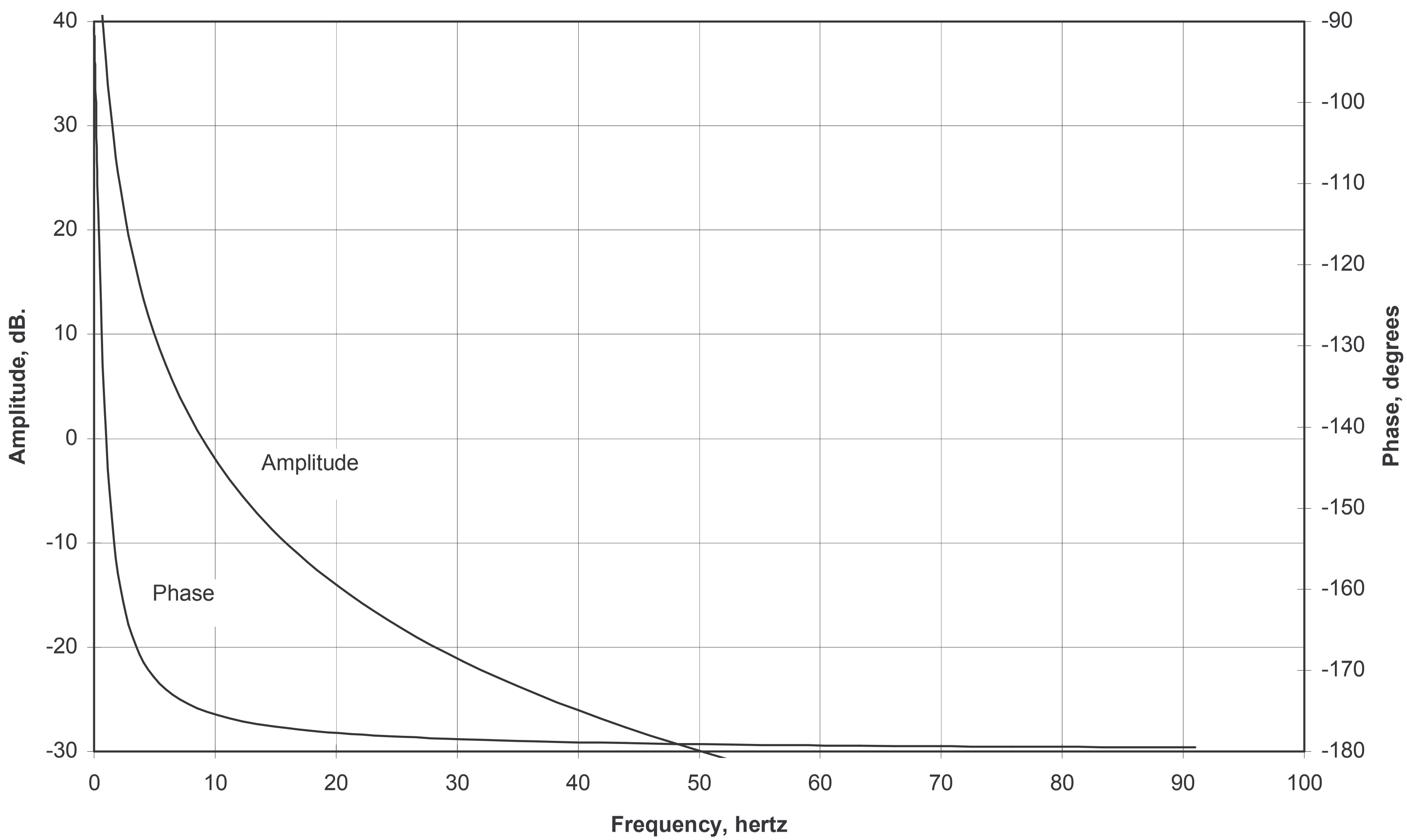
Open Loop Response =IMDIV(B12,B10)

