

# Commsim 2001 Tutorial

## Simulation of Communication Systems

### Introduction

A typical communication link includes, at a minimum, three key elements: a transmitter, a communication medium or channel, and a receiver. The ability to simulate all three of these elements is required in order to successfully model any end-to-end communication system.

The transmitter and receiver elements can in turn be subdivided into further sub-systems, as shown in the figure below. These include a data source (analog or digital), an optional data encoder, a modulator, a demodulator, an optional data decoder, and a signal sink.

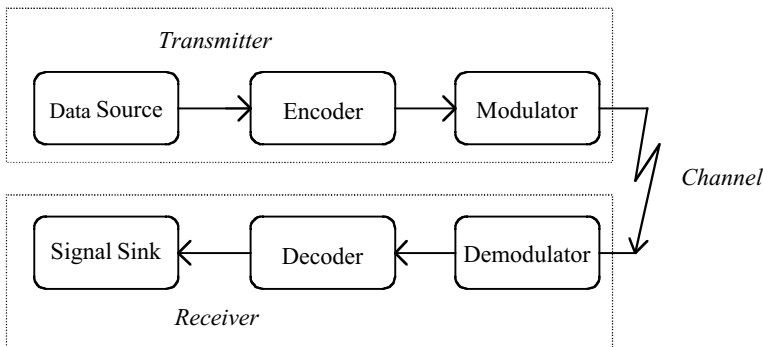


Figure 1. Communication System Elements

The *data source* generates the information signal that is intended to be sent to a particular receiver. This signal can be either an analog signal such as speech, or a digital signal such as a binary data sequence. This signal is typically a baseband signal represented by a voltage level.

For analog signals, it is often desirable to digitally *encode* the signal prior to transmission by undergoing a quantization process. This step converts the analog signal into a digital signal. While some information is lost in this process, the resulting digital signal is often far less susceptible to the effects of noise in the transmission channel.

An *encoder* can also be used to add redundancy to a digital data stream, in the form of additional data bits, in a way that provides an error correction capability at the receiver. This overall process is referred to as *Forward Error Correction* (FEC). Among the most popular FEC schemes are convolutional coding, block coding and trellis coding. It is important to note that usually the output bit rate of an encoder is not equal to the input bit rate. To properly distinguish between the two bit rates, the transmitter's input rate is referred to as the information data rate, while the transmitter output rate is referred to as the channel data rate.

Depending on the type of information signal and the particular transmission medium, different *modulation* techniques are employed. Modulation refers to the specific technique used to represent the information signal as it is physically transmitted to the receiver. For example, in Amplitude Modulation (AM), the information is represented by amplitude variations of the carrier signal.

Once the signal is modulated, it is sent through a transmission medium, also known as a *channel*, to reach the intended receiver. This may be a copper wire, coax cable, or the atmosphere in the case of a radio transmission. To some extent, all channels introduce some form of distortion to the original signal. Many different channel models have been developed to mathematically represent such distortions. A commonly used channel model is the Additive White Gaussian Noise (AWGN) channel. In this channel, noise with

uniform power spectral density (hence the term “white”) is assumed to be added to the information signal. Other types of channels include fading channels and multipath channels.

When the transmitted signal reaches the intended receiver, it undergoes a *demodulation* process. This step is the opposite of modulation and refers to the process required to extract the original information signal from the modulated signal. Demodulation also includes any steps associated with signal synchronization, such as the use of phase locked loops in achieving phase coherence between the incoming signal and the receiver’s local oscillator.

When data encoding is included at the transmitter, a data *decoding* step must be performed prior to recovering the original data signal. The signal decoding process is usually more complicated than the encoding process and can be very computationally intensive. Efficient decoding schemes, however, have been developed over the years -- one example is the Viterbi decoding algorithm, which is used to decode convolutionally encoded data.

Finally, an estimate of the original signal is produced at the output of the receiver. The receiver’s output port is sometimes referred to as the *signal sink*. As communications engineers, we are usually interested in knowing how well the source information was recreated at the receiver’s output. Several metrics are used by engineers to evaluate the success of the data transmission. The most common metric, in the case of digital signals, is the received *Bit Error Rate (BER)*. Other valuable performance indicators include the received *Signal to Noise Ratio*, *Eye Patterns* and *Phase Scatter Plots* to name a few.

## Signal Sources

Commsim 2001 provides both analog and digital signal sources. By combining basic signal producers blocks, the user can generate just about any analog time domain waveform that is desired. The basic building blocks provided include sinusoids, constants, ramps, step functions and impulses to name a few. The user can also import a signal from an external file, including wave files (.wav).

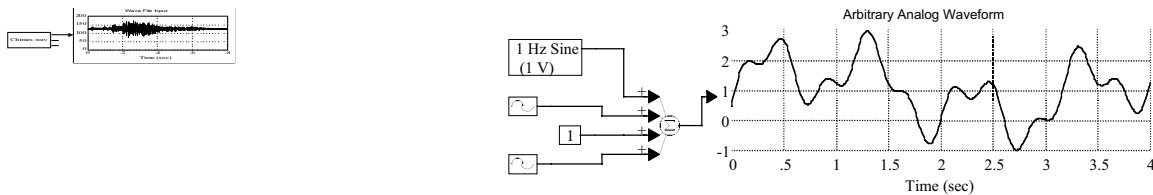


Figure 2. Analog Waveform and Wave File Examples

When working with digital signals, the Commsim 2001 environment provides sources for generating random bits or symbols, as well as pseudo-random bit patterns such as PN sequences. A *bit* can assume a value of either “1” or “0”, while a *symbol* is defined as a value which is a member of the set of  $M$  possible integer values  $\{0,1,2,\dots,M-1\}$ . Symbol values are typically used within Commsim 2001 when dealing with high order modulation constellations.

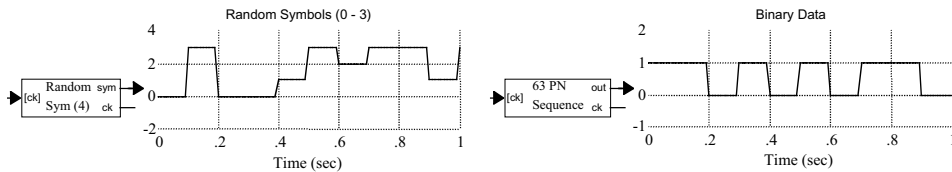


Figure 3. Digital Waveform Examples

Data source parameters are easily specified by accessing each block’s Setup dialog. For example, the Random Symbol Source block allows the user to specify the symbol alphabet size (2 for binary, 4 for QPSK), the symbol rate in hertz, and a data start time. In addition, this block allows the use of a user supplied clock

by selecting external timing within the setup dialog.

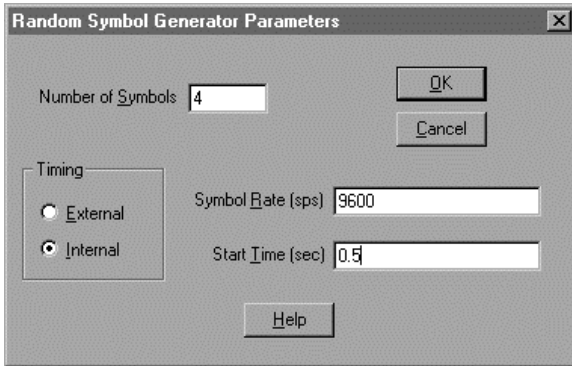


Figure 4. Random Symbol Source Setup Dialog

When working with digital sources, the simulation sampling frequency should be chosen so as to provide, at a minimum, on the order of 8 or 10 samples per symbol. This procedure will ensure that any filtering operations in the model will be accurately simulated. Of course, should there be another signal requiring a higher sampling frequency (e.g. a carrier), then the latter would dictate the necessary sampling frequency for the simulation.

## Signal Encoding

As previously mentioned, signal encoding is performed to increase the reliability of information transfer. Signal encoding can assume several forms, such as the quantization of an analog signal, or the addition of “smart” redundancy into a digital data stream for FEC purposes.

The quantization of an analog signal to obtain a digital data stream is usually performed using an Analog to Digital Converter (ADC). The Commsim 2001 environment provides several basic blocks to support this process, including an ADC block and a data compander block. Companding is a non-linear process used prior to quantization to provide higher resolution to the lower voltage range of a signal.

