C213: BIT ERROR ANALYSIS AND BEYOND

Communication Design Engineering Conference

In the past, bit error rate testing meant transmitting a pseudo-random sequence of bits through a channel under test and counting the number of incorrectly received bits at the receiving end. This technique was adequate because the basic question to be answered was, "is it working?" Today's communications systems are pushing the envelope further and further, trying to squeeze as much bandwidth out of a channel as possible, and the question has now become, "How can I make it work better?" Forward error correction, data compression, and sophisticated modulation codes are some examples of these techniques. In these systems, knowing the position of an error relative to the data being transmitted or relative to other errors, is crucial in order to optimize the efficiency of a given technique. This class discusses many techniques for bit-accurate analysis, and discusses the concept of an overall system error budget.

JIM WASCHURA
SYNTHESYS RESEARCH, INC.

Jim Waschura received a BA degree in math and computer science from Hiram College in northeast Ohio. He is a co-founder of SyntheSys Research Inc., which provides test equipment for digital channel development in the communications, tape recording, disk recording, and television industries.

TECH NOTES

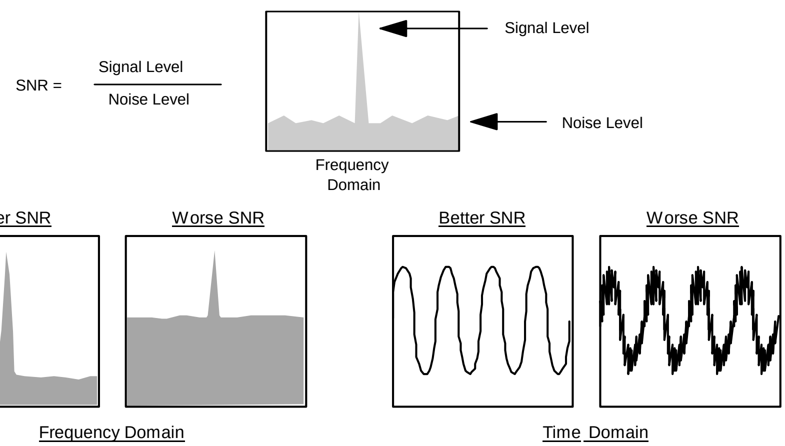
SYNTHESYS
RESEARCH, INC.
www.synthesysresearch.com
650 364-1853

INTRODUCTION

More and more innovative applications of digital communications are being engineered everyday, including digital cellular telephones, cellular TCP/IP, infrared desktop networks, digital cordless telephones, and direct satellite television broadcasts. Our existing communications infrastructure—made up of copper and optical land lines and terrestrial and satellite radio links—is constantly being pushed to provide higher bandwidth in support of the three big data applications: television, telephone, and computer networking. Thanks to CD players, modems, DAT recorders, modern cellular telephones, and direct satellite television broadcasts, “digital” is a now a household word.

Analog Performance Is Measured in SNR

Historically, designers have met the challenge of providing reliable communications systems on a case-by-case basis since each application had different requirements for accuracy and timeliness. These systems were generally implemented using analog techniques utilizing direct connections or specifically allocated spectra to transmit carrier wave-modulated signals with a goal of the highest signal-to-noise ratio (SNR) possible. The SNR easily summarizes the overall performance of an analog system by describing how strong the information is, relative to the unavoidable noise also present in the signal.



Digital Uses "Probability of Error" Performance Criteria

To achieve the higher degree of performance demanded today, communications system designers are utilizing techniques that make performance evaluation based solely on SNR impossible. These include techniques such as partial response signaling, adaptive equalization, forward error correction, and frequency spreading. These techniques have spawned new performance measurement criteria including eye diagram and constellation diagram analysis, jitter measurement, time interval analysis, and window margin testing. All of these techniques attempt to predict the statistical likelihood of encountering an error during communications. This predicted probability of error, P_E, can be verified by comparing it with the channel’s bit error rate (BER). The BER is measured empirically by counting the number of errors over an adequately long span of transmission.

Errors Are Related to Power and Noise

In digital communications systems (DCS), each bit is transmitted with a certain amount of energy, E_b, and the channel contains a certain amount of noise, N_0. In general, decreased E_b/N_0 values imply increased probability of error in the channel. The E_b/N_0 is often treated like the SNR of an analog channel—a single metric indicating performance.

Translating E_b/N_0 to P_E

There are difficulties with direct translation from Eb/N0 to PE. First, the mathematical models are becoming increasingly complex for sophisticated communications techniques. This has led some researchers to make erroneous performance claims[1]. The formula for describing the probability of error when transmitting NRZ bits along a piece of copper wire in the face of average noise level is very tractable. But systems with compression, forward error correction, adaptive equalization, advanced multibit encoding, frequency spreading, and carrier wave modulation, transmitted in environments with space noise, system noise, co-channel interference, multipath interference, and weather effects pose very challenging problems in modeling.

Secondly, the noise figure, N0, is generally modeled as a random process in mathematics, and sophisticated communications techniques are increasingly being faced with less random errors and more highly correlated errors. Accounting for these correlated errors, including error bursts and errors caused by discrete frequency interferences like multipath, are becoming the difficult part of communications system design, and single-parameter measurements like SNR and Eb/N0, and even BER for that matter, are not adequate for designing systems with instantaneous types of error conditions.

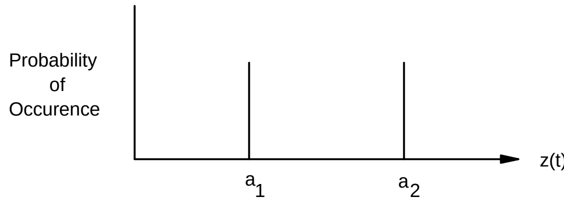
For a communications system designer to meet today's challenges head-on, a new performance measurement technique is required that can study precise error locations and report on error correlations.

THE DCS AND NOISE

Unlike analog communication systems, noise in a DCS is not inextricably linked to the transmitted signal. Selected analog waveforms are transmitted which represent symbols from a relatively small digital alphabet. Each waveform is transmitted for an amount of time equal to one symbol period. In the case of binary signaling, the transmitter issues one of two analog waveforms, $s_1(t)$ or $s_2(t)$, depending on whether the symbol represents a one or a zero.

