01 Jul 1994

Simulation of Communication Systems

William H. Tranter
Missouri University of Science and Technology

Kurt Louis Kosbar
Missouri University of Science and Technology, kosbar@mst.edu

Recommended Citation

W. H. Tranter and K. L. Kosbar, "Simulation of Communication Systems," *IEEE Communications Magazine*, vol. 32, no. 7, pp. 26-35, Institute of Electrical and Electronics Engineers (IEEE), Jul 1994.
The definitive version is available at https://doi.org/10.1109/35.295942

Simulation of Communication Systems

When both a complex system and a complex channel model are encountered, the result is typically a design or analysis problem that cannot be solved using traditional (pencil and paper) mathematical analysis. Computer-aided techniques, which usually involve some level of numerical simulation, can be a very valuable tool in these situations.

William H. Tranter and Kurt L. Kosbar

Over the past decade considerable attention has been paid to the development of computer-aided design and analysis tools that can be applied to communication systems. There are several reasons for this. Today's communication systems are much more complex than those of several decades ago. In addition, many systems operate in environments where the channel is not adequately described by a simple additive Gaussian noise model. The effects of severe bandlimiting, adjacent-channel interference, multipath, nonlinearities, and a host of other degrading effects must now be considered. When both a complex system and a complex channel model are encountered, the result is typically a design or analysis problem that cannot be solved using traditional (pencil and paper) mathematical analysis. Computer-aided techniques, which usually involve some level of numerical simulation, can be a very valuable tool in these situations. The purpose of this article is to provide a tutorial review of some of the basic techniques of communication system simulation.

Another reason for the current interest in simulation and computer-aided techniques is the widespread availability of powerful computers. These tools are currently within reach of most communication engineers and it is now possible to perform system-level simulations of complex systems at one's desk. The graphics capabilities of modern personal computers and workstations, together with laser printers, allows output to be generated in a readily usable form. These capabilities have been available for a relatively short time.

Both traditional mathematical analysis and computer simulation are based on a system model, which is typically a block diagram that describes the interconnection of the various subsystems comprising the overall system. Each functional block or subsystem is described by a signal processing operation that defines the subsystem input-output relationship. The accuracy of either the mathematical analysis or the computer simulation is dependent upon the accuracy of the system model. Thus, each and every approximation made in developing a system model must be well understood. Computer simulation has the same goal as conventional mathematical analysis — to determine the operating characteristics and performance of a communication system. Link-level simulations typically focus on the performance measures of a communication link. Typical performance measures include the time required to initialize a link, the length of time a link can be sustained, the signal-to-noise ratio (SNR) of the recovered message in analog systems, and the symbol error rate for digital systems. Despite these similar goals, simulation often differs from mathematical analysis in a fundamental way. Simulation typically focuses on performance *estimation* while mathematical analysis nearly always involves performance *calculation*. The result of a traditional mathematical analysis is a number, while the result of a simulation is typically a random variable. This is an important distinction.

There are basically two different classes of problems that can be addressed using simulation: the transient characteristics and the steady-state characteristics of a system. The time-to-lock of a PLL used as a bit synchronizer is a typical transient characteristic. Transient characteristics are usually determined using a simulation of the specific sub-system of interest rather than using a simulation of the system as a whole. When one uses simulation to determine the performance characteristics of a system, then the entire system, including the environment in which the system operates, must be included in the simulation. Performance measures are typically steady-state characteristics. Examples are the bit error rate, mean-square error, and signal-to-noise ratios. Link-level simulation allows these problems to be addressed for arbitrarily complex systems.

Simulation should never be viewed as a substitute for mathematical analysis. Some level of analysis is necessary if one is to establish that the simulation is working correctly and that the simulation results are reasonable. This is the area of validation, which will be addressed further. Simulation, when properly used, goes hand-in-hand with traditional analysis methods. Simulation results often allow us to

WILLIAM H. TRANTER is the Schlumberger Professor of Electrical Engineering at the University of Missouri-Rolla.

KURT L. KOSBAR is an assistant professor of electrical Engineering at the University of Missouri-Rolla.

identify the most important parameters in a system and also help identify those system parameters that can be neglected. In other words, simulation results often guide analysis, since a properly developed simulation provides insights into system behavior.

While simulation is a powerful tool for both design and analysis, new problems are created when one turns to simulation. Since the continuous-time waveforms present in the system must be represented by discrete-time samples in the simulation, the waveforms must be sampled so that aliasing errors are reduced to acceptable levels. Engineering judgments are necessary for even this simple problem. The reduction of aliasing errors to negligible levels requires high sampling frequencies. High sampling frequencies in turn result in large simulation run times, which is clearly not desirable. Thus, an obvious tradeoff exists. Another problem is that the analog filters that may be present in the actual system under study must be represented by digital equivalents in the simulation. These digital equivalents always involve approximations whose nature should be understood if the simulation user is to have complete confidence in the simulation results.

A comprehensive survey of the techniques used for the simulation of communication systems would fill a rather large book [1]. In this section we will briefly consider the basic techniques used to represent signals, generate signals, and model linear systems, nonlinear systems, and time-varying systems within a simulation. We then consider the important problem of using a simulation to estimate the performance of a communication system.

Signal and System Modeling

System-level simulations can be based on time-domain techniques, frequency-domain techniques, or on a combination of these techniques. In this section we focus on the problems associated with representing time-domain signals, and modeling systems, in a digital simulation of a communication system.

Signals and Complex Envelopes

Both lowpass signals and bandpass signals are usually present in a communication system. Lowpass signals are typically information bearing signals prior to modulation and bandpass signals typically represent modulated carriers at various points in the system, such as transmitter outputs and receiver inputs. Both lowpass and bandpass signals must be represented by discrete-time sequences within the simulation. The analog signals actually present in many parts of a communications system must obviously be sampled to form the discrete-time sequences processed by the simulation. These sampled sequences must accurately specify the corresponding analog waveform if an accurate simulation is to result.

In order for sample sequences to accurately specify the analog waveforms from which the samples are formed, the sampling frequency f_s must exceed twice the highest frequency in the waveform being sampled [2]. There are a number of factors that influence the choice of the sampling frequency. Among these factors are aliasing errors, frequency warping in digital filters, and the presence of nonlinearities. Computational constraints also affect the choice of the sampling frequency. Since the simulation program must process each sample, an excessive number of samples used to represent a given waveform leads to excessive computer time requirements. We are therefore rewarded by selecting the lowest possible sampling frequency that still results in an accurate simulation. An understanding of the tradeoff between simulation accuracy and the simulation sampling frequency is important. This is usually accomplished after a simulation is developed by varying the sampling frequency and observing the changes that result in the simulation outputs.

