Computer-Aided Modeling, Analysis, and Design of Communication Systems: Introduction and Issue Overview

DIGITAL communications and computers are having a tremendous impact on the world today. In order to meet the increasing demand for digital communications services, engineers must design systems in a timely cost-effective manner. The number of technologies available for providing a given service is growing daily, covering transmission media (microwave and coaxial cable transmission), optical communications via fibers and space satellite transmission), devices (GAS and silicon devices and VLSI circuits), and software (protocols, system administration, diagnostics, and maintenance). The resulting design, analysis, and optimization of performance can be very demanding and difficult.

Over the past decade, a large body of computer-aided engineering techniques have been developed to facilitate the design process of complex technological systems. These techniques rely on models of devices and systems, both *analytic* and *simulation*, to guide the analysis and design throughout the life cycle of a system. Computer-aided design, analysis, and simulation of communication systems constitute a new and important part of this process.

Performance evaluations and tradeoff analysis are the central issues in the design of communication links and networks. Unfortunately, except for some idealized and often oversimplified cases, it is extremely difficult to evaluate the performance of a complex communication network in closed form using analytical techniques alone. Digital simulation provides a useful and effective adjunct to direct analytical evaluation of communication system performance. Indeed, there are many situations where explicit performance evaluation defies analysis and meaningful results can be obtained only through either actual prototype hardware evaluation or digital computer simulations. The former approach is generally cumbersome, expensive, time consuming, and relatively inflexible. These are considerations which typically weigh heavily in favor of computer simulation. Simulation also frees the analyst from a great deal of repetitive work involved in substituting numbers into formulas and tables and enables the analyst to concentrate on results. Another advantage is the insight into system performance provided both by the modeling process itself and by the experience gained from simulation experiments.

Computer-aided analyses can occur at any point in the life cycle of a communication product: before significant development effort, as an ongoing part of development, in manufacturing, through installation, and on out into field operations, service, and support. The results of these simulations should be checked with data or measurements on actual systems whenever possible because inevitably any model ignores some phenomena (confidence is needed in both modeling assumptions as well as the numerical parameters in the model). Again, computers can assist us: it may be quite difficult or expensive to gather data or measurements, so computer-assisted analysis can quantify the importance of the data gathering and analysis procedures.

We present, in this issue, a set of papers that collectively describes the state of the art in simulation-based analysis and design of point-to-point communication links and networks. While the primary emphasis of this issue is on software packages that can be used for communication system simulation, we have also included papers that deal with simulation techniques as well as applications and case studies in computer-aided design, simulation, and modeling of communication systems.

Fundamental to developing models and simulations are the concepts of digital communications, sampled data approximations of signals and systems, digital signal processing, random processes and queueing theory, numerical methods, statistical analysis, and Monte Carlo procedures. Several texts that deal with the above topics are listed at the end of this editorial. Additional references may be found in the reference sections of the papers that appear in this issue.

I. Simulation of Communication Systems

A communication network is made up of a set of interconnected transmission facilities and is designed to serve the needs of a variety of users. For simulation purposes, it is convenient to model the network using two layers:[1] a transmission system layer and a network layer (Fig. 1). The transmission system includes the transmission channel, modulators, demodulators, encoders, decoders, filters, and amplifiers. The network layer deals with issues such as message formatting, multiplexing, concentration, protocols, routing, switching, flow control, and other network control functions.

[1]For a more elaborate model of a communication network, the reader may refer to Tanenbaum's book [11].

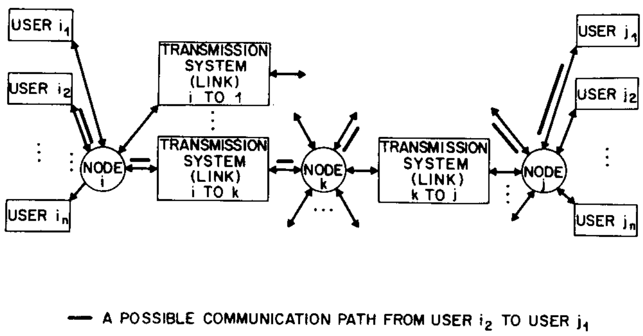


Fig. 1. Model of a communication network.