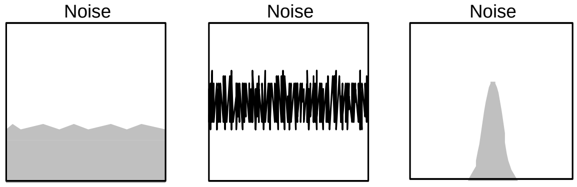
The job of the receiver is to recognize which of the waveforms was transmitted, and to assign either a one or a zero to the received symbol. To accomplish this, first the received signal is reduced to a single value function, z(t), which is evaluated at symbol period intervals. This is called the test statistic. For the case of unipolar baseband signaling, z(t) may be the voltage level at the center of the bit cell. For more sophisticated signaling, such as modulated carrier wave signaling, z(t) is the output of a correlator matching one of the two prototype waveforms.

Without noise, a received $s_i(t)$ will translate to a specific test statistic value, a_i. Transmitting an equally likely binary alphabet yields reception of similar quantities of a_1 and a_2 values. In this situation, the probability density function (PDF) shows that half of the time a_1 is received and half of the time a_2 is received.



Noise Makes Recognition More Difficult

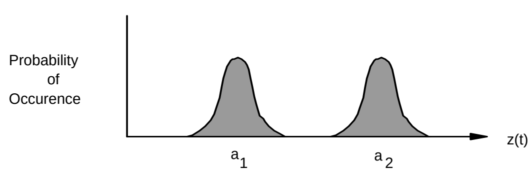
In the real world, noise is added to the transmitted signal. Man-made noise sources include sparkplug ignition noise, switching transients, and other radiating electromagnetic signals. Natural noise includes electrical circuit and component noise, atmospheric disturbances and galactic sources[2]. Natural noise is generally considered a Gaussian random process, and is modeled as a zero-mean random variable, n(t), which adds to or subtracts from the original signal level. It does not have a multiplicative effect[4]. In the frequency domain, noise has equal amounts of all frequency components.



The resulting signal that is received by the receiver is, therefore, the originally transmitted waveform plus the added noise.

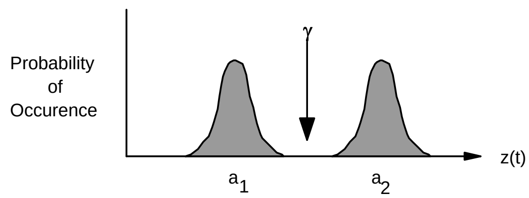
$$r(t) = s_i(t) + n(t) \qquad i=1,2$$

The addition of noise to the transmitted waveform will cause the receiver to transfer this noise into the domain of the test statistic. Assuming the determination of the test statistic is a linear process, the Gaussian noise in the transmission domain will produce a Gaussian random process in the domain of the test statistic[2]. In this situation, encountering pure a_1 and a_2 test statistic values will become less likely, and instead, each symbol will be represented by a Gaussian probability distribution, with a variance determined by the amount of noise in the system.

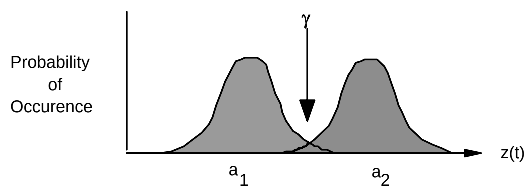


Noise Makes Errors in the DCS

Maximum likelihood receivers compare received z(t) values with an optimum threshold, γ, which is set midway between a_1 and a_2. In this fashion, small amounts of noise on the transmitted waveforms produce no digital errors because even with the additional noise in the derived test statistic, the entire probability density function for a_1 is on one side of the threshold, and a_2 is on the other side, and deciding ones and zeroes is easy.

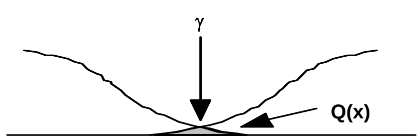


With more noise, however, the probability density functions for the two symbols can extend beyond the threshold, γ. This causes misdetection of symbols and subsequent bit errors.



Calculating P_E from Power and Noise Levels

The area of the tail portions of the probability distribution functions that fall on the wrong side of the γ threshold diagrammatically illustrate the probability of error, P_E. The integration of this region is called the complimentary error function or the Q(x) function, and represents P_E. This function is determined by the distance between a_2 and a_1 and the variance of the noise component, σ. A useful approximation for Q(x) is shown[2].

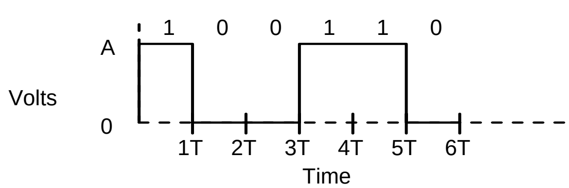


$$P_E = Q\left(\frac{a_1 - a_2}{2\sigma}\right)$$

The Q(x) function defines the relationship between bit energy level, noise level, and P_E.

Example of Q(x) for Unipolar NRZ

For example, in the case of unipolar baseband signaling, the transmitter transmits a voltage level of A to represent a one bit, and zero volts to represent a zero bit.

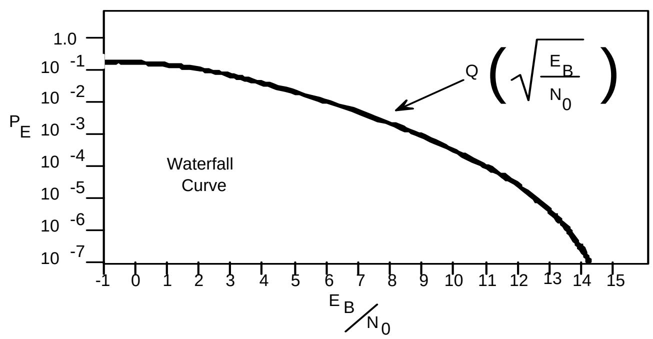


The energy per one-bit is the instantaneous power, A^2, times the duration of transmission, T. The energy per zero-bit is zero. Thus the average energy per bit $E_b=(A^2T)/2$. The energy of the noise in the system is characterized by the variance of the Gaussian probability distribution, σ, by the equation $N_0 = \sigma^2$. Now the Q(x) function can be evaluated as shown[2].

$$P_E = Q\left(\sqrt{\frac{A^2 T}{2 N_0}}\right) = Q\left(\sqrt{\frac{E_B}{N_0}}\right)$$

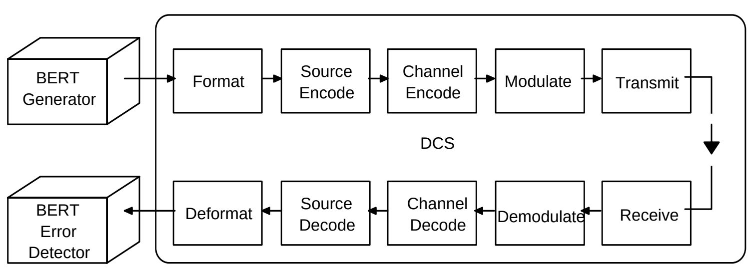
Waterfall Curves Relate E_b, N_0, and P_E

More sophisticated communications systems require more sophisticated calculations, and these techniques depend on the accuracy of modeling noise as a random process. This class of formulas defines relationships between E_b, N_0, and P_E. They are often drawn pictorially, and are called "Waterfall" curves. They are used to find expected error rates, given certain conditions of E_b and N_0.