For digital signals, Commsim 2001 provides several methods for implementing forward error correction of digital data streams. Support is included for both convolutional codes and trellis coded modulation. Blocks to perform data interleaving and de-interleaving are also provided.

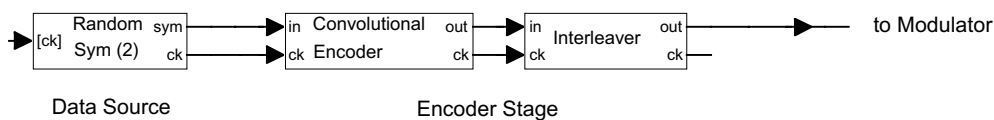


Figure 5. Example of Digital Data Encoding in Commsim 2001

## Convolutional Codes

Convolutional encoding is performed by passing a binary data sequence through a finite length shift register. A total of  $n$  outputs are then generated using different linear combinations (binary modulo addition) of the shift register contents. The specific tap connections are specified by the code *generator coefficients*, which are usually represented in octal format. Data bits are shifted  $k$  bits at a time into the encoder. The shift register size divided by  $k$  is referred to as the *constraint length  $L$*  of the encoder. The ratio  $k/n$  is referred to as the *code rate*, and specifies the amount of overhead added by the code. For example, for a code rate of  $\frac{1}{2}$ , two channel bits are produced for each input information bit (100% overhead). This process is illustrated in the following figure.

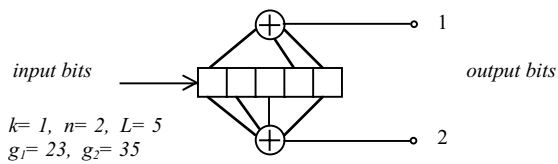


Figure 6. Convolutional Encoder Structure

Specifying a convolutional encoder in Commsim 2001 is very simple. The Convolutional Encoder block will automatically generate the required encoder structure based on the user supplied generator coefficients and other block parameters. The example below is for a commonly used  $R=1/2, k=7$  code.

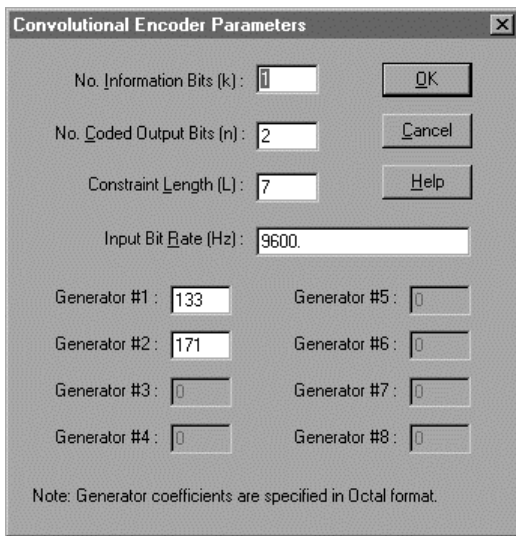


Figure 7. Convolutional Encoder Setup Diagram

## Trellis Codes

Trellis coding differs from other coding methods because it combines into one operation the data encoding and constellation mapping steps. For this reason, the term *Trellis Coded Modulation* is often used. The output of a trellis encoder is a baseband (I, Q) pair (i.e. a modulator constellation point). Unlike other codes, the transmitted channel symbol rate is often equal to the information symbol rate. This quality is highly desirable when the available channel bandwidth is limited. The coding overhead is hidden in an increase in the number of constellation points available to the transmitter. For example, the trellis coding approach used within the V.32 modem standard maps a 16-ary input word (4 bits) to a 32 point signal constellation (5 bits). In this fashion, a 25% coding bit rate increase is achieved without a corresponding increase in the necessary channel bandwidth. As can be easily shown, signal bandwidth is related to the transmitted symbol rate, as opposed to the channel bit rate.

The trellis mapping is specified in Commsim 2001 via an external input file. This file specifies the modulator IQ outputs and the encoder “new state” value for each possible input value and “current state” combination.

The example below illustrates a few lines from the Commsim 2001 V.32 Trellis Map file.

V.32 Trellis Map File					(1st header line)
4	5	8			(k, n, # states)
State	Input	New State	I out	Q out	(2nd header line)
0	0	0	-4	1	
0	1	0	0	-3	
...	...	...	...	...	
0	15	2	-2	1	
1	0	2	-4	1	
...	...	...	...	...	
7	15	7	-1	4	

In this example there are four input bits ( $k$ ), indicating an input symbol range of 0 to 15. The number of coded output bits is five ( $n$ ), which specifies one of 32 different constellation points. This particular trellis structure has eight states (0 - 7).

### Data Interleaving

Most codes that have been developed perform well under uniform channel error conditions, as those produced by a Gaussian channel. When channel errors tend to occur in bursts, however, additional steps are required to maximize the performance of the coding process. One common technique is to scramble the channel bits prior to transmission by using an Interleaver.

Data interleaving alters a coded bit stream so as to minimize the effects of burst type channel degradations. By using this technique, adjacent bit errors caused by signal fades or bursts of interference are spread out in time across a longer interval, giving the decoder a greater probability of recovering the original data stream. The following figure illustrates the operation of a very simple interleaver.

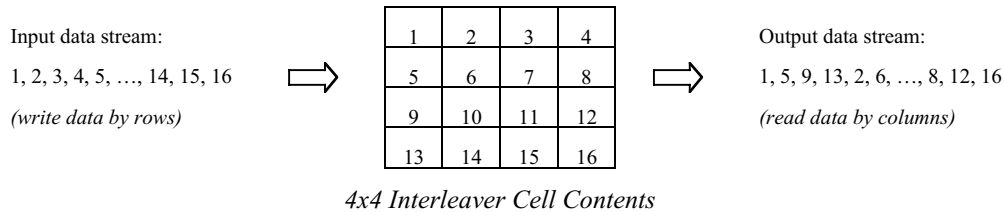


Figure 8. Data Interleaver (4x4)

## Modulation

In order to send an information signal from one point to another, the signal must often be altered prior to transmission. This process is commonly known as Modulation. There are many reasons why it is desirable for a new signal to “represent” the original waveform. The most common reasons relate to bandwidth allocation and signal recovery considerations. The modulation process, for example, allows the original baseband information signal to be “translated” in frequency so that many signals can coexist simultaneously without interfering with each other. This is achieved by allocating each modulated signal to a slightly different region of the available frequency spectrum.

Many types of modulation techniques have been devised for representing an information signal as it is being transmitted. In general, all modulation schemes rely somehow on varying the amplitude, phase or frequency of a “carrier” waveform. At a high level, modulation techniques can be subdivided into two basic groups: analog modulation and digital modulation.

## Analog Modulation

In analog modulation, the transmitted signal can be varied continuously over a specified range as opposed to assuming a fixed number of predetermined states. Examples of analog modulation techniques include Amplitude Modulation (AM), Phase Modulation (PM), and Frequency Modulation (FM). As the name implies, an AM transmitter operates by varying the amplitude of the carrier according to the voltage of the input signal. In a PM transmitter, the input signal is used to control the instantaneous phase of the carrier phase. With FM, on the other hand, the input signal is used to vary the instantaneous frequency of the carrier. In all three cases the input signal is an analog signal, such as a voice signal.

The following Commsim example compares the AM and FM modulation techniques using an arbitrary input waveform. The same carrier frequency is used in generating both the AM and FM waveforms.