Open loop mag, dB. =20*LOG(IMABS(B18))

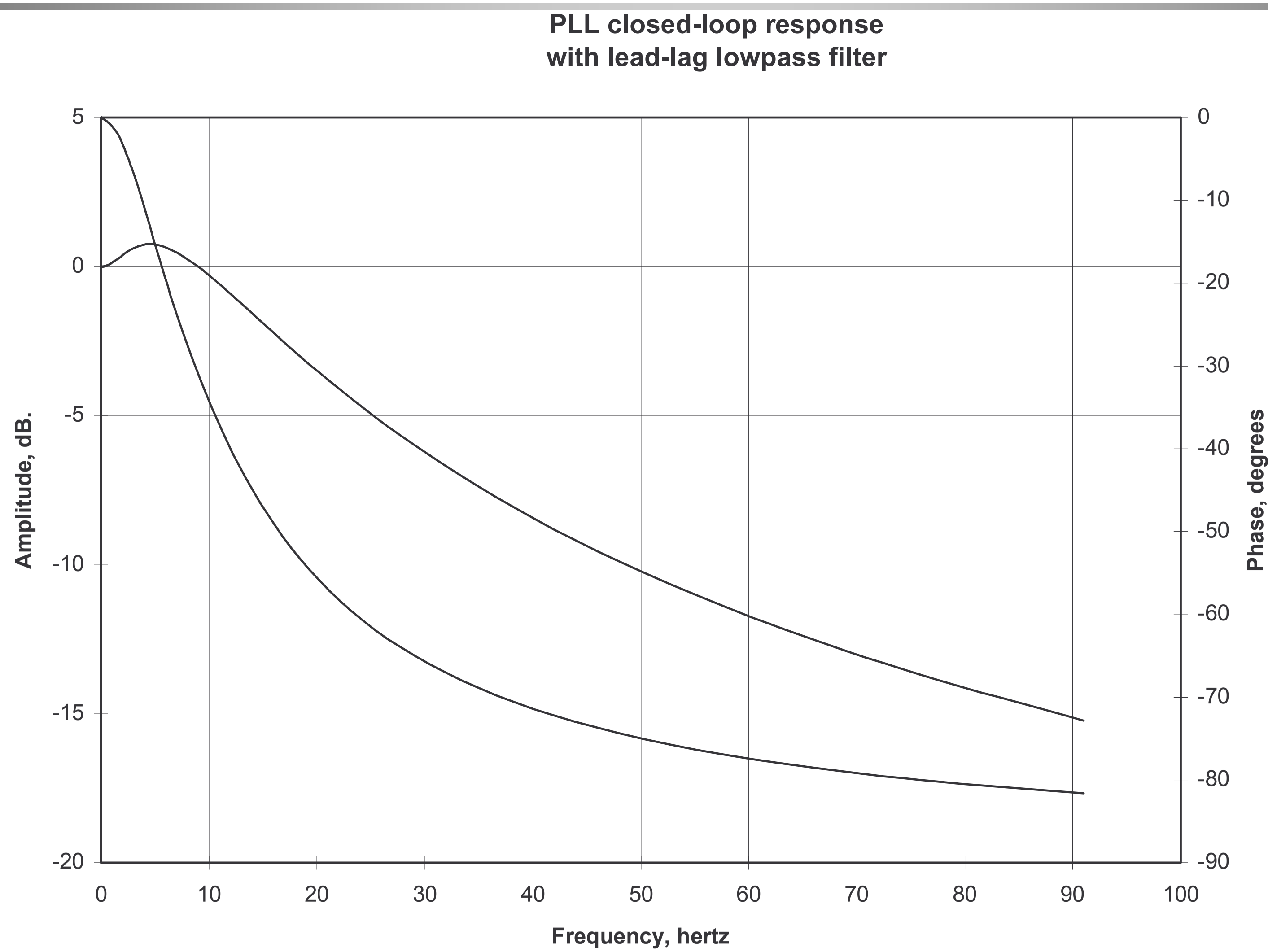
Open loop Phase, deg. =IMARGUMENT(B18)*180/PI()