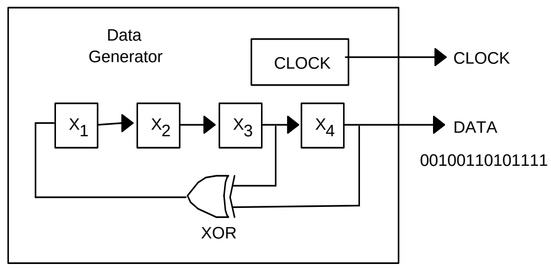
BIT ERROR RATE TESTING

The theoretical calculations for what P_E ought to be, based on E_b and N_0 conditions, are very useful especially in the planning stages of system design. Once a system is in place, however, evaluation of P_E is done by performing real measurements on the channel. This testing is called Bit Error Rate (BER) testing, and equipment which performs these tests is referred to as a Bit Error Rate Tester (BERT). These devices are often split into two main features: data generation and error detection. The data generator creates baseband data, which is given to the transmitter for transmission. The receiver receives the signal and recreates the baseband signal, which is then given to the error detector, which recognizes the data sequence and counts the number of errors it encounters.



Pseudo-Random Sequences are Used For Testing

The data generator creates a pseudo-random data sequence as a test pattern. Pseudo-random sequences are useful because they simulate signals with a wide range of frequency components and because they are easy to implement using shift registers and logic gates. A simple four-stage pseudo-random sequence generator is shown below. To initiate sequence generation, the four stages must be prepared with any value other than "0000". Seeding the shift register with all zeroes causes the circuit to always generate zero bits.



Pseudo-random data sequences are obviously not truly random. A bit error rate tester could not use truly random data because it wouldn't know what data to compare with. Instead, pseudo-random sequences are used, which have many of the same properties as truly random data, but repeat after a known quantity of bits. The properties of a maximal length pseudo-random data sequence are listed.

Balance Property — The number of 1's and the number of 0's differ by at most one.

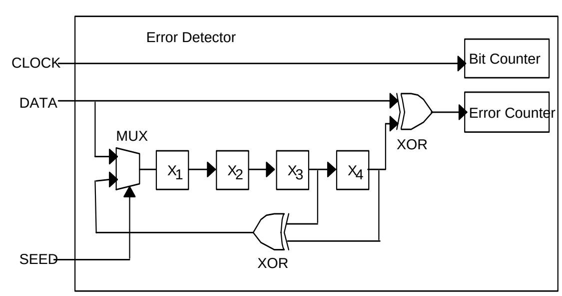
Run Property — Among the runs of 1's and 0's in the period, approximately half the runs of each type are of length 1, one-fourth are of length 2, one-eighth are of length 3, and so on.

Correlation Property — If the period of the sequence is compared term by term with any cyclic shift of itself, the number of agreements differs from the number of disagreements by not more than one count.

Maximal Length — When implementing the data generator using an N-stage linear feedback shift register, the number of bits in one period is equal to $2^N – 1$. This also means that every possible n-bit value is generated, except the n-zeroes case. A maximal length pseudo-random sequence will have at most n–1 zeroes in a row, and n ones in a row.

Error Detection

The error detector must recreate the pseudo-random sequence in the same phase as the received sequence in order to compare and count errors.

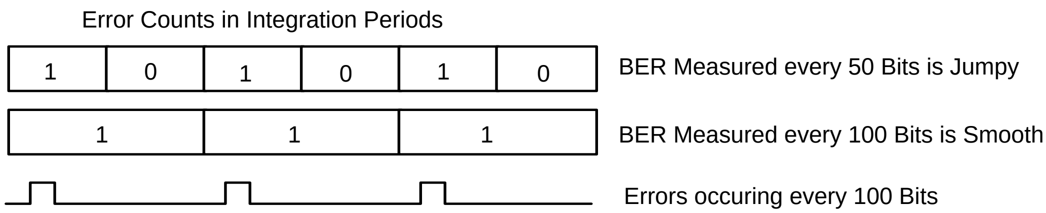


To do this, the error detector simply acquires bits from the incoming data stream and fills n-stages of a duplicate data generating circuit, and then clocks the circuit based on the received clock signal. Assuming there are no errors at the time of this "seeding" stage, the duplicate circuit generates the exact pseudo-random sequence in the exact phase as the received sequence. If an error occurs during the seeding step, initial comparisons will commonly fail, and this indication can trigger another seed attempt.

Calculating BER requires an Integration Period

The error detector produces a count of how many errors occurred and a count of how many bits have been tested. These are the basic components of the BER calculation, which is the number of errors divided by the number of bits tested.

To be meaningful, the BER calculation must be based on a minimum number of tested bits. During this accumulation period, errors are detected causing an error counter to increment. At the end of the period, the number of errors is divided by the number of bits in the accumulation period, and the BER is calculated. This accumulation period is often called an “integration period” because the errors are integrated over this time. The minimum integration period is set by the quantity of errors in the system. For example, if errors occur every 100 bits, calculating the BER every 50 bits would yield "jumpy" and misleading results.



A practical rule of thumb is to take the exponent of the expected BER, change its sign, add two, and integrate over that many bits. For example, a BER of $1x10^{-4}$ requires an integration period of at least $1x10^{6}$ bits. This ensures approximately two significant digits in the BER calculation. It is often useful to have an integration period that is much longer than this.