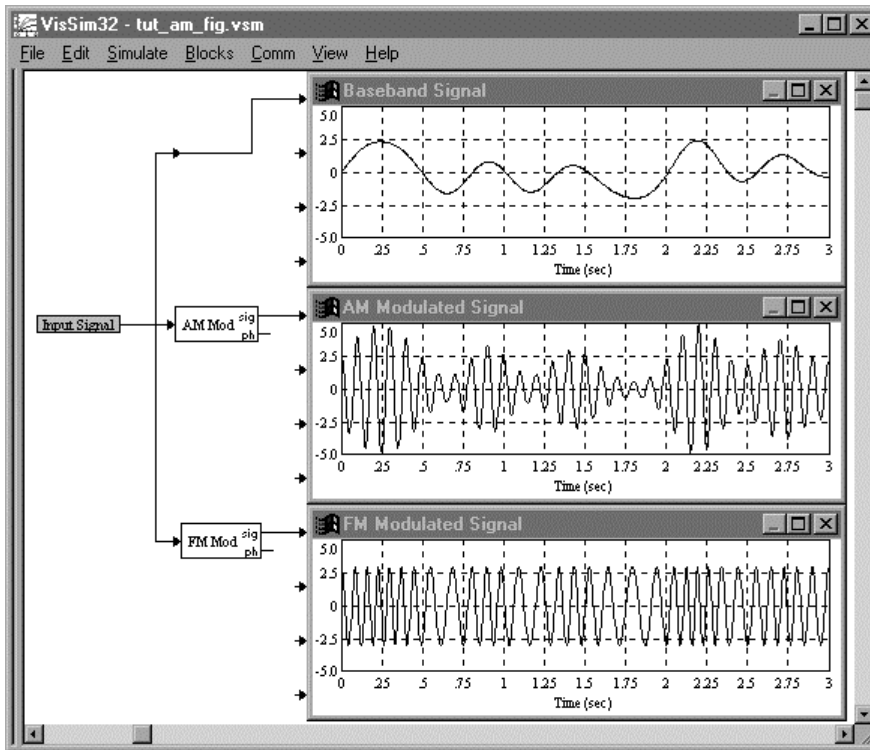


Figure 9. AM and FM Analog Modulation Formats

## Digital Modulation

In digital modulation, the transmitted signal can assume only a fixed number of predetermined states, usually referred to as the “alphabet size” or “constellation size” of the modulated signal. These include discrete amplitude levels, discrete phases, discrete frequencies, or combinations of the above. Examples of digital modulation techniques include Phase Shift Keying (PSK), Quadrature Amplitude Modulation (QAM), Frequency Shift Keying (FSK) and Pulse Position Modulation (PPM). Each of the above techniques can be implemented at various levels of complexity depending primarily on the total number of distinct states (constellation points) which are allowed within the modulator.

Digital modulation has inherent benefits over analog modulation as its distinct transmission states can more easily be detected at a receiver in the presence of noise than an analog signal which can assume an infinite number of values. When the transmitted signal originates as an analog waveform, a tradeoff occurs at the encoding stage since some information is always lost in the quantization process.

The following Commsim example compares two digital modulation techniques using an arbitrary digital input signal. The top plot shows the baseband input signal which can assume one of four specified levels [0 – 3]. The additional two plots show the difference between the output of a PSK modulator and an FSK modulator given the same input signal. PSK and FSK are described in more detail later in this section.

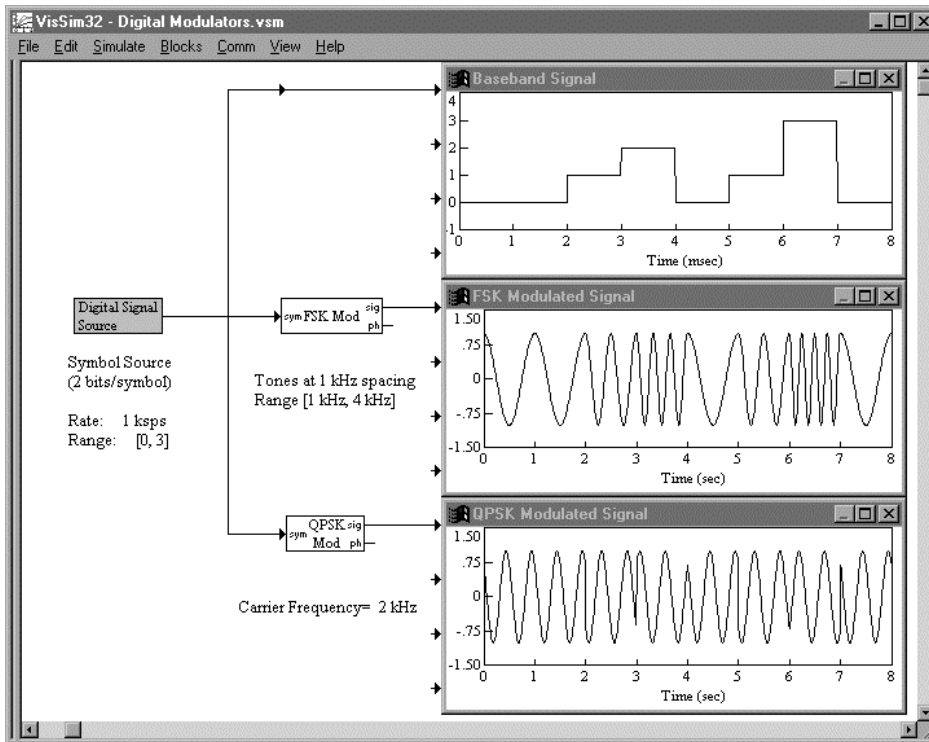


Figure 10. PSK and FSK Digital Modulation Formats

### FSK and MSK Modulation

In Frequency Shift Keying (FSK) modulation, the digital information is transmitted by assigning discrete output frequencies to each of the possible input symbols. The carrier amplitude remains constant. Two classes of FSK modulators exist: those which maintain a continuous carrier phase between states, and those which do not. The first case applies when a frequency synthesizer is used to generate the modulated output. The latter occurs when multiple independent oscillators are used to generate the various FSK tones. The bandwidth occupied by an FSK signal is directly proportional to the signaling rate. Commsim 2001 provides a basic FSK modulator block and also an MSK modulator (Minimum Shift Keying) which represents a special case of continuous phase FSK where the two output tones are spaced exactly by the symbol rate  $R$ . An example of FSK modulation is shown in Figure 10.

### PSK Modulation

In PSK modulation, the digital information is transmitted by varying the carrier phase between known phase states. The carrier amplitude remains constant. Among its desirable properties are its constant envelope characteristic, as all constellation points have equal power. The bandwidth occupied by a PSK signal is directly proportional to the symbol rate. Thus as the constellation size increases, no additional bandwidth is required. Of course, as the constellation points move closer together, higher power is required to maintain a given BER performance level.

Commsim 2001 offers the following PSK formats: BPSK, QPSK, SQPSK, 8-PSK, and 16-PSK. The user is given complete control over the mapping from input value to output constellation point. The examples below illustrate the signal constellations in the (I, Q) plane for QPSK and 8-PSK using a Gray encoded constellation mapping. Such a mapping ensures that neighboring constellation points differ by only one bit.

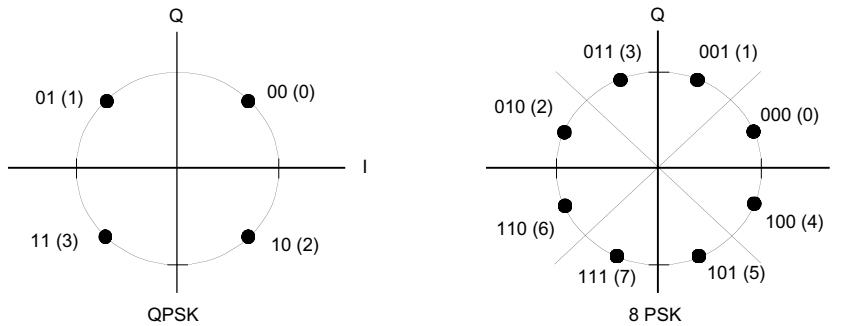


Figure 11. PSK Constellations for QPSK and 8PSK

An example of a BPSK modulated waveform, and the spectrum of a QPSK signal are shown in the illustrations below. The null to null bandwidth of a PSK signal is  $2R$ , where  $R$  represents the *symbol* signaling rate. For QPSK, the symbol rate is  $\frac{1}{2}$  the bit rate, as each constellation point represents two bits. The output spectrum is centered about the modulator carrier frequency  $f_c$ .