The desire to minimize the simulation sampling frequency points us toward using signals in the simulation having lowpass-type spectra. Lowpass signals present no problem, they are sampled directly using an appropriate sampling frequency. Bandpass signals can also be directly sampled but are usually represented by equivalent lowpass signals in order to reduce the number of samples necessary to represent the signal. The complex envelope representation allows us to accomplish this.

A general modulated signal, having carrier frequency f_c is usually written in the form

$$x(t) = R(t)\cos[2\pi f_c t + \phi(t)] \quad (1)$$

where $R(t)$ represents the real envelope of $x(t)$ and $\phi(t)$ represents the phase deviation. Equation (1) can be placed in the form

$$x(t) = \mathrm{Re}\{R(t)e^{j\phi(t)}e^{j2\pi f_c t}\} \quad (2)$$

or

$$x(t) = \mathrm{Re}\{\tilde{x}(t)e^{j2\pi f_c t}\} \quad (3)$$

where the quantity $\tilde{x}(t)$ is called the complex envelope of the real signal $x(t)$. Clearly

$$\tilde{x}(t) = R(t)e^{j\phi(t)} \quad (4)$$

is a complex function of time that is independent of the carrier frequency f_c. It is important to note that the complex envelope involves signals that are usually slowly varying with respect to the carrier frequency. Since the bandwidth of a bandpass signal is usually small compared to f_c, it takes a much lower sampling frequency to represent the complex envelope, $\tilde{x}(t)$, than to represent the real-time signal $x(t)$. The result is a smaller number of samples for a given time segment of $x(t)$. The complex envelope is usually expressed in rectangular form

$$\tilde{x}(t) = x_d(t) + j\,x_q(t) \quad (5)$$

where $x_d(t)$ is the direct (or real) component of $\tilde{x}(t)$ and $x_q(t)$ is the quadrature (or imaginary) component of $\tilde{x}(t)$ [3].

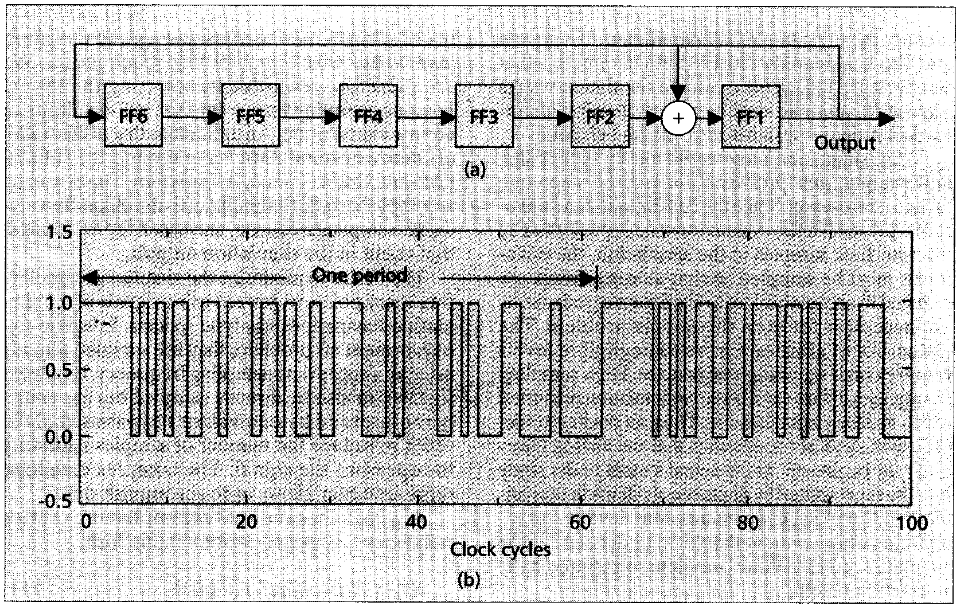
Assuming that the carrier frequency is *known*, the complex envelope contains *all* of the information contained in the original signal $x(t)$. As shown by Eq. 3, $x(t)$ can be reconstructed from $\tilde{x}(t)$ by multiplying $\tilde{x}(t)$ by $e^{j2\pi f_c t}$ and taking the real part.

Signal Generation

Both deterministic and random signals exist in almost all communication systems. Models must be developed for each of these signal types that can be implemented in a digital computer simulation. Deterministic signals are usually generated using the defining equation for the signal. Equation (1), with $R(t)$ and $\phi(t)$ properly specified to represent the signal of interest, is an example. The only other concern is the choice

Both lowpass signals and bandpass signals must be represented by discrete-time sequences within the simulation.

> **A** *PN sequence generator is usually envisioned as a linear binary shift register. The characteristics of the generator are established by the feedback taps.*



■ **Figure 1.** *PN sequence generation: a)implementator of a PN sequence generator for* m = *6; b) resulting waveform for a given seed.*

of sampling frequency, as discussed previously.

Random signals are usually generated using either a linear congruential algorithm or a PN sequence algorithm. Although the mathematical descriptions of these two algorithms are somewhat different, they are essentially equivalent. Since a digital computer is a finite-state machine, it is not possible to generate a truly random signal on a computer and all computer-generated sequences are periodic. We are content to generate a pseudo-random sequence, which is in reality a periodic deterministic signal with a long period. Within a period the pseudo-random sequence approximates many of the properties of a random signal. We are therefore able to generate "noise-like" waveforms for use in a simulation to represent both random signals and noise, thus the term pseudo-noise (PN) sequences.

A linear congruential algorithm is defined by the expression

$$x[n+1] = (a\,x[n]+c) \bmod m \qquad (6)$$

where m is the modulus, a is the multiplier, and c is the increment. In order to improve the speed at which samples are generated, we usually set $c = 0$. The initial value of the sequence, $x[0]$, is known as the seed number of the process. Once the seed is specified, the remaining values of the sequence are specified through Eq. 6 The problem is to determine the parameters a, c, and m so that the generator defined by Eq. 6 has a sufficiently long period for the application of interest. Although the theory for accomplishing this task is well understood [4, 5] the question of what really makes a good random sequence generator, and the determination of efficient algorithms for sequence generation, still constitute an active area of research [6]. Although all simulation packages available today — suitable for communication system simulation — contain random sequence generators, the user should ensure that the operation of these generators is well understood and appropriate to the problem being investigated.