ISSUES IN BER TESTING

Issue: Bit Slips

An important consideration for BER testing is how to handle “bit slips.” This condition occurs when clock pulses are erroneously generated, or when some clocks are lost. This condition causes the phase of the pseudo-random sequence being received to be unmatched with the phase of the duplicate reference sequence. Subsequent comparisons will indicate an extremely bad error condition, even though the incoming data is a valid pseudo-random sequence. In order to accommodate these situations, BERTs usually have some technique for detecting very bad error conditions, which can trigger another seed attempt. This reinitialization step will resynchronize the received data with the internally generated reference data.

Issue: Data Dropouts

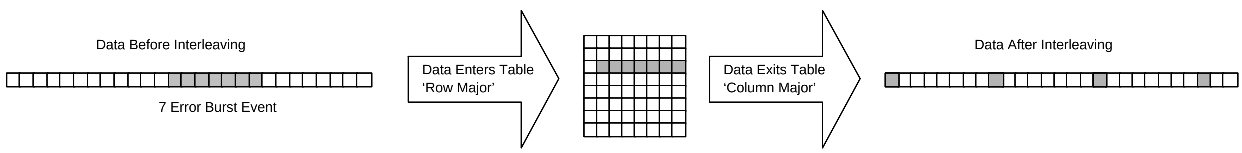
If a data dropout occurs which causes received data to become an all zeroes sequence instead of a proper pseudo-random sequence, misleading BER measurements can result. When this occurs, the error detector will encounter an enormously high error condition that is the result of comparing the incoming all-zeroes sequence with the duplicate reference pseudo-random sequence. Eventually a seed synchronization step is triggered, which loads the *n*-stages of the reference data generator circuit with all zeroes. This pathological case of the pseudo-random generator causes it to always generate zeroes, which subsequently matches the received all zeroes data sequence perfectly. In this situation, the bit error rate tester will report no errors, even though the received data is all-zeroes and is not the pseudo-random sequence that was expected.

Issue: Symbol Error Rate -vs- Bit Error Rate

When performing BER testing, it is important to consider whether bit-oriented or symbol-oriented statistics are important. Usually during channel testing, bit-oriented statistics are useful. This determines the probability of error, P_E. Many systems utilize M-ary signaling techniques, which communicate multibit symbols. In these systems, it is possible that multiple bit errors occurring close to each other will create only one symbol error. In this situation, the Symbol Error Rate (SER) is a useful metric.

Issue: De-Interleaving Requirements

It is also important to consider how the order of the bits being tested relates to the order of bits as they are transmitted. Many systems utilize interleaving techniques to uncorrelate burst errors and distribute them over many FEC data blocks. In these systems, consecutive errors in the transmission domain will be separated in the baseband domain by a number of bits equal to the depth of the interleaver. To achieve error statistics in the geometry of the transmission domain, a BERT must reverse this process.



Issue: Parallel Interfaces

Some channels now operate strictly on a symbol basis, and their electrical interfaces are symbol oriented. For instance, a 256 QAM system may have a byte interface for providing the symbols to the modulator and retrieving symbols from the demodulator. Some BERTs now accommodate these parallel interfaces, in addition to strict serial connections. Otherwise parallel-to-serial and serial-to-parallel converts will be required.

Issue: Formatted Data Testing

Many communications channels are not suited for testing with sequential streams of pseudo-random data. Many, for instance, embed formatting information in the transmission stream, which is required for proper reception and deformatting. In these cases, interfacing electronics may be required to create test sequences that have proper formatting information, and which embed pseudo-random sequences in the "payload" portions of the formatted stream. The receiver must be able to strip off the formatting information and reconstitute the pseudo-random stream in order for the BERT error detector to accomplish its task.

An alternative is to test formatted channels with non-pseudo-random sequences. Some BERT devices have RAM features so that arbitrary bit sequences can be used for testing. In this fashion, properly formatted bit sequences can be built and loaded into the BERT's RAM. Synchronizing with non-pseudo-random data is more challenging, however, since it is not as simple as loading *n*-stages of a feedback shift register circuit.

ERRORS IN A DCS

In a DCS, certain components of the processing chain are more responsible for the BER of the channel. Some components may contribute errors, some may be specifically added to eliminate errors, and still others may have the unfortunate side effect of multiplying existing errors into even more errors.

Source Encoding and Decoding

Encoding source information refers to the process by which original information is converted into an efficient digital representation for successful transmission. Some information begins as analog waveforms that are converted into digital values for transmission. This process starts by first eliminating any frequencies from the analog waveform that would not be able to be represented by the digital version of the waveform—given the sampling rate of the analog-to-digital conversion. This filtering causes loss of high frequency information. Noise is also created by quantization error and quantizer saturation error, as well as by jitter in the sampling clock and in the reconstruction clock. All these effects cause distortions in the reconstructed analog waveforms, but these problems are not evaluated using BER metrics.

Other types of source encoding and decoding operates on digital data. This includes data compression techniques, which are used to minimize the essential information in a data set. Recreating the original data from the compressed stream can either be a "lossy" process or a "lossless" process. Lossy compression systems cannot be evaluated using bit error statistics, but lossless ones can.

Once a data set's essential information is compressed down to a minimized bit stream, each individual bit becomes even more important to recreating the original information. One error in the compressed stream generally causes many errors in the uncompressed data. In some cases, bit errors in the compressed stream can make sections of the stream unrecoverable. This is one example of how a random error can be multiplied into a burst of correlated errors by the use of advanced digital communication techniques.

Baseband Communication

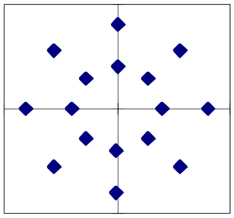
Once an information source is converted into a bit stream, it can be communicated by simple means such as clock and data signaling. The spectrum of this signal generally goes from DC to some cutoff frequency, and is therefore referred to as baseband. Communicating digital baseband signals even by simple means is prone to a number of error sources, including channel noise, clocking jitter, and intersymbol interference. Digital channels are resistant to small amounts of channel noise, but once a threshold of signal level noise is reached, errors will occur.

Errors from thermal noise will be random, but many other error sources will create correlated errors. Transient impulses will create error bursts. Errors from clock jitter will depend on the nature of the jitter. Often jitter is the result of discrete frequency interferences, and in this case, errors will also be correlated to the interference frequency. Also, many forms of failing digital components will produce errors in a systematic way.