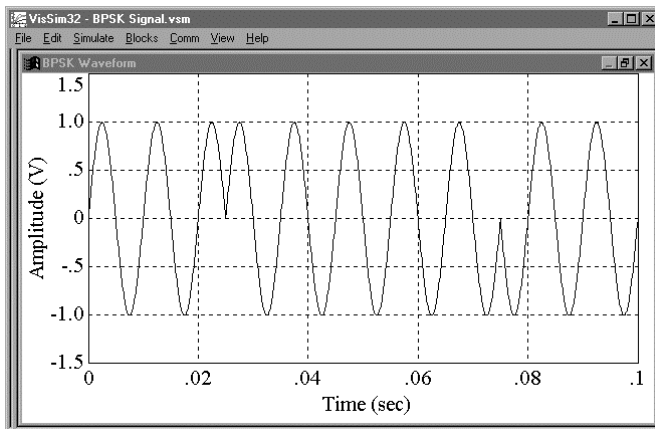


Figure 12. BPSK Modulated Waveform

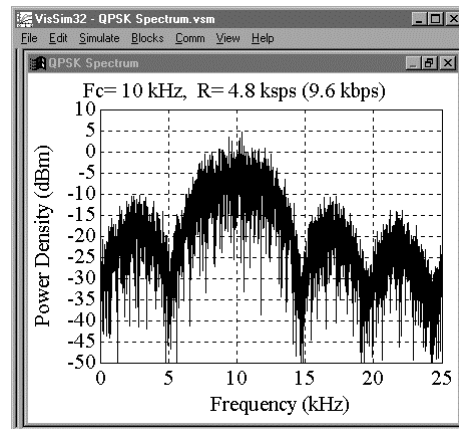


Figure 13. QPSK Spectrum (10 kHz Carrier)

### QAM Modulation

In QAM modulation, the digital information is transmitted by varying both the carrier phase and carrier amplitude between known constellation points in the (I, Q) plane. While QAM signals are not constant envelope, they make very efficient use of available bandwidth, with constellation sizes up to 256 points being common. As with PSK signals, the null to null bandwidth of a QAM modulated waveform is  $2R$ , where  $R$  represents the *symbol* signaling rate. For 16-QAM, the symbol rate is  $\frac{1}{4}$  the bit rate as each constellation point represents four bits. The output spectrum is centered about the modulator carrier frequency  $f_c$ .

Commsim 2001 offers the following QAM formats: 16-QAM, 32-QAM, 64-QAM, and 256-QAM. The user is given complete control over the mapping from input value to output constellation point. The following example illustrates the signal constellation in the (I, Q) plane for 16-QAM using a Gray encoded mapping.

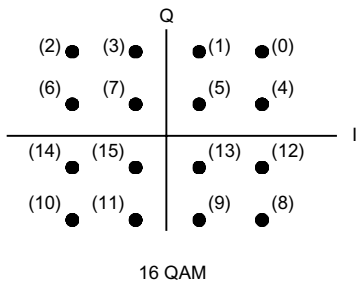


Figure 14. 16-QAM Constellation

### PAM Modulation

In pulse Amplitude Modulation (PAM) modulation, also known as Amplitude Shift Keying (ASK), the digital information is transmitted by varying the carrier amplitude between known discrete levels. The following PAM formats are offered in Commsim 2001: 4-PAM and 8-PAM. The user is given complete control over the mapping from input value to output constellation point. The example below illustrates the signal constellation in the (I, Q) plane for 8-PAM using a Gray encoded constellation mapping.

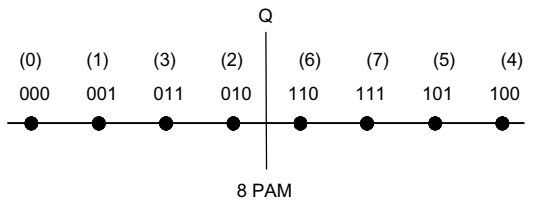


Figure 15. 8-PAM Constellation

### Differential Modulation Formats

Commsim 2001 also offers several differential modulation formats such as DQPSK and  $\pi/4$ -DQPSK. In addition the user can easily convert any digital modulator into a differential modulator by appropriately modifying the input bit stream(s) into the transmitter.

Differential modulation schemes are used when coherent phase detection cannot be readily implemented. Rather than mapping each information symbol to a particular carrier phase state, a differential modulator will map the input signal bits into carrier phase *transitions*, which from a detection point of view can be more easily identified. The table below illustrates the mapping from input symbol to output phase transition for both DQPSK and  $\pi/4$ -DQPSK.

<i>Input Symbol</i>	<i>DQPSK</i>	<i><math>\pi/4</math>-DQPSK</i>
0	0°	45°
1	90°	135°
2	-90°	-45°
3	180°	-135°

Figure 16. DQPSK and  $\pi/4$ -DQPSK Phase Transition Mapping

### Pulse Position Modulation

In Pulse Position Modulation (PPM) the information is transmitted by varying the occurrence of a rectangular or shaped pulse within a pre-defined symbol frame. The location of the pulse is proportional to the input signal level. This scheme requires a synchronized transmitter and receiver to maintain symbol frame timing. As shown in the following illustration, the pulse width can be varied from relatively narrow to wide enough that adjacent pulse positions overlap.

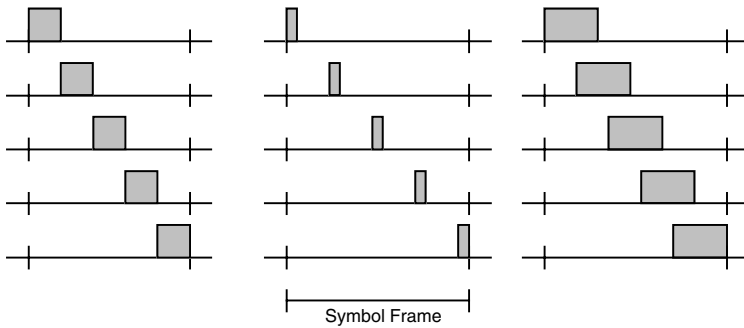


Figure 17. PPM Implementations (5 level scheme shown)

## Channels

In order to send information from one point to another, a transmitted signal must travel through a medium to reach the receiver. This path from transmitter to receiver is generally referred to as a *channel*. Examples of channels include a copper wire, fiber optic cable, or free space. Relevant characteristics of a channel include its propagation delay and any distortions it may add to the transmitted signal.

Commsim 2001 includes a variety of predefined channel models supporting both fixed and mobile service scenarios. Included are fading, multipath, band-limited and gaussian noise models. In each case, the engineer has the ability to modify key model parameters to suit their specific needs. The engineer can also expand upon the models provided by forming new compound blocks or creating custom C code implementations.

### Additive White Gaussian Noise Channel

A channel model used very often in the analysis of communication systems is the Additive White Gaussian Noise (AWGN) channel, or more simply the gaussian noise channel. In this model, noise power of a uniform spectral density (white noise) is added to the original signal. The resulting noise distribution is a zero mean gaussian process. While not always realistic, this assumption greatly simplifies the mathematics associated with estimating the performance of a given communication system. In fact, most Bit Error Rate (BER) curves are generated with the assumption of a gaussian noise channel.

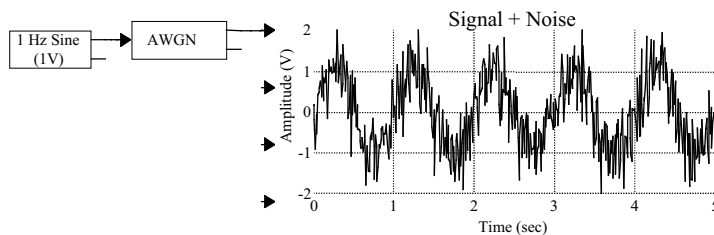


Figure 18. Adding Gaussian Noise to a Signal

### Fading Channels

Another frequently used class of channel models are those which simulate signal fading. Fading channels are typically produced by scattering or multipath processes, and are characterized by a randomly varying received signal amplitude and phase. The fading process is considered frequency-nonselctive when the signal bandwidth is much smaller than the coherence bandwidth of the channel, i.e. when all spectral components within the signal bandwidth are affected equally by the channel. When the received signal is

comprised solely of uncorrelated scattered components, the fading process is described as Rayleigh. When a strong signal component is present in the received waveform, such as from a direct signal path or a fixed reflector in the medium, the fading process is described as Ricean.

To model a fading channels, the input signal is typically multiplied by a single complex random variable with a Rayleigh or Ricean amplitude distribution, and uniform phase. Commsim 2001 provides two types of fading channel models: the Jakes mobile channel and a pure scattering based Rayleigh/Rice fading channel.

The Jakes fading channel is commonly used when modeling mobile communications systems. This model approximates a Rayleigh fading process via summation of multiple complex sinusoids. In this case, the fading process is a result of the velocity of the mobile and the associated varying multipath. Model parameters include the maximum doppler frequency and the number of sinusoids to be used.