A PN sequence generator is usually envisioned as a linear binary shift register as shown in Fig. 1a. The characteristics of the generator are established by the feedback taps. The taps are defined by a polynomial and the generator achieves a maximum period of $2^m - 1$ if the polynomial is primitive [5]. For the generator shown in Figure 1a, the feedback connections are defined by

$$g(X) = 1 + X + X^6 \qquad (7)$$

which indicates feedback to the first and the sixth stages of the shift register. The register is initially placed in some state, equivalent to a seed number, and it then cycles through all possible states with period 63. This is the maximum period of $2^m - 1$ possible since the polynomial in Eq.7 is primitive and $m = 6$. The corresponding waveform is shown in Fig. 1b.

Models for Linear Systems

A model for a linear system, suitable for implementation on a digital computer is usually determined from the transfer function of the system $H(f)$, or the unit impulse response $h(t)$. If the transfer function $H(f)$, is for a lowpass type system, a computer model is easily determined directly from $H(f)$ using one of the standard digital filter synthesis techniques that map a transfer function into an equivalent digital filter. Perhaps the most popular synthesis techniques are those that yield impulse-invariant, step-invariant, and bi-linear z-transform filters [2]. All of these synthesis techniques involve approximations and it is important that the approximations be understood if the simulation user is to have confidence in the simulation result.

If, however, the starting point for a filter design is not a transfer function but an amplitude response mask, one can usually develop a linear-phase filter satisfying the requirements of the amplitude response mask. Frequency sampling filters, or finite-duration impulse response (FIR) filters based on the Parks-McClellan synthesis technique, are often useful [2]. If one is to simulate a filter with an arbitrary amplitude and phase response, it is often necessary to take frequency samples of both the desired amplitude and phase response. These samples can then be inverse transformed using the FFT to obtain the unit-pulse response $h[n]$. The input sequence can then be convolved with $h[n]$ to form the filter output. As an alternative, block FFT processing can be used. In block FFT processing the input sequence, $x[n]$, is divided into blocks of appropriate size. These blocks are then Fourier transformed using the FFT, multiplied by the filter transfer function (samples of the amplitude and phase response), and then inverse transformed to obtain the output samples. The overlap-save method [2] is typically used for these applications since $h[n]$ is a short sequence compared to $x[n]$.

We saw previously that complex envelope signal representations are generally used for bandpass signals. If the system is a bandpass system, the unit-impulse response of the system will be a bandpass signal. As such, the unit-impulse response is usually represented by the complex envelope model of the bandpass system, defined by

$$\tilde{h}(t) = h_d(t) + j\, h_q(t) \tag{8}$$

The complex envelope of the system output $\tilde{y}(t)$, is the convolution of the complex envelope of the input, represented by Eq. 5, and as given by Eq. 8. This yields

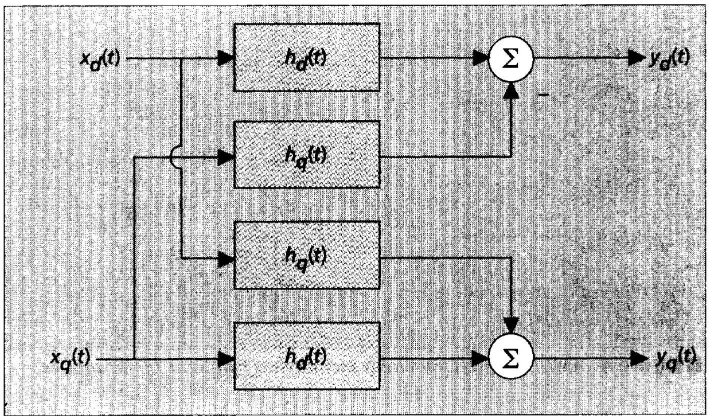
$$\tilde{y}(t) = [x_d(t) + jx_q(t)] * [h_d(t) + jh_q(t)] \tag{9a}$$

where '*' denotes convolution. The preceding expression can be written

$$\tilde{y}(t) = [x_d(t)*h_d(t) - x_q(t) * h_q(t)] \\ + j\, [x_d(t) * h_q(t) + x_q(t) * h_d(t)] \tag{9b}$$

This yields the structure shown in Fig. 2. Since the functions $h_d(t)$ and $h_q(t)$ represent lowpass signals, computer models for these signals can be realized using the same techniques described in the preceding paragraph. Two filters will be necessary, one for $h_d(t)$ and one for $h_q(t)$.

Many of the linear systems used in a communication system involve a filtering operation. Filters, of course, have memory so that past input or output samples are used in forming the current system output. Because of this structure, filtering is computationally expensive compared to many of the other signal processing operations involved in simulation. Efficient filtering routines are therefore essential elements in any simulation program.



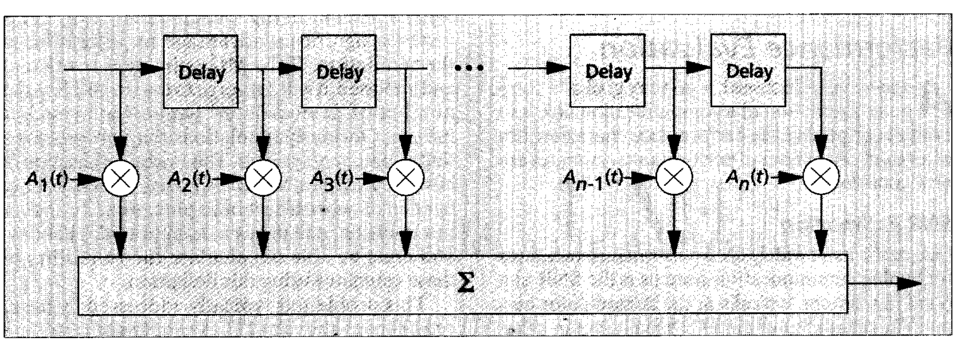
■ **Figure 2.** *Complex envelope representation of bandpass linear system.*

Models for Nonlinear and Time-Varying Systems

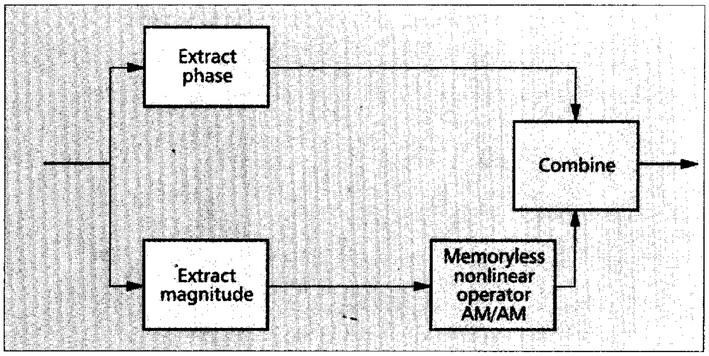
Nonlinear and time-varying systems present special difficulties when bandpass models for these systems are needed. While complex envelope models exist for linear, time-varying systems there is no guarantee that a complex envelope model exists for systems that are both nonlinear and time-varying. One must rely on approximation methods to model these devices.