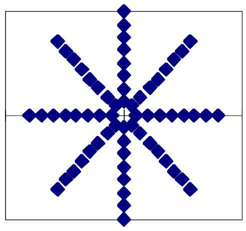
Carrier Wave Modulation and Demodulation

Modulation is the process that transforms digital baseband symbols into waveforms that are compatible with the transmission channel. Carrier wave modulation is used to propagate baseband information through band-limited channels, such as telephone lines, or through space using electromagnetic fields. Transmission is accomplished by coupling the information with a base frequency called the "carrier." The cosine waveform representing the carrier is modified in amplitude and phase by the transmitter. The receiver synchronizes with the received carrier cosine, and notices the amounts of variation in amplitude and phase. These amounts indicate the symbols from the digital alphabet that are being communicated. Constellation diagrams display the symbol assignments for different phase angles and amplitude values. For instance, here are two ways to distribute symbols, radially and rectangularly.

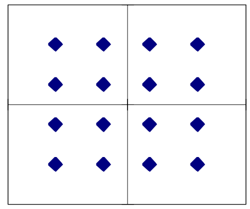
Radial-16



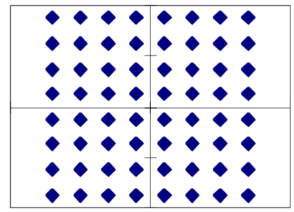
Radial-64



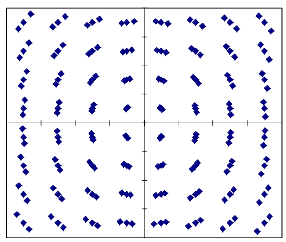
Rectangular-16



Rectangular-64



In both systems, phase noise and amplitude noise will cause digital errors attributable to the modulation step. In particular, phase noise is very interesting in non-radial schemes. As phase noise causes the constellation to rotate slightly, jittering back and forth, this phase error will cause more disturbance in outlying symbols.



Symbols assigned to these outlying regions of the constellation will have a higher probability of error than symbols that are nearer in. In this situation, errors will correlate with the symbols being transmitted, which produces a data-dependent error syndrome.

Channel Coding and Decoding

Channel coding is used to improve BER performance by modifying the transmitted bit stream such that subsequent symbol detection accuracy will be improved. Channel coding may involve selection of the analog waveforms that represent the digital symbols so that all waveforms are as unlike as possible. Another form of channel coding converts structured blocks of data into new structured blocks before transmission in an effort to better identify and/or correct errors upon detection. For instance, adding a checksum to a packet of data helps in identifying errors. Furthermore, converting data blocks using forward error correction (FEC) techniques, into new data blocks for transmission, can help the receiver identify and correct a certain number of errors. A third type of channel coding utilizes convolutional techniques to recognize the most probable received symbol, given a received waveform, recent history concerning symbol decisions, and strict rules governing which symbols can be adjacent to each other. These techniques permit controlled amounts of intersymbol interference, enabling enhanced detection. These convolutional codes are often referred to as "trellis" codes and include classic Viterbi detectors.

Errors attributable to channel coding can also be systematic. For instance, a particular (n,k) FEC code can identify and correct $(n–k)/2$ errors in each received data block. If a data block has more than $(n–k)/2$ errors, an unrecoverable error syndrome occurs, which will cause many, if not all, of the symbols in the block to be recognized incorrectly. This appears as a large burst of errors in the data stream. Convolutional codes utilize a memory of recent symbol decisions, which means that one misdetected symbol can have a ripple effect and cause future symbols to be misdetected. The amount of ripple effect is determined by the depth of the memory.

Multiplexing and Demultiplexing

Both time division and frequency division multiplexing techniques are commonly employed to allow multiple simultaneous subchannels to share the same transmission medium. In both situations, analyzing the combined stream BER may hide the fact that a particular subchannel within the stream is more responsible for the total errors in the channel than the others.

In frequency division multiplexing, errors may result from co-channel interference, adjacent channel interference, inter modulation distortions, or interference from nearby narrow band noise sources. In time division multiplexing, errors may be correlated to the position that the time slice occupies in the overall multiplexing format. These situations represent correlated errors in the channel.

RF Transmission and Reception

Radio frequency (RF) transmission commonly consists of a frequency up-conversion stage, a high power amplifier, and an antenna. Reception usually consists of an antenna, a low noise amplifier (LNA) and a down-converter stage. Individual stages within the transmitter/receiver chain contribute noise that may cause digital errors. Important sources of noise in RF communications include antenna loss, radome loss, pointing loss, polarization loss, space loss, and galactic, cosmic, star, and terrestrial noises. Atmospheric effects commonly cause channel fading and instantaneous transients that produce burst error syndromes.

Synchronization

There are many types of synchronization effects that cause correlated errors in a DCS.

Problems in clock recovery can cause phase error between clock and data signals. This jitter may be the result of frequency-dependent variables like the tracking frequency of a PLL design, or data-dependent variables like the number of signal transitions in the received bit stream.

Often times transmissions are formatted with a training pattern followed by a recognizable sync word, followed by a block of "payload" data. Problems recognizing the sync word may cause the entire block to be lost. Furthermore, clock recovery initiated by the training pattern may pick up additional phase noise as a function of time since the last training pattern. This would cause more errors at the end of these blocks.

Synchronizing some types of channel demodulators will create increased errors. For instance, a demodulator that relies on receiver synchronization with the carrier frequency only will synchronize faster than one that requires frequency and phase synchronization. Error bursts during resynchronization will be longer for the latter type.

Even with guard times, synchronization of time-division multiple access channels may interfere with one another if propagation times are not accurately accounted for. This situation will cause error bursts.

New technologies like code division multiple access (CDMA) techniques utilize frequency spreading to avoid narrow band interferences. These techniques multiply data by pseudo-random data streams upon transmission. Upon reception, the proper phase of the pseudo-random stream must be found in order to decode the original data. If synchronization is lost, resynchronizing can take a long time.

Error Control

Many forms of error control mechanisms may be found in the modern DCS. Simple techniques include adding a parity bit when transmitting RS-232 characters. More efficient error detection can be accomplished by computing cyclic redundancy codes (CRC) on data blocks. This facility is commonly included as part of any packet-oriented communications channel. Using CRC codes, a receiver can identify packets that contain errors and automatically request a retransmission of the errored packet. Of course this only works when there is a back-channel capable of informing the transmitter to resend a packet, and only if real time requirements for timeliness of data can still be met. In unidirectional, prerecorded, or real time applications, error control is either omitted or a form of forward error correction (FEC) coding is used which adds redundant information to the transmitted signal enabling error identification and correction.