The Rayleigh/Rice fading channel simulates tropospheric or ionospheric scattering in a communications link. The fading process is spectrally shaped according to the specified rms doppler spread. The inverse of this value denotes the approximate Coherence Time of the fading process. Note that in this case the doppler spread (spectral broadening) is due to the time domain amplitude fluctuations of the signal and not to any relative motion between the transmitter and receiver. Block parameters include the Rice Factor, the RMS Doppler Spread Bandwidth, and the RMS Fading Loss. The Rice Factor controls the proportion of direct to scattered signal power in the channel. The example below illustrates the effects of Rayleigh fading on the received power of a signal.

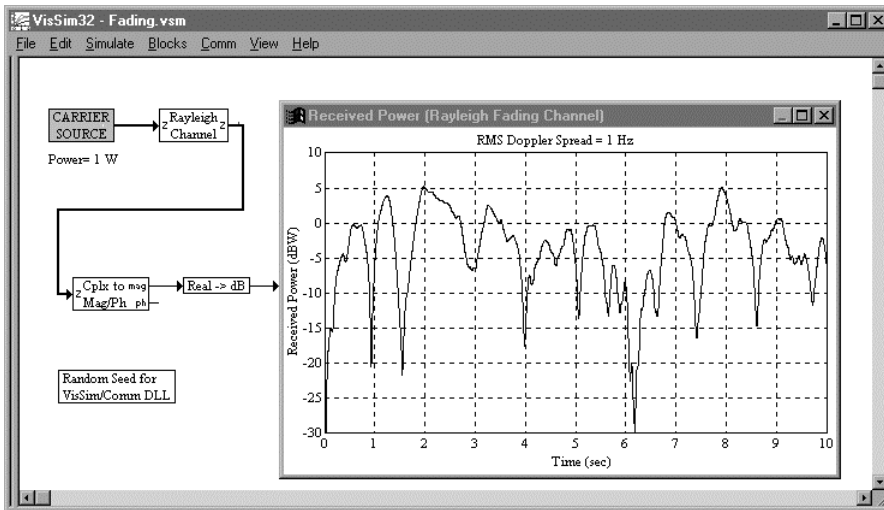


Figure 19. Example of Rayleigh Fading

### Multipath Channels

Another commonly used model in signal propagation is the Multipath channel. Multipath channels are characterized by multiple versions of the same signal arriving at a receiver with slightly different delays and relative signal amplitudes, thus interfering with each other either destructively or constructively.

Commsim 2001 includes the Rummler multipath fading channel, which is commonly used to model digital microwave links. The Rummler multipath channel models the interaction between a direct path signal and a reflected path signal. The reflected signal affects both the amplitude and phase of the received signal. Block parameters include the reflected path delay, reflected path relative gain, overall path attenuation, and the frequency of the fade null. The user can also easily create custom multipath channels, as shown below, by simply adding together multiple delayed versions of a signal and assigning each random amplitude weights.

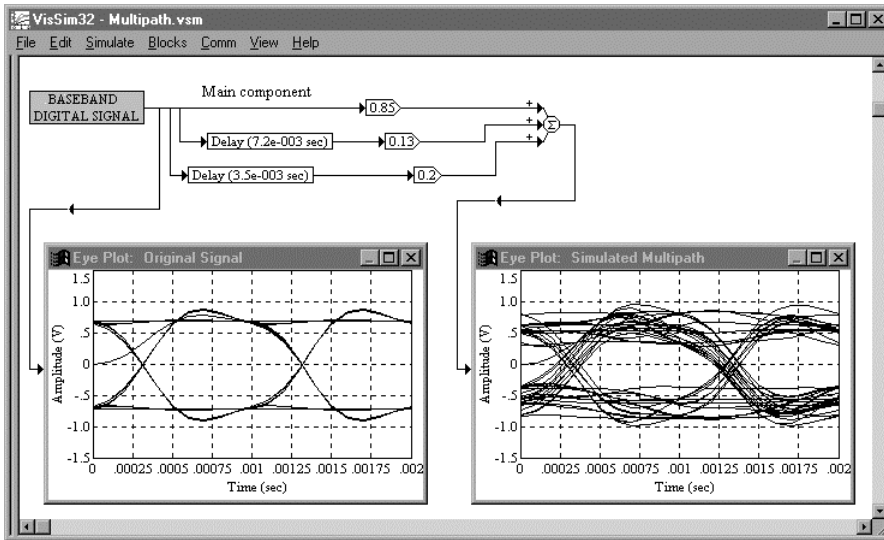


Figure 20. Implementing a Custom Multipath Channel

### TWTA Channel

Several models have been developed to simulate the nonlinear behavior of high power amplifiers, and in particular the Traveling Wave Tube Amplifier (TWTA). The TWTA channel includes both amplitude conversion (AM/AM) and phase conversion (AM/PM) effects. AM/AM conversion maps power fluctuations in the signal envelope into gain variations, while AM/PM conversion maps power fluctuations in the signal envelope into carrier phase rotation. The TWTA model provided by Commsim 2001 is based on a model by Saleh and allows the user to specify the tube operating point, and the average input signal power. Shown below are representative AM/AM and AM/PM curves as a function of tube operating point (backoff level).

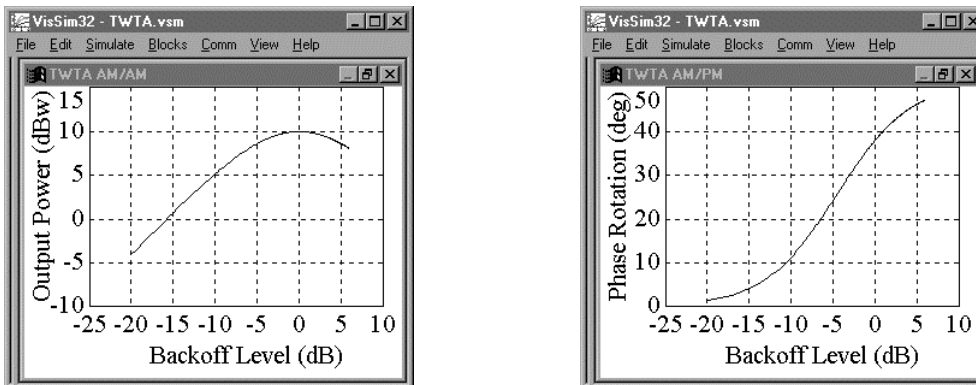


Figure 21. TWTA AM/AM and AM/PM Curves

### Binary Symmetric Channel

When simulating the transmission of digital data streams, a frequently used mathematical channel model is the binary symmetric channel (BSC). This channel model assumes that all bit errors are equally likely and that errors occur with a fixed probability which is specified by the user. As illustrated below, a transmitted “1” bit has a probability  $p$  of being received incorrectly as a “0”, and a probability  $[1 - p]$  of being received correctly, where  $p$  is typically a small number. This channel model is useful when the engineer is not interested in modeling system behavior below the digital bit level, as for example when evaluating the performance of a coding scheme at a specified channel error rate.

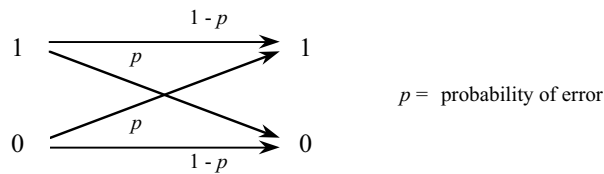


Figure 22. Binary Symmetric channel

## Demodulation

Once a signal arrives at the receiver, several steps are required before the transmitted signal can be recovered. The received signal is usually first filtered and amplified so that only the portion of the spectrum containing the desired information bearing signal is passed. Quite often an automatic gain control (AGC) circuit is also included to obtain a specified input level within the receiver. The signal is then demodulated, which loosely refers to the process of recovering the original baseband waveform which was modulated. The demodulation process is dependent on the modulation format used at the transmitter, and often involves multiple steps.

With most analog modulation formats, the transmitted information signal can often be recovered in a single step. Examples include recovery of an AM signal by envelope detection, or the use of a phase locked loop to recover an FM signal. Digital modulation formats, on the other hand, usually require multiple steps. At a minimum, these include: carrier recovery, symbol timing recovery, and symbol detection. The detection circuit is often referred to as a slicer. More complex systems may also include adaptive equalization or echo cancellation. A block diagram illustrating a candidate QAM digital receiver structure is shown below.