Little can be said about the most general class of time-varying, nonlinear systems. The only method that ensures that these systems can be accurately modeled is to translate the complex envelope back to a bandpass signal and pass it though an appropriate device model. To develop more computationally efficient models, one must make assumptions about the device. In some cases one has a linear, but time-varying element. The model shown in Fig. 3 can then be used to represent the system. This is essentially a transversal filter with time-varying coefficients.

There are also a variety of models for nonlinear but time invariant systems. A well-known example is the Volterra Series expansion [7]. Unfortunately this expansion is computationally expensive and therefore rarely used. There is a special class of nonlinear devices that have very short, or no memory. In a true memoryless device, such as a square-law device, the output is only a function of the current input. If a simple sinusoid is placed into these devices, the output will have terms only at the harmonics of the input frequency and if a bandpass filter follows the memoryless nonlinearity, all but the first harmonic term can be removed. Thus a sinusoidal input produces a sinusoidal output, where the amplitude of the output may be a nonlinear function of the ampli-



■ **Figure 3.** *Time-varying linear system model.*



■ **Figure 4.** *AM/AM model for strictly memoryless bandpass nonlinearities.*

tude of the input. This type of device lends itself well to the complex envelope representation. As shown in Fig. 4, one merely needs to decompose the complex envelope into a magnitude and phase component, pass the magnitude through a non-linear device, and recombine it with the unaltered phase term.

Another class of interesting systems have "short" memory, i.e., the time constant of the nonlinearity is long with respect to the carrier frequency but short with respect to the message waveform. These systems can be called complex envelope memoryless systems, or envelope nonlinearities [8]. This is because the complex envelope of the output can be approximated by a memoryless, but nonlinear, function of the complex envelope of the input. Saleh [9] showed that the traveling-wave tube microwave amplifier fits this description, and the complex envelope representation of this device is shown in Fig. 5. As with the truly memoryless nonlinearity, the complex envelope of the input is decomposed into its amplitude and phase. The amplitude is both passed through a nonlinear device and used to alter the phase of the signal. If the input to the (assumed memoryless) nonlinearity is

$$x(t) = A(t)\cos[2\pi f_c t + \phi(t)] \tag{10a}$$

the output is represented by

$$y(t) = f[A(t)]\cos\{2\pi f_c t + g[A(t)] + \phi(t)\} \tag{10b}$$

The function $f[A(t)]$ is known as the AM-to-AM conversion characteristic and $g[A(t)]$ is known as the AM-to-PM conversion characteristic. For a constant envelope $x(t)$, $A(t)$ is a constant and thus $f[A(t)]$ and $g[A(t)]$ are constants. This explains the interest in constant envelope modulation techniques.

Performance Evaluation

As previously discussed, a primary goal of a computer simulation of a communication link is to evaluate or predict the performance characteristics of a system. A number of performance estimates are now considered.

SNR Estimation

One of the most widely used performance measures for analog communication systems is the SNR at a point in a system, typically at the demodulator output. The calculation of the SNR usually requires that the waveform of interest (the test waveform) be compared to a "desired" or "ideal" waveform at that point. This desired waveform is often chosen to be an amplitude-scaled and time-delayed version of the information-bearing waveform since amplitude scaling and time delay do not contribute to waveform distortion. The test waveform is then compared to the desired waveform and that portion of the test waveform that is orthogonal to the desired waveform is defined as noise. For this case the SNR estimate becomes

$$\tilde{SNR} = \frac{\rho^2}{1-\rho^2} \tag{11}$$

where ρ is the correlation coefficient between the test and desired waveforms [1].

Simulation is used to establish the test waveform for the system under study. As a simple example, if the complex envelope of the test waveform is

$$\tilde{y}(t) = Ae^{j\theta}\tilde{x}(t-\tau) + n(t) \tag{12}$$

the SNR is A^2P_x/P_n where P_x and P_n are the signal and noise powers, respectively. In most applications, the values of A, θ, τ, P_x, and P_n must be estimated before the SNR can be determined. Simulation can assist in this undertaking.

Symbol Error Rate Estimation and Monte Carlo Simulation

In digital communication systems the probability of demodulation error P_e, is typically the prime performance measure. For simplicity we will only consider binary communication systems and refer to P_e as the bit-error rate, or BER. The techniques discussed here can typically be extended to include M-ary communication systems.

The Monte Carlo (MC) method is a widely known technique for estimating the BER of a communication system [1, 10]. This method is based on the relative frequency definition of probability. A simulation is first developed that closely replicates the behavior of the system under study. The simulation will include pseudorandom data and noise sources, along with models of the devices that process the waveforms present in the system. A number of symbols are then processed by the simulation, and the experimental BER is estimated as the number of errors divided by the total number of symbols processed by the simulation. In most systems, this sample BER will be a consistent and unbiased estimate of the true BER. MC simulation is an intuitively pleasing approach that can be applied to virtually any system. It also has the side benefit of generating signals that very closely replicate the signals present in the system under study. This can be a significant advantage for validation of the simulation. If the MC estimate is consistent and unbiased, it will converge to the true BER as the number of demodulated symbols approaches infinity. Obviously simulations can only process a finite number of symbols. This raises the question of how accurate is a MC BER estimate after a finite number of symbols have been processed? To answer this question, one needs a definition of reliability, and must be able to calculate the reliability of these estimates using this definition.