Graphing the Result

**Open-loop response of PLL
with lead-lag filter**



Closed-Loop Response with Improved Loop Filter



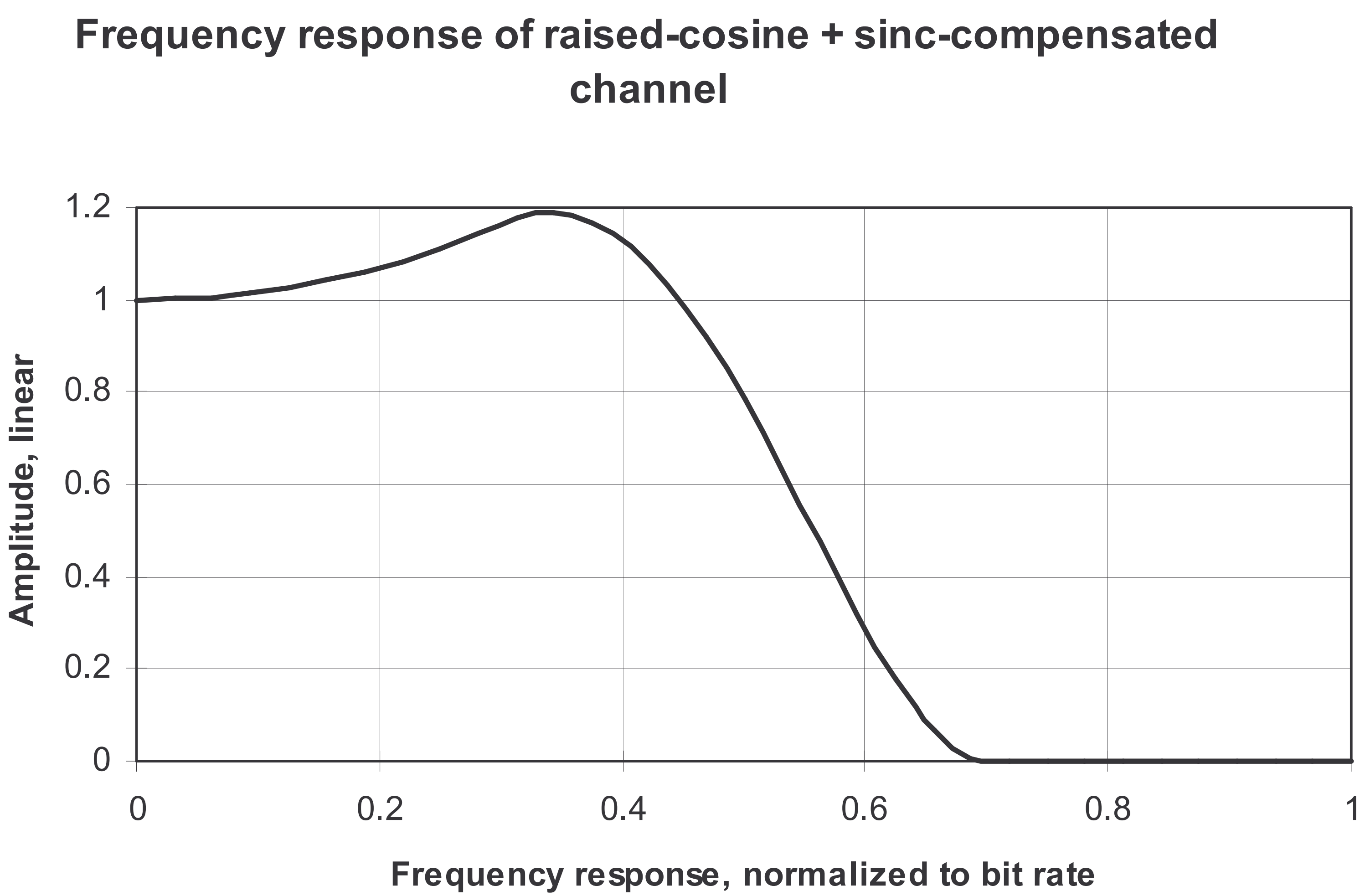
Eye Pattern vs. Modem Channel Response

This function is somewhat more complicated than previous examples.