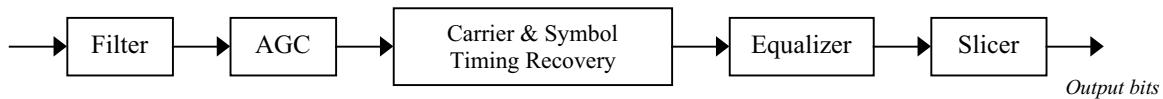


Figure 23. Sample QAM Receiver Block Diagram

Commsim 2001 includes all the necessary building blocks to create suitable receivers for any of the supported modulation formats. These include an integrate and dump block, phase locked loop elements, equalizer blocks, detector blocks for each of the digital modulation formats (slicers), an A/D block, and a variety of filter blocks to implement matched filter based receivers.

In general, whether analog or digital, two main classes of demodulation exist: coherent demodulation and non-coherent demodulation. Coherent demodulation requires the receiver to achieve synchronization, from a carrier phase point of view, with the transmitted carrier. Non-coherent detection does not impose this requirement. Coherent detection typically exhibits a 3 dB advantage over non-coherent approaches at the expense of greater receiver complexity.

### Coherent Demodulation

Most digital phase and amplitude/phase modulation techniques, such as PSK and QAM, are best received via the use of coherent detection. This is because the transmitted information is represented by the instantaneous phase (amplitude also for QAM) of the carrier.

Unless a direct means of obtaining the carrier phase is provided, such as via a coherent pilot tone or residual carrier, the receiver must estimate the carrier phase by observing the received signal itself. This process is known as carrier recovery and is performed using a carrier tracking loop. One example of such a circuit is the Phase Locked Loop (PLL).

In a PLL, a phase detector is used to estimate the phase error between the received carrier signal and the output of a local voltage controlled oscillator (VCO) also at or near the carrier frequency. This error signal is

appropriately averaged using a Loop Filter and is then used in feedback to control the VCO. Depending on the type of modulation used, the design of the phase detector will vary accordingly. The following illustration depicts a type of PLL, known as a Costas loop, being used to track a BPSK modulated signal. This particular example employs a second order PLL, which is able to track a phase or frequency step, but not doppler rate. For cases where doppler rate (varying frequency offset) is a factor, a third order PLL is recommended. The order of the PLL is usually determined by the type of loop filter employed.

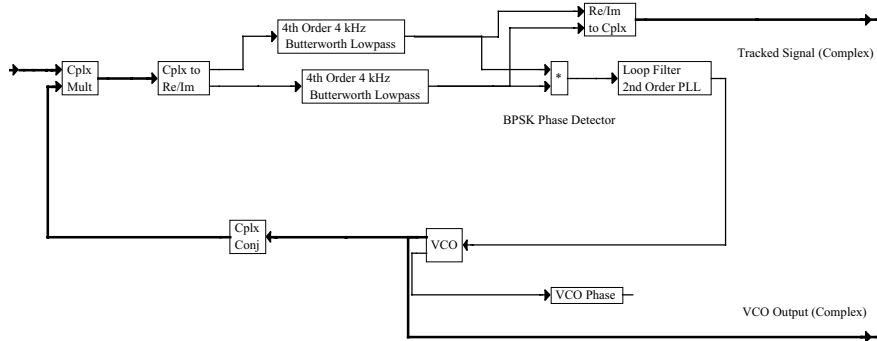


Figure 24. Commsim 2001 Implementation of a PLL

### Non-Coherent Demodulation

In non-coherent detection, an estimate of the carrier phase is not required by the receiver. Examples include FSK and Differential PSK (DPSK). When demodulating an FSK signal, the receiver must decide which of  $M$  orthogonal frequencies is being received; this can be accomplished using multiple individual tone detectors and without the need for any signal phase information. Similarly, in a DPSK receiver, an absolute estimate of the carrier phase is not required since the information is represented by the changes in carrier phase. A non-coherent receiver need simply compare the signal phase during a symbol period to the phase during the previous symbol period to make a decision.

While non-coherent demodulation approaches are not as efficient as coherent approaches (~3 dB loss), their ease of implementation makes their designs very attractive from a cost point of view.

## Signal Decoding

If signal encoding was used at the transmitter, an additional decoding step is required at the output of the demodulator before the original information signal can be recovered. The decoding step requires knowledge of the encoding process used at the transmitter. The signal decoding process is usually more complicated than the encoding process and can be very computationally intensive. Commsim 2001 provides decoders for both convolutional and trellis based codes, as well as a de-interleave block.

In addition to different encoding and decoding schemes, an additional distinction is made between “hard decision” and “soft decision” decoding. In hard decision decoding, the encoder is provided with “hard” bit decisions by the demodulator. This means that the decoder either gets a “0” input or a “1” input for each received channel bit. In soft decision decoding, on the other hand, the decoder is passed a multi-scaled value representative of the received channel bit (e.g. a level from 0 – 7). This additional information provides the decoder with a level of confidence metric associated with each received bit. A good rule of thumb is that a soft decision implementation has a performance advantage of about 2 dB over the hard decision approach.

### Viterbi Decoding

The decoding of convolutionally encoded data is commonly accomplished using the Viterbi algorithm. This algorithm builds a finite trellis of all possible received bit sequences and maintains a cumulative metric for each path within the trellis. As each new channel bit is detected, the cumulative metrics for each path are

updated based on a look-up table determined by the code generator coefficients. After a specified “truncation length” the decoder selects the path with the best metric as the winner and outputs a bit decision. Typical truncation lengths are ~5 times the constraint length of the code (*see encoders*). For a constraint length 7 code, an acceptable truncation length is therefore 35 bits. This means that the decoder operates with a delay of 35 bit periods.

Commsim 2001 provides both hard decision and soft decision Viterbi decoders. The primary difference between the two is that the hard decision block expects a digital bit stream as input from the previous stage (e.g. bit detector), while the latter expects an analog input signal. The signal quantization step is carried out within the soft decision block, which includes the number of A/D quantizer bits as a parameter.

## Trellis Decoding

Trellis decoding is used to decode a trellis modulated signal. As was mentioned in the *Encoders* section, trellis coded modulation combines in one step the coding and constellation point mapping. Because of this, a trellis decoder operates at the constellation mapping level in the IQ plane. The trellis decoder is provided with the raw detected I and Q channel outputs. Internally, a metric (mean square distance) is computed for each possible constellation point based on the received IQ input pair. These values are used in turn to update cumulative metrics for each possible sequence of states within the trellis. As with convolutional codes, the Viterbi algorithm is used to keep track of all possible paths within the trellis and select a winning path.

The trellis decoder block makes use of the same mapping file used to specify the trellis code at the encoder. Details of this file were presented in the *Encoders* section.

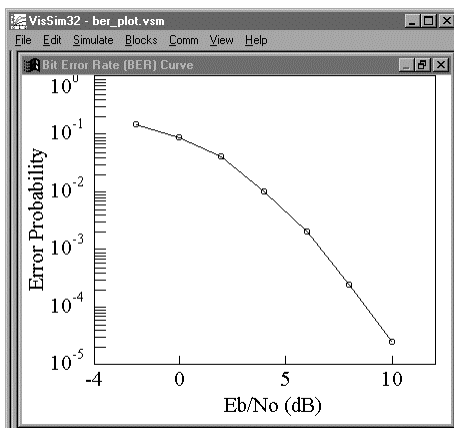
## Signal Sinks

Simulation results can be immediately visualized in Commsim 2001 through its many graphic output capabilities. Most outputs are viewed using the Commsim Plot block. In its basic mode, the Plot block displays up to four time domain signals. Most of the examples presented so far in this tutorial included the use of Commsim plot blocks to display various time domain signals.

When properly configured, the Plot block can also produce many of the more specific outputs used in the field of communications. These include BER plots, phase scatter plots, eye diagrams, spectral plots and other visual outputs.