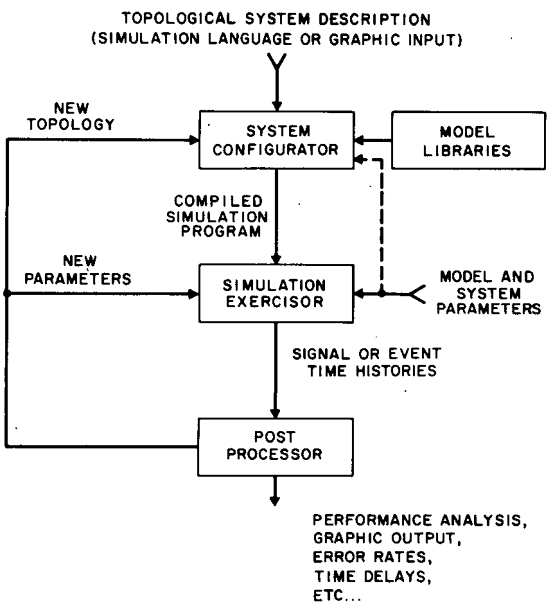


Fig. 2. Operational flow of a simulation package.

Several software packages are now used in the communication industry for simulating communication systems. While there is no clear-cut choice as to which one of these packages is the best, we list below a set of desirable features that a software simulation package should have.

A. Modular Structure

In order to provide a maximum flexibility, simulation software packages used to aid communication systems analysis and design should have a modular structure. Most of the software packages that are in use now are made up of four major components (Fig. 2):

- model library
- system configurator
- simulation exercisor
- postprocessor.

The system configurator is used to select a set of models of functional blocks from the model library and connect them in the desired topology as specified by the block diagram of the system being designed. Parameters of various functional blocks are specified either when the system is being configured or during the execution of the simulation, which is supervised by the simulation exercisor. Time histories of events or waveforms or bit streams at various points in the system are generated and stored by the simulation exercisor. These time histories are examined by the postprocessor at the conclusion of the simulation run, and performance measures are computed from the time histories. At the end of a simulation, the design engineer uses the postprocessor output to verify if the performance requirements and design constraints are met. If the design objective are not met, then several iterations are made until a suitable design is found.

B. Simulation and Programming Languages

Software should be written in a higher level *programming* language such as Fortran, Pascal, or C. Unfortunately, these languages do *not* allow input via block diagrams, which requires a preprocessor *simulation* language description of the system being analyzed or designed. This approach lends itself to portability and frees the user from having to know the details of the underlying operating system and hardware configuration of the simulation facility. For discrete event simulation, two widely used languages are GPSS and Simscript. Each of these are preprocessors to a lower level language. No such widely accepted standard languages exist for transmission system simulators. For additional flexibility the simulation software may permit the intermixing of statements written in the simulation and programming languages. This will result in a minor restriction of the free format input of blocks, namely, the models will have to appear in the order of their precedence in the description of the simulated system.

C. Topological Configuration

The model configurator of the simulation package should permit the design engineer to connect the functional blocks in any desired topological interconnection. While this free topological requirement may complicate the simulation software structure, it provides the maximum flexibility.

D. Model Library

The usefulness of a simulation package depends heavily on the availability of a model library that contains a large number of models of various functional blocks that make up transmission systems and networks. Computer routines that model functional blocks should be reentrant so that functional blocks may be used at multiple and arbitrary instances in the simulation of a transmission system. The model configurator should permit unlimited nesting of models such that subsystem models may be built using library models. It is also desirable to make provisions to enable the user to write his own model, use it in the simulation directly, and/or enter it into the model library.

E. Time and Event Driven Simulation

A simulator can be designed to be time driven where processing is initiated at every "tick" of the simulation clock, or event driven where processing takes place only when an event of interest (such as the arrival of a message) takes place. For maximum flexibility, provisions should be made for both modes of processing such that some blocks in the system are event driven whereas others could be time driven.

F. Check Pointing

The reason for checking the status of the simulation is to gather a variety of either transient or long-term time averaged statistics. If long-term time averaged statistics are of interest, then the question of how much data to gather at each sample point, and with what degree of confidence, will influence the simulation parameters. This monitoring feature can be very useful in long Monte Carlo simulations.

G. Postprocessor

The postprocessing routines are an important part of a simulation package since these routines are the ones that enable the design engineer to view the results of the simulation. The model configurator and the simulation exercisor should be designed to allow a designer to draw direct cause-and-effect inferences about system operation. As a minimum, the postprocessor package should have routines that perform the functions of common laboratory test instruments, both hardware (spectrum analyzers, power meters, error rate estimators, and oscilloscopes) and software (load generators, profilers of resource utilization for each job step and for time delays of each job step). Statistical analysis routines as well as graphics display routines are also essential.

H. User Interface

Finally, the whole simulation package should be made user friendly. This includes both on-line and off-line documentation and help. Two currently popular approaches here are using menus to trace a tree of commands or actions, and using key words (which appears to offer certain advantages for sophisticated users) by allowing menus to be bypassed if need be. This is currently an area of active research, encompassing novel interactive graphics and mouse controls, among other approaches. The simulation software should also have the following provisions to aid the user: symbolic debugging, run-time diagnostics, parameter checking, and on-line error control for inputting.