This problem is typically addressed by using confidence intervals [11]. A simulation result is a sample estimate of the BER, $\hat{P}_e$, and we wish to know the true BER, P_e. To apply confidence intervals, one must be able to map $\hat{P}_e$ to an interval of the real line, $[P_L, P_H]$. This interval is a 100α percent confidence interval if

$$\Pr[P_L < P_e < P_H] > \alpha \text{ for all } P_e.$$

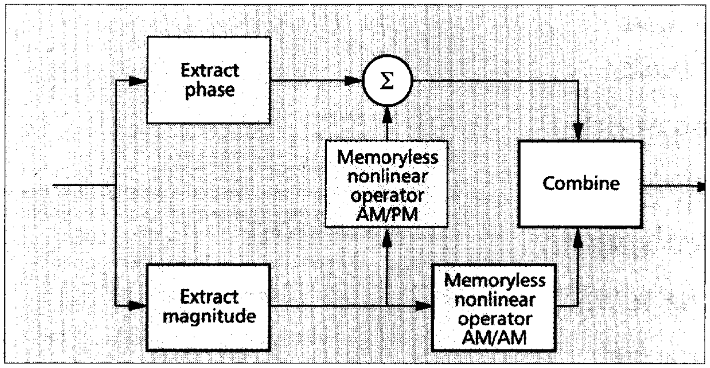
This mapping is in general very difficult to obtain and is not unique. Fortunately, for MC simulations with independent and identically distributed errors, there are well-known methods for finding the confidence interval. The 99 percent confidence interval for a simulation that has a $\hat{P}_e$ of 10^{-6} is shown in Fig 6. The important measure for this type of simulation is not the number of bits processed by the simulation, but the number of errors observed. A rule of thumb is that after one error has been observed, the 99 percent confidence interval covers approximately three orders of magnitude. One can also state, with 99 percent confidence, that after 10 errors, the estimated BER is within a factor of 2 of P_e, and after 100 errors the estimated BER is within a factor of 1.3 of P_e. To emphasize the drawback of MC simulation, the horizontal axis of Fig. 6 has been labeled in years of computer execution time, assuming the simulation will process one symbol per second of CPU time. For complex systems having low error rates, the Monte Carlo approach may require a considerable investment of processing time if accurate BER estimates are required.

The shortcomings of the Monte Carlo approach have been recognized for some time, and considerable research has been performed to find faster methods of BER estimation. These approaches are typically called variance-reduction techniques. Although a wide variety of techniques have been investigated, all share a common theme; by making additional assumptions about the system architecture and signal sources, one can reduce the number of symbols required to generate an estimate of a particular accuracy. Generally, the greater the number of assumptions used in the simulation, the greater the reduction in simulation execution time. While there is virtually no reward in this area for those who do not understand system behavior, there are tremendous rewards for those who can skillfully apply these techniques. There are also tremendous dangers for those who misapply the techniques.

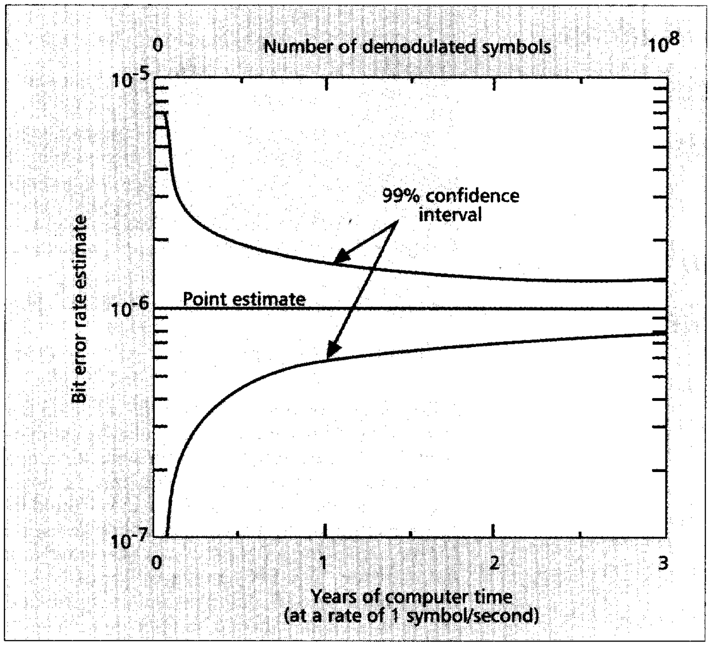
Semi-Analytic Analysis

The semi-analytic (SA) approach places substantial demands on the analyst and system architecture, but the reward is an incredibly fast simulation [1, 10]. This approach can be described by first reviewing the simple communication system shown in Fig 7. This is obviously an analytically tractable system. The decision metric will be a Gaussian random variable, with a known mean and variance. One can calculate the probability of demodulation error, and there is no need to perform a simulation.

A more interesting, and less tractable, system is shown in Fig. 8. The transmitter now has a nonlinear power amplifier. The channel adds white Gaussian noise to the signal and passes the result through a linear filter that introduces intersymbol interference (ISI). The nonlinear amplifier and ISI cause the decision metric to be decidedly non-Gaussian, making the BER difficult to calculate. However, it is not difficult to show that the decision metric is *conditionally* Gaussian. If one specifies the transmitted data pattern, the decision metric will be Gaussian, with a mean that is a function only of the data pattern and a variance that is only a function of the noise level. The BER calculation can now be decomposed into three parts: determining the variance of the decision metric, determining the conditional mean of the decision metric, and calculating the BER by using the total probability theorem.



■ Figure 5. *AM/AM, AM/PM model for complex envelope memoryless bandpass nonlinearities.*



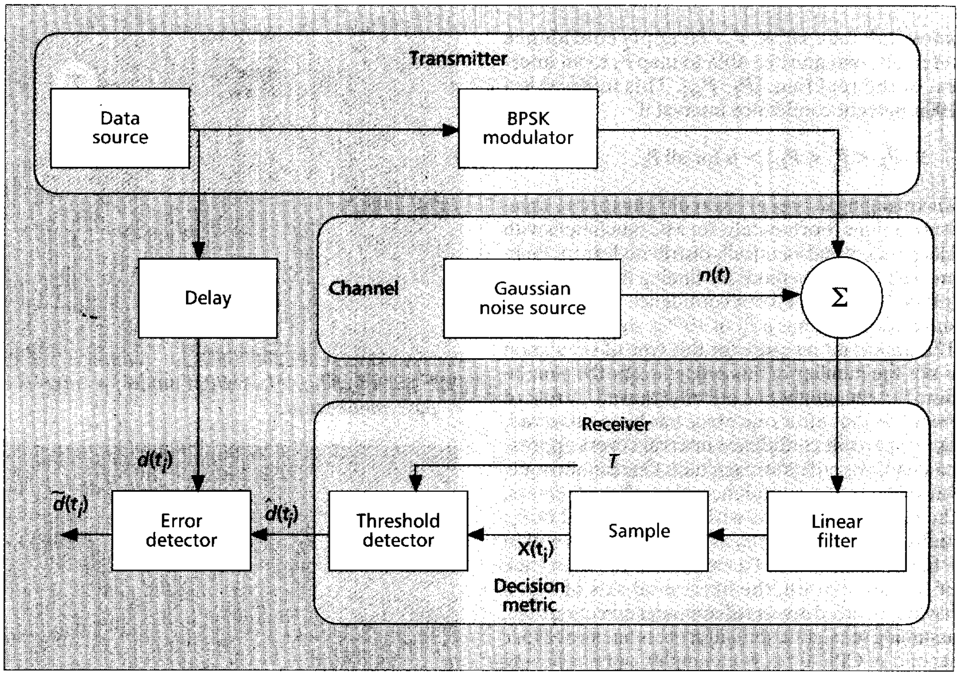
■ Figure 6. *Point and interval BER estimates.*