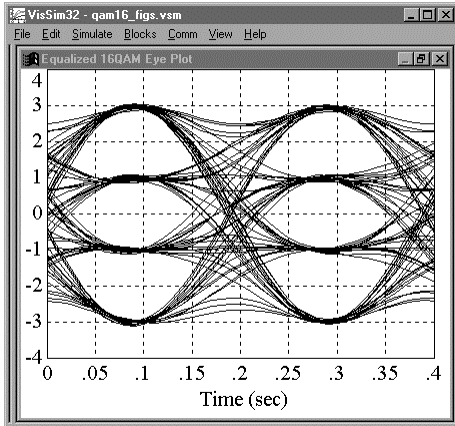
## BER Curves

Bit Error Rate (BER) curves are often the final indicator of performance in communication systems. They display the required Signal to Noise Ratio (SNR), usually expressed in  $E_b/N_o$ , necessary to achieve a specified bit error rate at the receiver. Shown below is a BER curve example for a QPSK modulated signal.



## Eye Plots

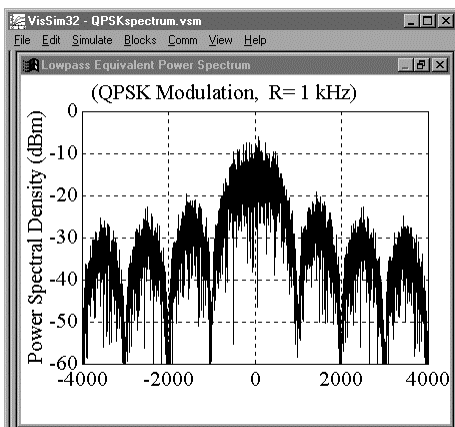
The eye plot simulates the use of an oscilloscope and is particularly useful for analyzing digital data waveforms. Its applications include estimating signal SNR, detecting amplitude compression and observing the effects of Inter-Symbol Interference (ISI). Shown below is an I channel eye plot of a 16-QAM modulated signal. The eye plot span, in this case, is two symbol periods.



## Frequency Domain Plots

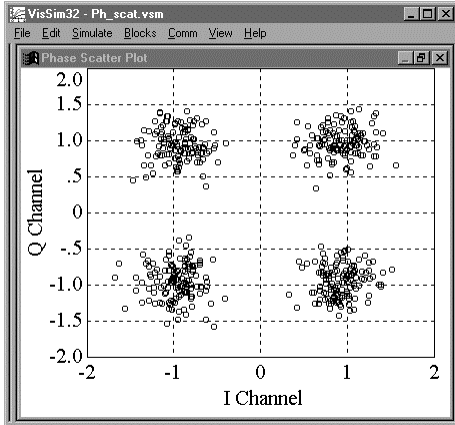
Frequency domain plots are generally used to view the power spectrum associated with a time domain signal. They can also be used to view the magnitude and phase response (transfer function) of a filter or system. Two methods exist within the Commsim 2001 environment for viewing frequency domain plots. All plot blocks have the basic ability to show a frequency domain FFT representation of any time domain signal after a simulation run has been completed. In addition, a special Complex FFT block is provided which can calculate a complex FFT or IFFT during a run.

Shown below is the power spectrum associated with a QPSK modulated signal.



## Phase Scatter Plots

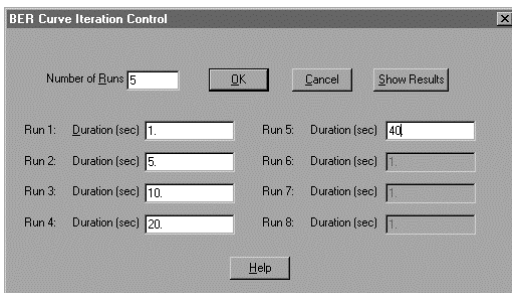
Phase scatter plots are used to view a received signal constellation in (I, Q) space, and observe the impacts of channel noise and/or distortions. Below is an example of a QPSK constellation phase scatter plot. The effects of noise are evident in the “cloud” observed at each of the four constellation point locations.



## Generating BER Curves

Bit error rate curves can be easily generated using Commsim 2001. When a BER Curve Control block is added to a simulation, it becomes possible to execute up to eight consecutive runs of the same simulation, each with a different simulation duration and signal to noise ratio (specified using an AWGN block). At the end of the simulation, the BER curve results are reported via a screen message. The BER curve can also be displayed by using an appropriately configured Plot block.

The BER Curve Control block resides in the Comm/Estimators menu. Block setup parameters include the overall number of runs and the duration of each run in seconds. The block's three outputs are intended to drive a Plot block configured in XY mode and External Trigger.

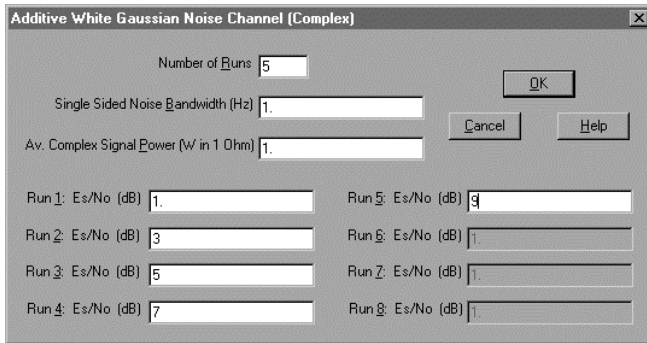


In order to enable multiple consecutive runs the Auto Restart parameter in the Commsim Simulation Setup dialog must be activated. The Range End parameter must be equal to, or exceed, the longest duration specified in the BER Curve Control block setup dialog.

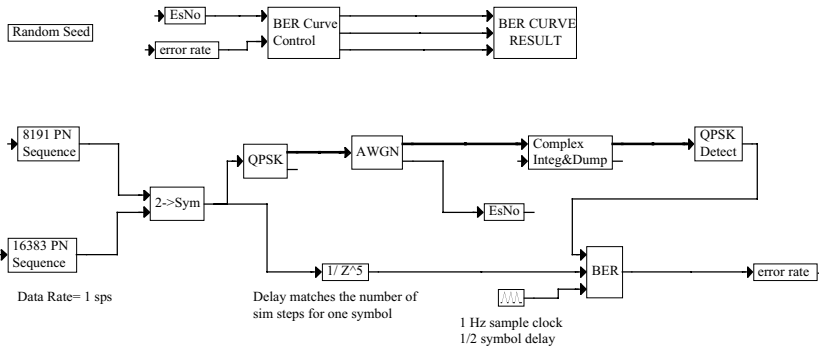
The actual BER measurement for each run is calculated by using a Bit Error Rate block, which is found in the Comm/Estimators menu. This block compares an original data sequence to a received data stream. It is very important that the original sequence be appropriately time shifted to compensate for any delays introduced by filters or an integrate and dump operation. The block's error rate output is connected to the BER Curve Control's top input.

Varying the signal to noise ratio for each run is done using the Additive White Gaussian Noise (AWGN) channel block. The AWGN Setup dialog box allows for the specification of up to eight individual signal to

noise ratios, corresponding to eight separate runs. The value of the Number of Runs parameter must match that specified in the BER Control block.

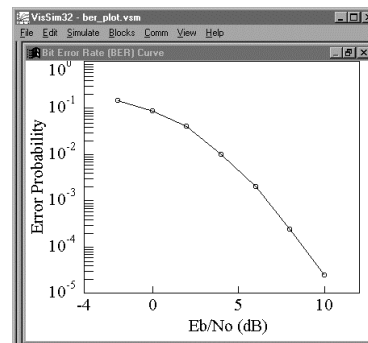
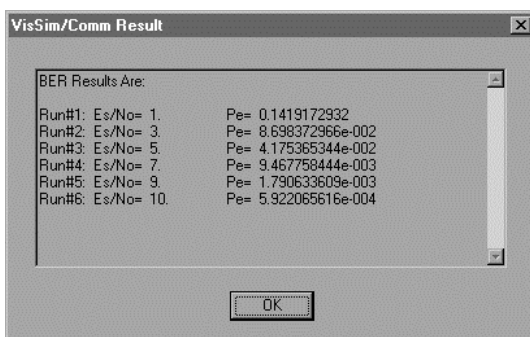


The following example illustrates generating a Bit Error Rate curve. This example represents a simplified end-to-end QPSK communication link where the carrier recovery loop has been omitted.



In the above diagram, two binary pseudo-random data sequences are combined to form a two-bit symbol, which is then modulated using QPSK. After adding channel noise (AWGN block), an integrate and dump operation is performed to obtain the received I and Q values. These are in turn passed to a detector which determines the closest constellation point. This final output is then compared, in the BER block, to the value originally sent. Note that the BER block's input clock signal is offset by 1/2 a symbol to sample at the data symbol's midpoint. The additional one symbol delay in the source data stream is required to achieve synchronization with the received output data. Had any filtering been included in the simulation, an additional signal delay would have been necessary.