When FEC is used on channels with burst errors, interleaving is often used to distribute individual errors across as many FEC data blocks as possible. This is necessary because FEC codes are best suited for correcting a few errors in any one data block. One large burst error could saturate the correction ability in a particular data block while neighboring blocks could have no errors in them at all. Interleaving effectively uncorrelates error bursts and distributes them amongst many FEC blocks by transposing a two-dimensional matrix of symbols.

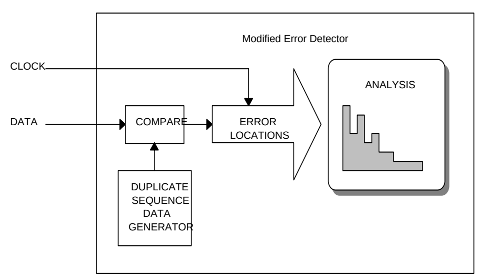
All error control mechanisms have limitations. Parity bits can only detect an odd number of bit errors in a word. Likewise, CRC codes can miss errors. FEC codes correct errors successfully until their correcting strength is saturated, then error bursts occur.

BIT ERROR LOCATION ANALYSIS

Gaussian noise can be fully described by the variance of the Gaussian distribution. Likewise, to understand digital errors that are distributed based on a Gaussian distribution, one needs to know only the BER. However, many errors are correlated, and therefore not Gaussian, and to understand these error syndromes, error location analysis is required.

Method for Locating Errors

Error location can be identified using the same basic techniques as traditional BER testing involving test transmission of pseudo-random sequences. It is required to modify the error detector so as to communicate error locations for analysis. This is the subject of a patent held by SyntheSys Research.

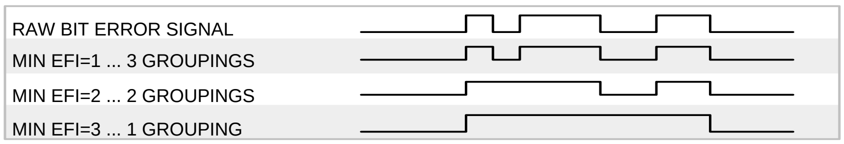


In addition to locating errors, it is important to be able to correlate errors to other external events, and so identifying locations of other input signals is also required.

Burst Error Identification

When the modified error detector identifies a cluster of errors, analysis determines whether this cluster constitutes an error burst or not. Specification of error burst criteria is application-dependent, and requires selection of two parameters:

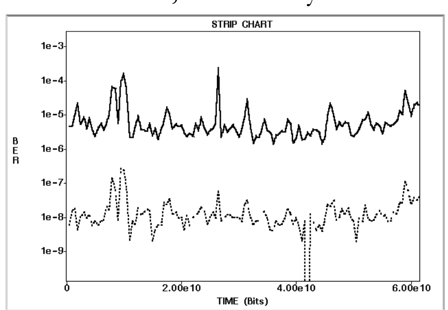
Minimum Error Free Interval — Identifies how many consecutive non-errored bits may be in a burst error. In most situations, ten errors followed by one good bit, followed by ten more errors is not considered as two separate bursts. Rather, the good bit is considered as part of an error burst of length 21; it just got lucky. Another way to explain this parameter is that it is the number of good bits required to terminate a grouping of errors for burst identification. Generally, increasing the minimum error free interval creates longer and fewer groupings to be recognized.



Minimum Burst Length — Once a grouping is collected, its length, including any intervening good bits, is compared with the minimum burst length parameter to determine if the grouping is large enough to qualify as a burst error.

Two Components of Total BER

Using error location information, bit errors can be classified in real time as to whether or not they participate in a burst event. This yields two categories, burst errors and non-burst errors, which combine to form the total quantity of errors. These separate components also lead to three error rate metrics for a given channel: Total BER, Burst BER, and Non-Burst BER. The example below shows a strip chart of total BER (top trace) and non-burst BER (bottom trace), where the non-burst category comprises only single-bit errors. Notice that there is approximately three orders of magnitude difference between the number of single-bit errors and the number of errors found in burst events, and when the total BER is worst, it is chiefly due to burst error.



It is useful to separate these components of the BER and to monitor small errors separately from large errors because they can often be attributed to different types of channel problems. Often, a DCS has different mechanisms for handling these two different types of errors.

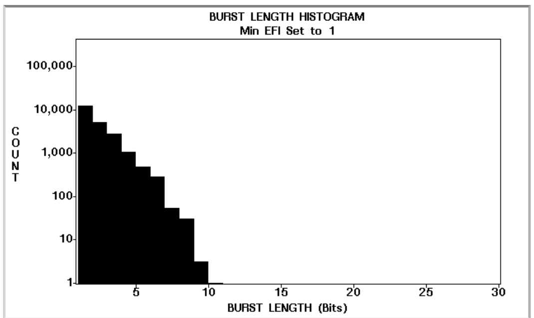
GRAPHICAL ERROR LOCATION ANALYSIS

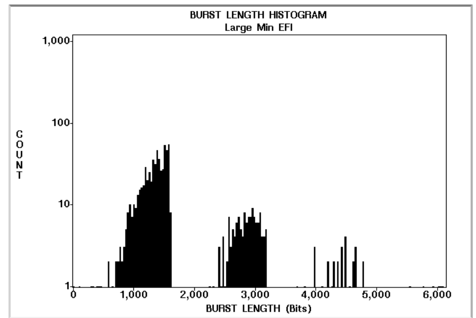
A variety of graphical results can be obtained by processing error location information. Obtaining these results in real time permits dynamic selection of channel parameters and immediate viewing of the performance differences attributable to the new selections.

Burst Length Histogram

The burst length histogram of a channel demonstrates the probability of receiving error groupings of different lengths. This information is very useful during DCS design to anticipate error densities in order to accommodate them. For instance, when designing an interleaver, the objective is to accommodate more than the average burst error length in the depth of the interleaver.

Errors in digital channels commonly occur in bursts due to instantaneous transients, interferences, synchronization problems, and error control failures. Quite often the length of the error bursts can identify the source of the error within the channel. For instance, uncorrectable FEC data blocks may cause error bursts that are approximately equal in length to the size of the data correction block.