Assume the bandpass filter has a impulse response (or memory) that is n data symbols long. By total probability, the BER of this system is

$$P_e = \frac{1}{2^n}\sum_{i=1}^{2^n} Q\left(\frac{\left|E[X_i]-T\right|}{\sigma_X}\right) \qquad (12)$$

where each value of i corresponds to one of the 2^n possible data patterns, $E[X_i]$ is the mean of the decision metric of the i^{th} data pattern, σ_x is the variance of the decision metric, T is the threshold value, and $Q[x]$ is the familiar integral

U*nlike most other simulation techniques, the semi-analytic approach calculates the BER of the system, as opposed to estimating the BER.*



■ **Figure 7.** *Analytically tractable communication system model.*

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp\left(\frac{-\alpha^2}{2}\right) d\alpha . \qquad (13)$$

It is typically possible to analytically calculate the value of σ_X. This parameter may also be estimated by disabling the transmitter in the simulation, and measuring the variance of X_i when only the noise source is active. The mean of X_i can be found by reconnecting the transmitter, disabling the noise source, and using a PN sequence generator as a data source. The PN generator should cycle through all 2^n possible data patterns of length n, and the value of $E[X_i]$ should be recorded for each pattern. The BER for the system can then be calculated by inserting this data into Eq. 12. The SA approach can be used whenever one can calculate the BER of the system given the transmitted data pattern. It is most frequently used when the noise is additive and Gaussian, and the system is linear from the point of noise injection to the point where decisions are made.

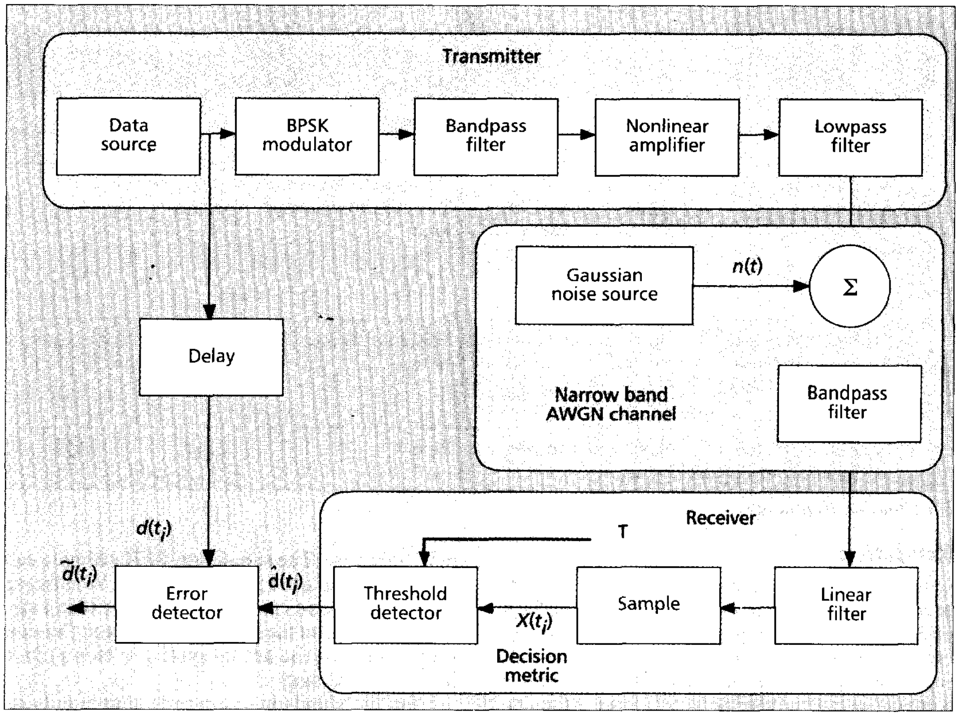
Unlike most other simulation techniques, the semi-analytic approach *calculates* the BER of the system, as opposed to *estimating* the BER. It makes very efficient use of the computer resources, and once one has performed the simulation and stored the mean and variance data, they can easily calculate the BER for any SNR. The BER can therefore be determined for a range of system noise levels with a single simulation. Given all these advantages, one expects to find a significant disadvantage. The disadvantage is that one must be able to calculate the error rate of the system conditioned on the transmitted data pattern. Notice that this is typically difficult or impossible when the noise is non-Gaussian, the noise and data are correlated, the noise is not additive, the noise is non-stationary, or when there are nonlinearities after the insertion of the noise. While there is a class of systems where semi-analytic simulations are useful, there is still a need for efficient simulation techniques that place fewer demands on the system architecture.

Importance Sampling

One technique that has received considerable attention in the literature is the modified Monte Carlo, or importance sampling, (IS) technique [1, 10]. When using importance sampling, the statistics of the noise sources in the system are biased in some manner so that errors (i.e., the important events) occur with greater probability, thereby reducing the required execution time. An MC simulation is run using the biased noise source. It is possible to unbias the BER estimate of this modified simulation by applying

$$\hat{P}_{e,IS} = \frac{1}{N} \sum_{i=1}^{N} \frac{f_n(n_i)}{f_{\tilde{n}}(n_i)} I(n_i) \qquad (14)$$

Where f_n is the pdf of the original noise source, $f_{\tilde{n}}$ is the pdf of the biased noise source, n_i is a particular noise vector and $I()$ is an indicator function that is one when an error occurs and zero when the correct symbol is demodulated. This leads to the hope that after a fixed number of demodulated symbols, the IS BER estimate will be more accurate than a conventional MC BER estimate. One can show that there is virtually no limit on how much one can gain, or lose, by using IS. If an analyst is sufficiently clever to select a good IS biasing scheme for a given system, an accurate estimate of the BER can be obtained with very short computer runs. If a poor biasing scheme is selected the BER estimate may even converge at a slower rate than the MC estimate. Many different biasing methods have been suggested in the literature and, before using IS, one should ascertain if a particular biasing scheme will produce an improvement, or a degradation, over a conventional MC simulation.