The Bit Error Rate block's output is passed to the BER Curve Control block, as is the signal to noise setting ("EsNo") from the AWGN channel block. At the end of the simulation, the BER Curve Control block creates a message box summarizing the results of all runs, and outputs the BER Curve result as shown below.



## Lowpass Equivalent Systems

The sampling requirements of many simulations can be reduced using a technique known as *complex envelope* notation. When a data signal is modulated, a sampling rate of at least 8 samples per carrier cycle is usually selected in order for the simulation to accurately represent the carrier signal. This means that the simulation step size is dictated by the carrier frequency and not the data rate, which may be several orders of magnitude lower in frequency. When a simulation is transformed to a lowpass (or baseband) equivalent, its carrier frequency is mathematically translated to zero, and the simulation sampling interval can then be based on the sampling requirements of the data signal. The result is a reduction in execution time without a loss of simulation accuracy. When in complex envelope notation, both *real* and *imaginary* versions of a signal are used, often referred to the in-phase (I) and quadrature (Q) components of the signal. Most modulators and channel models in Commsim 2001 support the use of complex envelope notation.

The following is a brief mathematical description on the lowpass equivalent concept.

A modulated bandpass signal  $s(t)$  usually occupies a narrow band of frequencies near the carrier  $\omega_c$ , and can be represented as

$$s(t) = a(t) \cos[\omega_c t + \phi(t)] \quad (2.1)$$

where  $a(t)$  represents the amplitude and  $\phi(t)$  the phase of  $s(t)$ . After expanding, the above equation becomes

$$\begin{aligned} s(t) &= a(t) \cos \phi(t) \cos \omega_c t - a(t) \sin \phi(t) \sin \omega_c t \\ s(t) &= x(t) \cos \omega_c t - y(t) \sin \omega_c t \end{aligned} \quad (2.2)$$

$$x(t) = a(t) \cos \phi(t)$$

$$y(t) = a(t) \sin \phi(t)$$

where  $x(t)$  and  $y(t)$  are respectively the inphase and quadrature components of  $s(t)$ . Using complex exponentials, we can also write (2.1) as

$$s(t) = \text{Re}[a(t)e^{j(\omega_c t + \phi(t))}] = \text{Re}[a(t)e^{j\phi(t)}e^{j\omega_c t}]$$

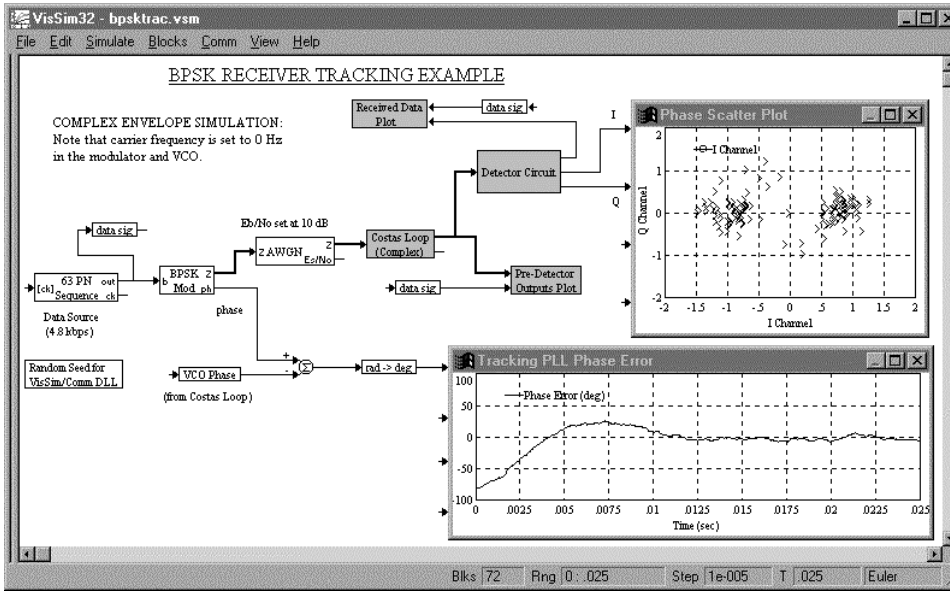
and

$$\begin{aligned} s(t) &= \text{Re}[u(t)e^{j\omega_c t}] \\ u(t) &= a(t)e^{j\phi(t)} = x(t) + jy(t) \end{aligned} \quad (2.3)$$

The signal  $u(t)$  above is defined as the *complex envelope* of  $s(t)$ . Note that  $u(t)$  and  $s(t)$  are directly related by a simple frequency translation. When simulating linear systems, it can be shown that as long as the bandwidth  $B$  of the complex envelope signal is much smaller than the carrier frequency, that is  $B \ll \omega_c$ , then the signal  $s(t)$  may be equivalently represented by  $u(t)$ . The primary advantage of using  $u(t)$  is that a much lower simulation sampling rate can be used. For more information on lowpass equivalent signals, see *Digital Communications* (J. Proakis, McGraw-Hill, 1989).

# Sample Communication Simulation

An example of a Commsim 2001 system level simulation is shown below.

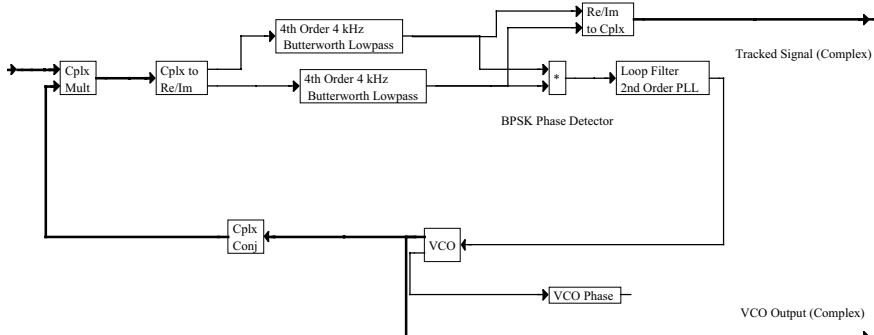


This example provides a simplified model of a transmitter, channel, and receiver connected in sequence to form an end-to-end system. In this example, a binary pseudo-random sequence at 4.8 kbps is used as the data source signal. The input data sequence is then passed to a PSK Modulator block to be BPSK modulated. This example implements a complex envelope (lowpass equivalent) simulation, so the carrier frequency is set at 0 Hz. The simulation step size is 10 milliseconds.

Once modulated, the complex signal is corrupted by using an Additive White Gaussian Noise (AWGN) channel block. The specified  $E_b/N_0$  is 10 dB. The corrupted BPSK modulated signal is then acquired and tracked at the receiver using a Costas Loop implementation. The tracking loop phase error and the received phase scatter plot are shown in the illustration.

The Random Seed block is used to specify a seed for all Commsim 2001 blocks which generate random values, such as the AWGN block. By specifying a seed, the results of a given simulation can be repeated. If the Random Seed block is omitted, different results will be obtained from run to run.

The receiver is implemented using two compound blocks: Costas Loop and Detector Circuit. Details of the Costas Loop hierarchical compound block are shown in the following illustration.



The purpose of the Costas Loop is to perform carrier recovery of the signal. The above implementation uses a 2nd order Phase Locked Loop (PLL). The input signal is complex multiplied with the complex VCO output (also at 0 Hz), and the resulting complex signal is filtered using a pair of 4 kHz Butterworth Low Pass Filters. The BPSK Phase Detector produces an error signal proportional to the phase offset between the incoming signal and the VCO output. The error signal is then passed to the Loop Filter, which performs an averaging function and drives the VCO. In this particular example, the VCO gain is set to 10 Hz/volt and the phase detector gain is approximately 1 rad/volt (a one radian phase error produces a one volt input to the loop filter).

The tracked complex signal from the Costas Loop is then passed to the Detector Circuit, which implements an integrate and dump operation followed by data detection. The output of this stage is used to generate the Phase Scatter Plot. Details of the detector are shown below. To keep things simple in this system level example, a data transition synchronizing circuit was purposely not shown. In a real system, this circuit would provide the necessary timing pulses to the integrate and dump circuit.