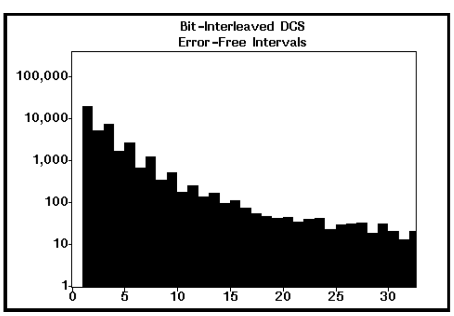


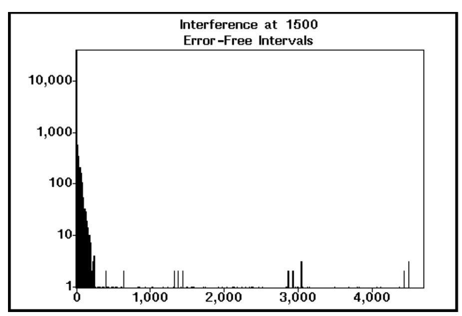
The example on the left shows a burst length histogram with the minimum error free interval specification set to one. This requires bit errors to be contiguous in order to form burst events. Notice there are generally fewer quantities of longer bursts. This channel had a BER of approximately 1.0×10^{-5}. Assuming Gaussian distribution of these errors, there ought to be a 1.0×10^{-10} chance of having two errors consecutively, 1.0×10^{-15} of three consecutive errors, and so on. At 30 MHz, that's 106,000 years before encountering four errors in a row. And yet, in only 35 seconds of testing we've encountered over 1,000 four-bit errors, and many even larger. Therefore, these errors are definitely not randomly distributed.

The example on the right shows a burst length histogram with the minimum error free interval specification set very large. This groups errors together into very large bursts. This channel uses FEC blocks that are approximately 1,500 bits in length. This histogram demonstrates that bursts often come in at 1X, 2X, and 3X the size of the error correction blocks, which is consistent with the failure modes for this error control mechanism.

Error Free Interval Histogram

The Error Free Interval histogram demonstrates the probability of having specific error free distances in the channel. This is a good way to see repetitive error syndromes, since errors occurring at a particular frequency will also produce error free distances approximately equal to the number of bits in the period of the repetition. The error free interval histogram is especially suited for discovering correlated errors that are the result of systematic interferences. Examining very small error free distances is useful to understand the density of error bursts.

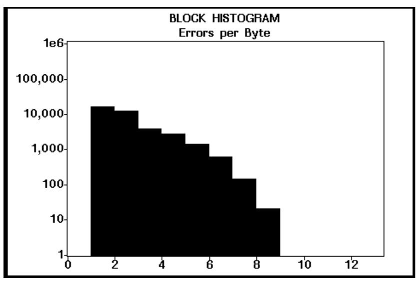


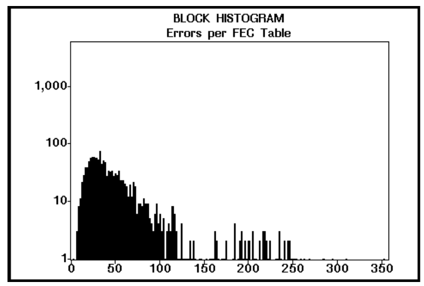


On the left is an error free interval histogram looking at the very small error free distances occurring in the channel. This channel is actually comprised of two independent subchannels that are combined in a bit-interleaving process to make a single channel. Notice there are more odd length small error free distances in this DCS. This occurs because a two-bit error in one of the original subchannels will end up being separated by a single bit in the combined channel, thus creating the odd error free distance. The diagram on the right shows an interference occurring within this DCS. Errors are trickling in on approximately 1500 bit boundaries, sometimes 3000, and sometimes 4500. These are definitely systematic error syndromes.

Block Errors Histogram

The Block Error histogram demonstrates the number of errors in user-specified blocks. Picking the block size is dependent on the application. In FEC design, for instance, it is important to know how many errors are common in an FEC block to know how strong the error corrector needs to be. The diagram on the right demonstrates that an error correction strength capable of correcting 150 bit errors would be able to correct most of the errors in this channel, leaving the rest for a different form of error control, perhaps packet retransmission.

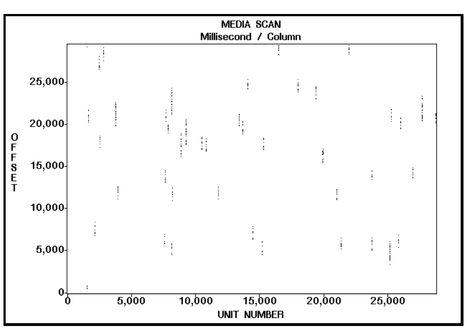


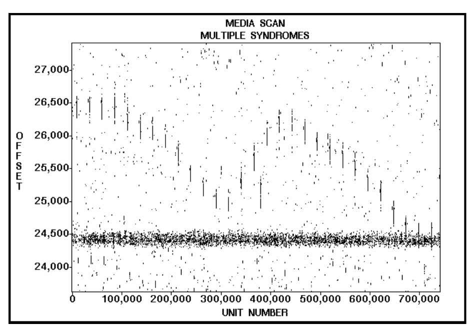


Many block sizes can be pertinent to a DCS. For instance, the diagram on the left demonstrates a block size equal to eight—the number of bits in one byte. This analysis shows that most bytes that are in error have a single bit in error, and that it is less likely that all eight bits are in error.

2D Error Mapping

In communications channels, error patterns can occur which the human eye can pick out of a two-dimensional map of the errors. This is similar to seeing the interference "herringbone" patterns displayed by television receivers when a hairdryer is turned on nearby. Because the eye is good at seeing these correlations, it is useful to pick certain blocking factors and rectangularly plot the errors on a 2D map. The blocking factor may be something significant to the channel, such as the number of bits in a formatted frame or packet, or it can be something arbitrary, like a millisecond window.



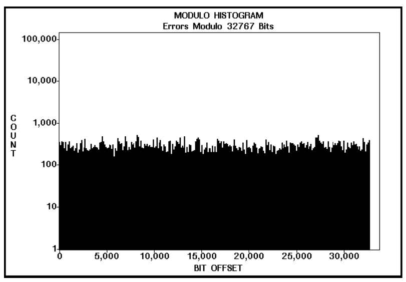


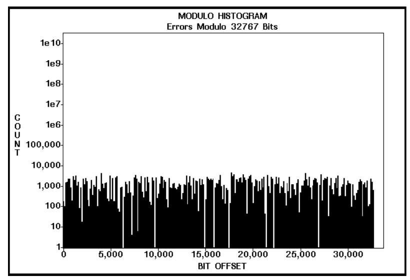
The diagram on the left is from a 30 MHz communications channel. Each column represents an arbitrary one millisecond time period. It is clear that this channel is experiencing burst errors. The diagram on the right demonstrates that multiple error syndromes may be occurring at the same time. This can obscure error analysis, but when viewed with a 2D error map, different syndromes can be identified.