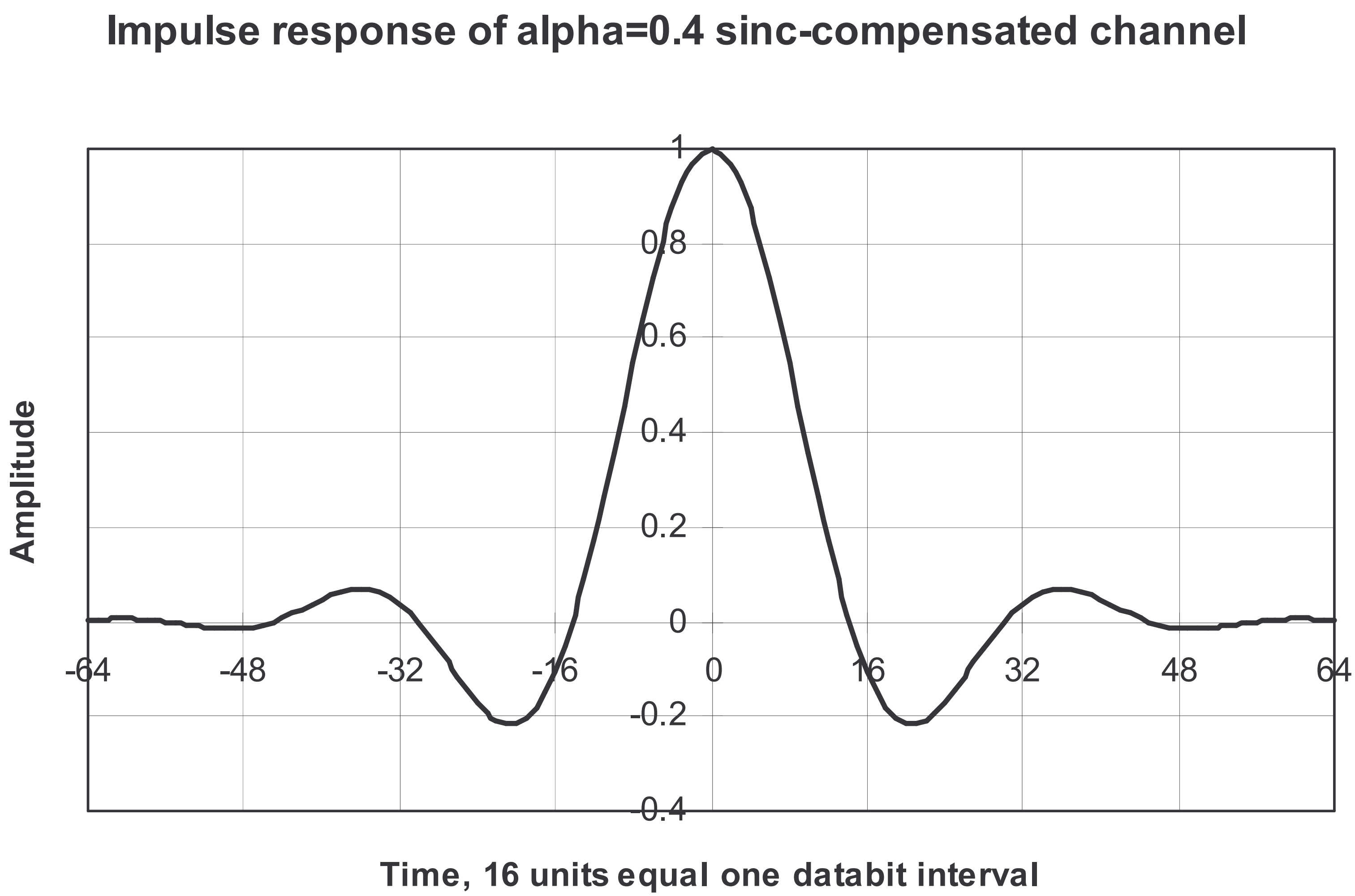
The general procedure is:

1. Generate channel amplitude/phase response vs. frequency. We'll use a raised-cosine function in this example.
2. Use inverse Fourier transform to generate channel impulse response. A large amount of over-sampling gives smoother results in the time domain.
3. Generate pseudo-random data stream.
4. Convolve data stream with channel impulse response. This generates the time-domain filter output with ringing.
5. Carve up the ringing waveform into 3-bit wide overlapping slices and display against bit periods -- this is the eye pattern.

Step 1 - Generate Amplitude vs. Frequency



Step 2 - Inverse FFT to Generate Impulse Response



Step 3 - Generate Pseudo-Random Data Stream using VBA Macro

```
' PRBS5 Generator Procedure
' 1/14/95 by Tom McDermott
' PRBS5(expansion factor)
' Generates PRBS sequence, replicates each bit 'expand' times
' from currently selected cell DOWN
'
Sub PRBS5()
' 2 and 5 are feedback taps
Expand = InputBox(prompt:="How many cells per data bit?", default:=1)
  s1 = 1
  s2 = 0
  s3 = 0
  s4 = 0
  s5 = 0
 For I = 0 To 31    'sequence length = 31
                    'output bits from S.R. 'expand' times
    For J = 1 To Expand
        RowI = I * Expand + J
        If s5 = 1 Then
            Selection.Cells(RowI, 1).Value = 1
        Else
            Selection.Cells(RowI, 1).Value = -1
        End If
'       Selection.Cells(RowI, 2).Value = s1 + s2 * 2 + s3 * 4 + s4 * 8 + s5 * 16
    Next J
    NextIn = s2 Xor s5
```

PRBS-5
-1
-1
-1
-1
-1
-1
-1
-1
-1

Step 4 - Perform Linear Convolution of Data with Impulse Response

Convolution Integral is: $$y(t) = \int_{-\infty}^{+\infty} h(\tau)x(t-\tau)d\tau$$

Key to most DSP functions (Convolution, Correlation, Autocorrelation, etc.) is the Multiply-Accumulate kernel. This can be written as an Excel VBA function:

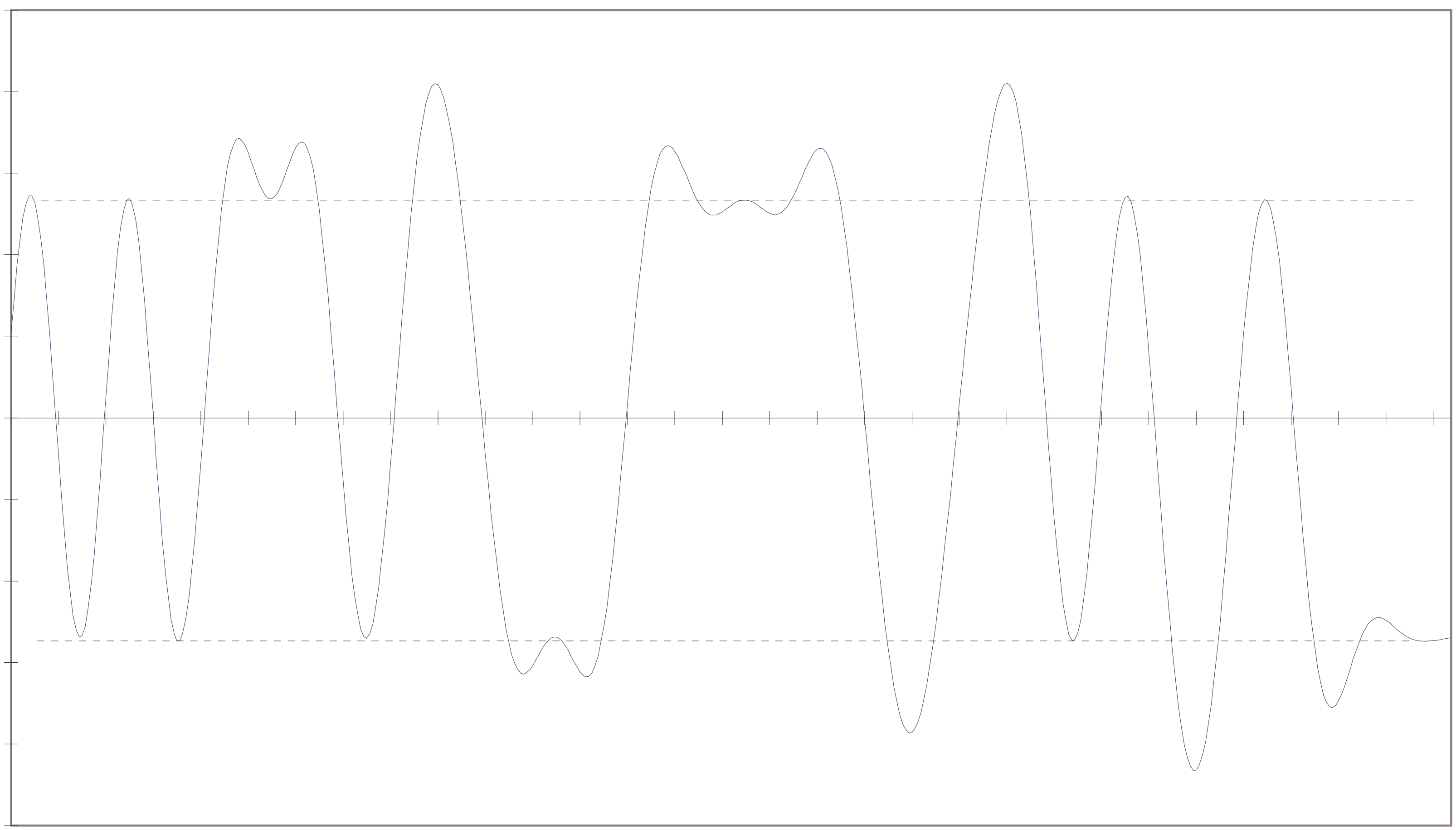
```
' MpyAcum Procedure
' 1/12/95 by Tom McDermott
' Kernel of many DSP routines
' Especially useful for convolution
Function MpyAcum(ImpCellRef, DataCellRef)
   Dim Acum As Single
   Irow = ImpCellRef.Rows.Count
   Icol = ImpCellRef.Columns.Count
   Drow = DataCellRef.Rows.Count
   Dcol = DataCellRef.Columns.Count
   Acum = 0
   For I = 1 To NumCalc
     Acum = Acum + ImpCellRef(I) * DataCellRef(I)
   Next I
   MpyAcum = Acum
End Function
```

=MpyAcum($B$10:$B$138,G10:G138)