■ **Figure 8.** *Analytically tedious communication system model.*

Tail Extrapolation

The BER estimation problem is essentially a numerical integration problem. The BER of a given system is the area under the tail of an unknown probability density function (pdf). One can assume that the pdf belongs to a particular class and then perform a curve fit to the observed data. This should identify a highly likely pdf, from which one can generate a BER estimate. This is the concept behind tail extrapolation. In these simulations, one sets multiple thresholds as shown in Fig. 9. A normal MC simulation is executed, and the number of times the decision metric exceeds each threshold is recorded. A broad class of pdfs is then identified. One class that is often useful is the general exponential class

$$f_{\upsilon,\sigma,m}(x) = \frac{\upsilon}{\sqrt{8}\sigma\Gamma\left(\frac{1}{\upsilon}\right)}\exp\left(-\left|\frac{x-m}{\sqrt{2}\sigma}\right|^{\upsilon}\right) \qquad (15)$$

The parameters available in this class (ν, σ and μ) are then adjusted to find the pdf that best fits the available data. The BER can be estimated by numerically evaluating the integral of the pdf for the actual threshold used in the system. It is not always clear which class of pdfs should be used for this simulation method, or how the thresholds should be chosen. As with importance sampling, in most cases it is not possible to generate a confidence interval that describes the accuracy of the BER estimate [1, 10].

Other Variance Reduction Methods

There is no shortage of techniques that can be applied to the BER estimation problem. Extreme value theory is useful for some systems and research is currently being performed on large deviation techniques. These, and many other approaches all face the same fundamental problem. One must make assumptions concerning the behavior of an analytically intractable system and then exploit the assumptions to reduce the simulation execution time. A particular technique is useful only when the design engineer can clearly identify the assumptions that were made in the analysis, and verify that the assumptions apply to the system under study. The engineer will also need to verify the accuracy of the estimate produced by the simulation. While this is often straightforward for MC simulations, it can be a much more difficult problem for advanced BER estimation techniques. Table I discusses the concerns and advantages for some of the more common simulation techniques.

Simulation of Coded Systems

Coded communication links, especially those with large coding gains, may have such low error rates that Monte Carlo techniques, and even some of the variance reduction techniques mentioned above, cannot provide accurate BER estimates with reasonable simulation execution times. Often the only feasible approach to evaluating the performance of these systems is to determine the "raw" error rate of the symbols passed through the channel. Coding theory approximations and bounds can then be used to estimate the end-to-end performance of the coded system. Caution must be exercised when using this approach since a good understanding of coding theory is necessary. In addition, a perturbation analysis should be performed to determine how small changes in the estimated uncoded error rate will influence the calculated coded probability of error. For systems with large coding gains, very small errors in the estimated BER of the uncoded system can result in unacceptably large BER estimates for the coded system.

The BER estimation problem is essentially a numerical integration problem. The BER of a given system is the area under the tail of an unknown probability density function.

Technique \ Concern	Flexibility	Analytical difficulties	Computer execution times	Accuracy of BER estimate
Monte Carlo	Applicable to any system	None	Often prohibitively long	Measurable using confidence intervals
Variance reduction techniques Importance sampling Tail extrapolation Large deviation theory Extreme value theory Others	Applicable to many systems	Must select biasing schemes, thresholds, etc. Often more of an art than a science.	Variable, from extremely short to prohibitively long	Often very difficult to access
Semi-analytic simulation	Restricted largely to systems with linear receivers and AWGN channels	Need conditional BER statistics	Very short execution times	Exact to within accuracy of DSP models
Strictly mathematical analysis	Typically restricted to highly idealized systems	Numerous and well known	Zero	Exact

■ **Table 1.** *Summary of BER estimation techniques.*

Validation

Large simulation programs are often developed to produce a reasonably simple result, such as the BER under various operating conditions. Before a simulation result can be used in any meaningful way, such as a step in the design of a complex system, it is important that the user have confidence in the simulation result. There are many reasons why a simulation result may be inaccurate, such as insufficient data to form an accurate BER estimate (discussed above), conceptual errors such as modeling inaccuracies, and software bugs. Validation of a large and complex simulation program is an important, although sometimes difficult, undertaking.

Validation of a large program does not mean a line-by-line review of the source code. This type of evaluation is time consuming, error prone, and in most cases impractical. Such a review is also unlikely to reveal conceptual errors that were made when the software models were developed. When functions are supplied by outside vendors, source code may not be supplied, making line-by-line evaluation impossible. However there are a number of techniques that can be used with reasonable success.

As with any large system, the performance of each subsystem should be evaluated before it is integrated into the simulation. While this is a necessary step, it is not sufficient to guarantee the correct operation of the final simulation. System level tests are needed to validate the overall design. When MC simulations are used, one can compare signals at selected "test-points" in the simulation with the corresponding test-points in the hardware design. This may involve plotting the time domain waveform from the simulation and comparing it to an oscilloscope trace, or may involve calculating statistics of a signal such as a histogram, mean, variance, or power spectral density. Unfortunately, some of the more advanced BER estimation techniques do not produce these intermediate signals. Perturbation analysis can be helpful when validating a simulation. One can sometimes make a few changes to the simulation, and reduce the system to one that is analytically tractable. These changes are often minor from a software standpoint, such as temporarily replacing nonlinear amplifiers with linear amplifiers, or eliminating synchronization errors by passing allowing the transmitter and receiver to share a common time base. The simulation BER estimate can then be compared to a theoretical result. When these values agree, one can gain confidence that a significant portion of the simulation is correct. One can then return to the actual system with a higher level of confidence.

Analytic bounds on system performance and simulation results complement each other. The bounds can give assurance that the simulation results are reasonable, and the simulation results can guide analysis, indicating the tightness of various bounds. Since simulation packages are becoming increasingly common, powerful, and easy to use, it may now be reasonable to have redundant simulation efforts. Two separate development teams working with different simulation packages are unlikely to make the same coding and implementation errors. Even if a single team and package is used, it is helpful to use more than one simulation approach. For example, MC simulations have long execution times, but do not suffer from some of the problems that more advanced techniques face. It is helpful to write an MC simulation, and occasionally check the performance of an advanced technique with the results of an MC simulation.