Modulo Pattern Histogram

The Modulo Pattern histogram demonstrates correlations between errors and the pseudo-random sequence being tested. For a maximal length pseudo-random pattern with N shift register stages, the sequence will repeat in 2^N-1 bits. This means there are 2^N-1 positions along the pseudo-random sequence, and if any are more prone to error than others, a data-dependent error syndrome exists. For example, if a phase lock loop requires clock transitions to be able to recover a stable clock, then portions of the sequence where there are a lot of zeroes may be more prone to jitter and therefore cause increased errors.

The following two modulo pattern period histograms are results from two different DCS channels. The PN-15 pseudo-random sequence was being used, so the sequence repeated every 32,767 bits.



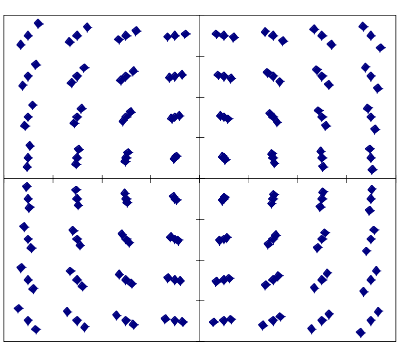


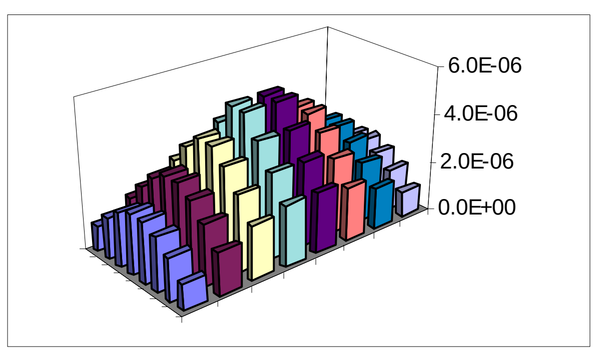
Notice that the channel on the left has roughly the same numbers of errors at all positions along the pseudo-random sequence. The channel on the right, however, has a definite correlation between errors and positions along the pseudo-random sequence. Notice that there are certain locations within the sequence that have no errors whatsoever.

Symbol BER

In a symbol-oriented DCS, pseudo-random data is blocked into multibit symbols as it is transmitted. By knowing the location of symbol values along the pseudo-random sequence being tested, it is possible to correlate errors along the pseudo-random back to the symbol values that are responsible for the errors. Using this mechanism it is possible to identify distinct bit error rates for each symbol in the alphabet.

The analysis on the right represents the errors caused by phase noise in a rectangular modulation scheme, as shown on the left. Notice error rates are better in the center and worse along the outside of the constellation.

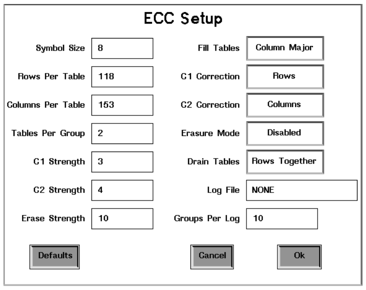


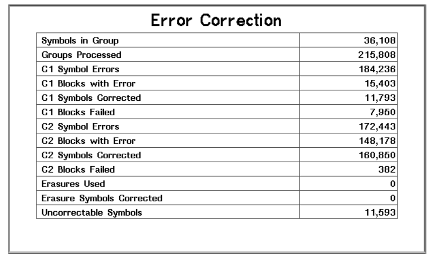


By simply doing symbol-oriented BER testing, this technique can identify problems in the analog modulation domain.

Forward Error Correction Emulation

Knowing the location of errors within a data stream permits FEC Emulation. Using this technique, errors from a raw channel can be filtered through a hypothetical error correction system, and corrected BER results can be demonstrated in real time. This requires selection of block-oriented FEC parameters including the table geometry, interleaving architecture, and correction strengths.





Error Correction

Symbols in Group	36,108
Groups Processed	215,808
C1 Symbol Errors	184,236
C1 Blocks with Error	15,403
C1 Symbols Corrected	11,793
C1 Blocks Failed	7,950
C2 Symbol Errors	172,443
C2 Blocks with Error	148,178
C2 Symbols Corrected	160,850
C2 Blocks Failed	382
Erasures Used	0
Erasure Symbols Corrected	0
Uncorrectable Symbols	11,593

This technique is useful for analyzing the before-and-after performances for different FEC proposals in order to design the best overhead -vs- performance tradeoff.

CONCLUSIONS

Measuring performance in a DCS can be done by transmitting pseudo-random data sequences through the channel and recognizing errors at the receiving end. This Bit Error Rate testing technique can be significantly improved by also recognizing error *location* information in an error detector. Using error location information, many useful statistics including Burst Length histograms, Modulo Pattern histograms, Block Errors histograms, Error Free Interval histograms, and 2D Error maps can be achieved. Advanced error location processing can be performed to calculate separate BER measurements for each symbol value being communicated, and to emulate the performance of candidate FEC architectures without developing any hardware prototypes. These newly developed tools permit DCS developers greater visibility into channel performance, and help them to solve problems more quickly.

REFERENCES

[1] I. Korn, "Performance of Digital Communications Schemes with Multidimensional Signals in Gaussian Noise," IEEE Trans. Commun., vol. 44, no. 9, September 1996.

[2] B. Sklar, "Digital Communications Fundamentals and Applications," P T R, Prentice Hall Publishing Company, Englewood Cliffs, New Jersey

[3] J. Proakis, "Digital Communications Third Edition," McGraw-Hill Publishing Company, New York, New York.

[4] D.J. Kennedy and M.B. Nakhla, "Burst Error Characterization of FEC Coded Digital Channels," Intelsat Publication, Washington D.C.,

[5] J. Waschura, "BitAlyzer400 Bit Error Analyzer User Guide," SyntheSys Research Inc. Publication BA400-701, Menlo Park, California