II. Simulation of Transmission Systems

The transmission system includes the physical link, modulators, demodulators, encoders, decoders, filters, amplifiers, transmitters, and receivers. The functional block diagram of a typical digital satellite transmission system is shown in Fig. 3 (other systems such as digital radio and fiber optic can be modeled similarly). The overall perfor-

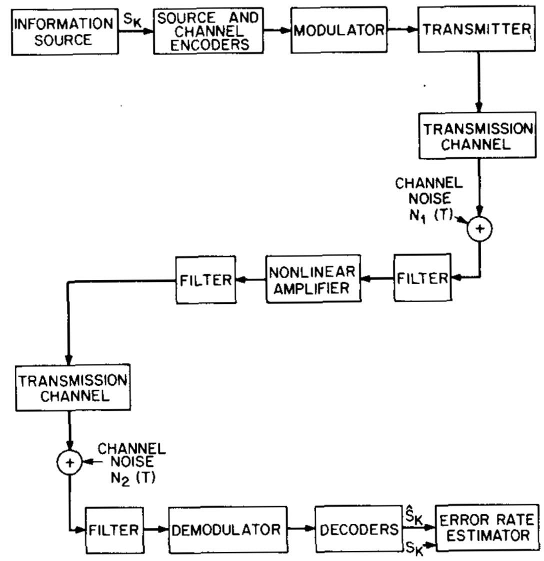


Fig. 3. Block diagram of a transmission system.

mance of the system is assessed by the probability of error. Closed-form analytical expression for the probability of error cannot be derived for this transmission system because of the presence of the nonlinear amplifier and bandlimiting filters. If the system was designed to carry analog information, then the overall performance will be assessed by the output signal-to-noise power ratio, which again will be difficult to compute analytically because of the presence of the nonlinearity.

Analysis and design of a transmission system such as the one shown in Fig. 3 can be carried out easily using simulations. For simulation purposes, the transmission process is assumed to consist of a sequence of signal processing operations. Sampled values of the signals are generated in the computer and operated on by a set of computer routines that simulate the effects of filtering, nonlinearities, and other functional blocks in the system. After the sampled signals pass through all the functional blocks, the output is compared with the input signal to determine the degradation introduced by the elements in the transmission system.

In order to reduce the sampling rate and the processing requirements, bandpass signals are usually represented by their complex envelopes, which are their low-pass equivalents. The bandwidths of various signals in a transmission system may sometimes vary over a wide range. In such situations it might be desirable to use variable sampling rates in simulation, sampling *each* signal at a rate appropriate for its bandwidth rather than sampling *all* the signals at the highest rate corresponding to signal with the larger bandwidth. While improving the simulation speed and reducing storage, this approach would require the use of signal interpolation algorithms.

Signals and systems can be represented either in the time domain or in the frequency domain. For linear functional blocks such as filters, it is advantageous to use frequency domain processing. Besides improved speed through the use of FFT, frequency domain processing permits the use of measured values of filter characteristics such as amplitude and phase response. For nonlinear elements, it may be easier to do the modeling and processing in the time domain. It is desirable that the simulation package permit the use of both time domain and frequency domain processing.

Signals used in the simulation process may be real, complex, or multidimensional functions of time. The simulation software should be capable of handling these types of signals and should also be able to process them on a point-by-point and/or block-by-block basis. In the block mode, a set of samples of the signals are passed to the processing routines per each call, which leads to a reduction in the simulation overhead. While block mode processing results in an increase in simulation speed, it cannot be used in parts of the system with feedback present and it does require more storage allocation for each processing routine.

If the signals and/or system parameters contain random variations, then Monte Carlo techniques are used to generate statistically sampled values of the random variations and the simulation is repeated in order to obtain a statistically valid measure of system performance. While this might require a great deal of processing, the computational burden can be somewhat reduced by combining analytical approaches with simulations. For the example shown in Fig. 3, while it may be necessary to use Monte Carlo techniques to introduce samples of $N_1(t)$ in the simulation process, the effects of the second noise source $N_2(t)$ on system performance can be handled analytically if the channel and receiver are time invariant and linear. In such a case, it is not necessary to generate and process samples of $N_2(t)$.

At the end of the simulation, the postprocessor package is used to view the waveforms, compute and display spectral plot, perform statistical analysis, and calculate the transmission system performance measured in terms of error rate or signal-to-noise ratio.