Summary: Developing A Simulation

Simulation is a useful tool for the design and analysis of communication links. Indeed, for complex systems, such as are common today, some level of simulation is often essential if insights into system behavior and performance predictions are to be made. The usual steps in developing and using such a simulation are as follows.

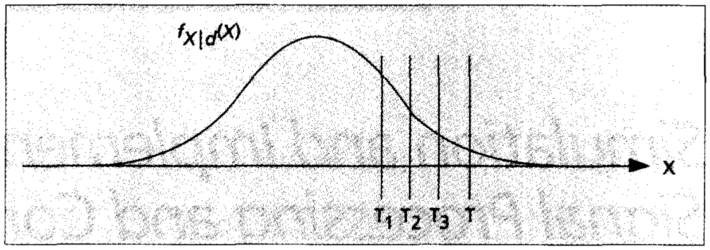
The first step is to develop a model of the system under study. This model often takes the form of a block diagram that defines the individual subsystems that make up the overall communication system. It is important to identify the approximations made in forming the system model. The important parameters of each subsystem must be identified so that they are carried through to the simulation.

The second step is to identify the signal processing operation necessary to define each of the subsystems in the overall communication system. At this point mathematical models for each subsystem are introduced. Thus, a choice is made concerning which signals are to be represented using complex envelope techniques. The strategy to use for representing analog filters by digital equivalents is also selected. One must appreciate the additional approximations incurred in this step.

The next step is to define the simulation products, which is the set of outputs required from the simulation. Examples are displays of the time-domain waveforms or the power-spectral density at a point in the system. If a performance prediction is to be made, such as the bit error rate for the overall communication system, the method to be used for estimating the performance must be selected. We have seen that a number of techniques may be applied to this important problem and that these techniques range from the Monte-Carlo method, which weights all errors equally and makes no assumption about the form of the decision metric, to more complex estimation schemes which do make assumptions about the decision metric. Recall that this decision allows one to expect a tradeoff between prior knowledge and computer execution time.

At this point the structure of the simulation is known and we can move to software. If a dedicated simulation language is to be used, one now selects models from the model library to implement the various subsystems in the overall communication system. One also selects a strategy for performance evaluation and this determines the estimation routines to be used in the simulation. Other simulation products, such as time-domain waveforms, spectra, and histograms are directed to a postprocessor that provides the tools for processing and displaying the data generated by a simulation. If one is developing code for a custom simulation, the previously selected signal processing and estimation strategies determine the code to be developed. After the simulation code has been developed and executed, one must ensure that the simulation results are reasonable. As previously discussed, this is the important area of validation.

In conclusion, it should be pointed out that for extremely complex systems, it is usually desirable to start out with the simplest model that incorporates only the essential features of the system under study. Simulations based on simple models are easier to verify and errors are more easily identified. The simulation can then be enhanced to include other interesting and important features of the communication system under study.



■ **Figure 9.** *Setting multiple thresholds for trail extrapolation.*

References

[1] M. Jeruchim, P. Balaban and K. Shanmugan, Simulation of Communication Systems, (New York, NY: Plenum Press, 1992).

[2] A. Oppenheim, and R. Schafer, Discrete-Time Signal Processing, (Englewood Cliffs, NY, Prentice-Hall, 1989).

[3] S. Haykin, Communication Systems, (New York, NY: Wiley, 1983).

[4] R. Rubinstein, Simulation and the Monte Carlo Method, (New York, NY: Wiley, 1981).

[5] P. Bratley, B. L. Fox and L. E. Schrage, *A Guide to Simulation* Second Edition, (Springer-Verlag, 1987).

[6] K. Park and K. W. Miller, "Random Number Generators: Good Ones are Hard to Find," *Comm. of the ACM*, vol. 31, no. 10, Oct. 1988.

[7] V. Volterra, Theory of Functionals and of Integral and Integro-Differential Equations, (Dover, New York, NY: 1959).

[8] T. T. Ha, Digital Satellite Communications, (Macmillan, New York, NY: 1986).

[9] A. A. M. Saleh, "Frequency-Independent and Frequency-Dependent Nonlinear Models of TWT Amplifiers," *IEEE Trans. on Comm.*, vol. COM-29, no. 11, Nov. 1981, pp. 1715-1720.

[10] M. Jeruchim, "Techniques for Estimating the Bit Error Rate in the Simulation of Digital Communication Systems, *IEEE JSAC*, Vol SAC-2, Jan. 1984, pp. 153-171.

[11] L. Bain and M Engelhardt, Introduction to Probability and Mathematical Statistics, (Boston, MA: Kent Pub. Co., 1987).

Biographies

WILLIAM H. TRANTER [F '85] received B.S.E.E., M.S.E.E., and Ph.D. degrees from the University of Alabama in 1964, 1965, and 1970, respectively. He joined the faculty of the University of Missouri-Rolla in 1969. His research interests are digital signal processing and computer-aided design of communication systems. His research efforts have been funded by NASA, the National Science Foundation, and G.E. Aerospace. He has recently consulted for the Education Development Corporation, Battelle Memorial Institute, and McDonnell-Douglas. He has authored numerous technical papers and is a co-author of two textbooks: *Principles of Communications* (Boston, MA: Houghton-Mifflin) and *Signals and Systems* (New York: Macmillan). From 1980 to 1985 he served as associate dean of engineering and was named Schlumberger Professor of Electrical Engineering in 1985. He is a member of Sigma Xi, Eta Kappa Nu, Tau Beta Pi, and ASEE. He currently serves as Editor-in-Chief of the *IEEE Journal on Selected Areas in Communications*.

KURT L. KOSBAR [M '88] received a B.S. in electrical and computer engineering from Oakland University, Michigan in 1982, M.S. and Ph.D. degrees in electrical engineering from the University of Southern Californiain 1984 and 1988 respectively. From 1982 to 1988 he was a member of the technical staff at Hughes Aircraft Company in the Space and Communications Group, where he spent the majority of this time in the Advanced Digital Signal Processing Department, designing and analyzing spacecraft-based digital demodulation systems for a variety of commercial and military applications. Since 1988, he has been an assistant professor of electrical engineering at the University of Missouri-Rolla. His primary research interests are synchronization, link level simulation of communication systems, spread spectrum communications, and adaptive equalization. His research has been supported by General Electric Aerospace, IEEE, the U.S. Air Force, the Engineering Foundation, and the National Science Foundation. He is a member of Tau Beta, Pi, Sigma Xi, Eta Kappa Nu and Phi Kappa Phi.