Graphing the Result gives the Time-Domain Response of the Channel

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

alpha = 0.4, sequence = 2^5 - 1 PRBS

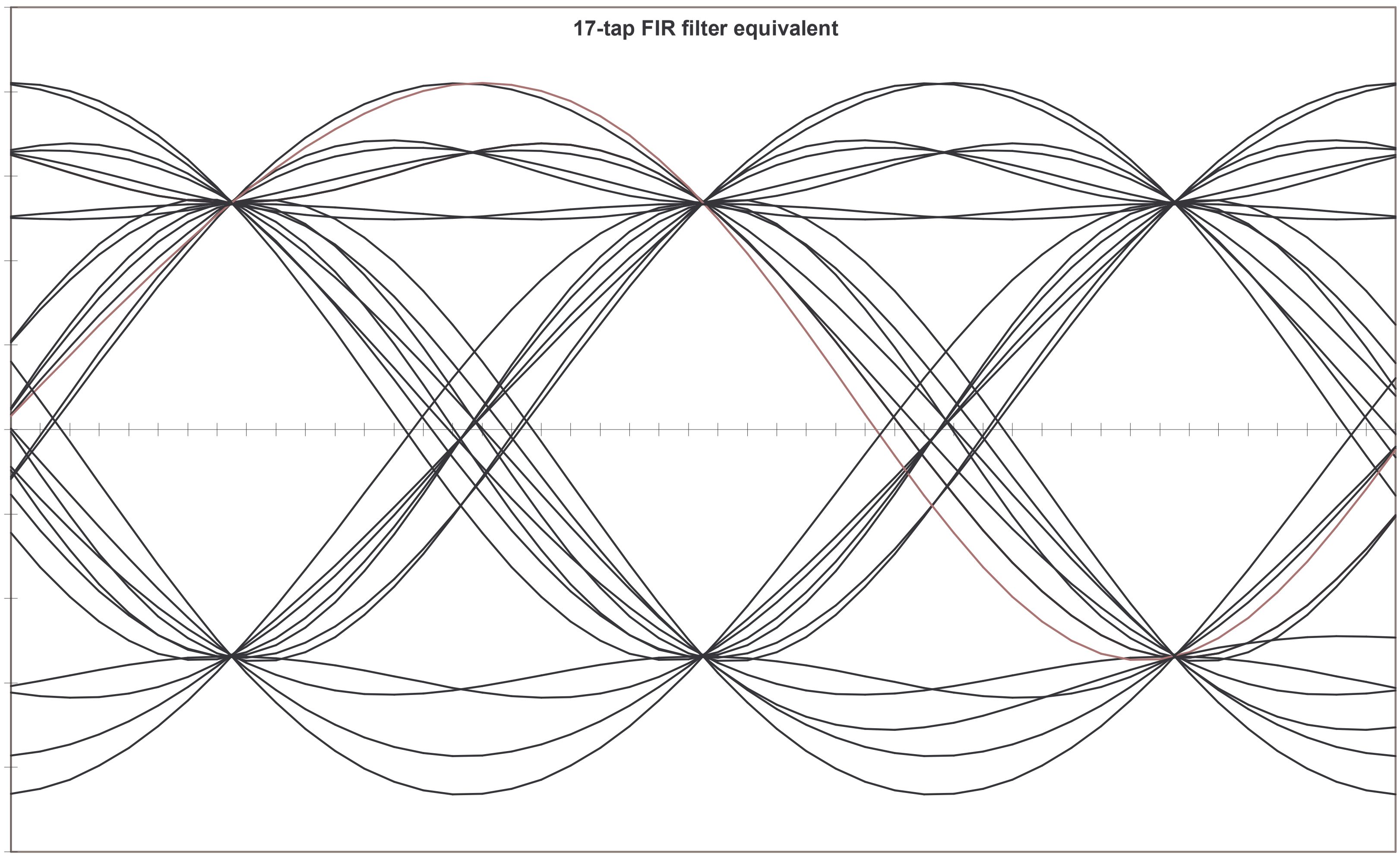


One division equals one bit time

Step 5 - Carve-up the Time-Domain Response into 3-bit wide Pieces

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

alpha = 0.4, sequence = 2^5 - 1 PRBS



Summary

With the aid of:

- **Complex numbers,**
- **Fourier transform,**
- **User-definable macros and functions,**
- **and graphical presentation aids.**

Modern spreadsheets permit the analysis of many data communications functions.