III. Communication Network Simulation

A communication network multiplexes and switches its transmission capacity among many users. The network building blocks perform the functions of switching, transmission, concentration, multiplexing, and demultiplexing. The network is used for two purposes, to control itself and to transmit information. Each subsystem is controlled via rules or protocols. Network design involves tackling a sequence of subproblems, each of which can be handled independently, for the throughput and delay for control and information flow. Modeling and simulation based on computer-aided design and analysis techniques can reduce this complexity to a manageable level, allowing systematic and quantitative exploration of design alternatives.

While Monte Carlo techniques are used for both transmission systems and network simulations, there are many differences that distinguish them.

- First, in transmission system simulation, the emphasis is on waveform and the simulation is time driven. The network simulation in contrast is event based, i.e., processing is driven by the occurrence of random events such as arrival of messages.
- Second, the workload is more complicated:

  a) multiple types of messages with different urgencies arrive at random or arbitrary time instants from different users connected to the network at different points;

  b) messages have different lengths or attributes (control or video/voice/data) and require different amounts of processing;

  c) messages are routed to different places in a network according to very complex routing and flow control procedures;

  d) different schedules or policies are used for controlling the order of execution of messages.
- Third, the computational burden placed on switching systems by different scheduling policies can be significant with what is acceptable in a well-engineered transmission system.
- Fourth, the functional blocks for transmission systems are made of hardware subsystems and the simulation software incorporates models of the hardware. Many of the functional blocks in a network are software or software controlled and it is relatively easy to incorporate them into simulation.

Because of these differences, the simulation software for transmission systems and network analysis and design are developed independently. However, since network simulation incorporates the results of transmission system simulation, there is a need for maintaining some compatibility between the two.

IV. Issue Overview

This issue is broken down into two groups of papers, those dealing with transmission systems engineering and those dealing with link and transport level system engineering. Within each category, there are three types of papers:

- simulation techniques useful for communication system engineering
- general purpose simulation packages for communication system engineering
- applications and case studies in computer-aided design, simulation, and modeling of communication systems.

These topics are not mutually exclusive, but the reader should find this a useful distinction in reading the issue.

A. Transmission Systems

The first paper in this group, by M. Fashano and A. L. Strodtbeck, describes a general purpose simulation package, SYSTID, which is one of the earliest and most comprehensive software packages developed for communication systems modeling and analysis. SYSTID makes use of both time-domain and frequency-domain techniques, permits free topology, and has a large library of reentrant functional modules. The package is operational on several mainframe and minicomputers. One version uses a dedicated array processor for faster throughput. Case studies of an 18 channel FDM satellite channel and a burst TDMA satellite system are presented.

A package with similar features, TOPSIM, is described in the next paper by M. Ajmone Marsan, S. Benedetto, E. Biglieri, V. Castellani, M. Elia, L. Lo Presti, and M. Pent. TOPSIM is a simulation package with a free topology and a large library of functional reentrant modules. The simulation is done strictly in the time domain. Filters specified in the frequency domain are converted to their transversal time domain equivalents. Error rates are estimated by semianalytic and importance sampling techniques. A complementary paper, by G. Canovai, M. Elia, L. Lo Presti, and M. Pent, describes a case study of a satellite channel with convolutional channel encoding using TOPSIM. The error rate of the analog channel is estimated first. An encoded digital channel with prescribed error statistics is simulated subsequently.

Another transmission system simulation package, ICS, and its features are discussed in the paper by J. W. Modestino and K. R. Matis. In contrast to SYSTID and TOPSIM, ICS uses fixed topology and makes use of an interactive menu-driven graphics approach for system configuration. Fixed topology, the use of an array-processor, and simulation code written in assembly language enhance the speed of the ICS package while somewhat restricting its flexibility and portability.

The next paper, by G. Benelli, V. Capellini, and E. Del Re, describes a fixed topology communication system simulator that provides both source and channel encoding capabilities. The channel can be selected as a pure digital link with a prescribed error rate or a simulated analog transmission channel. The analog channel is modeled in frequency domain. A case study of a mixed AM and PSK modulation channel is described.

Two more simulation packages are described in the next two papers. The paper by L. C. Palmer describes a command driven satellite communication system simulator. This paper gives a review of modeling methods of a communication channel including filters, nonlinearities, and performance evaluation. A case study of an optimum signal shaping for a satellite channel is described. The paper by W. D. Wade, M. E. Mortara, P. K. Leong, and V. S. Frost describes the simulation package ICSSM and its use for simulating spread-spectrum systems. ICSSM has many unique features such as the use of both time-step and event-based simulation, checkpoint triggering, and capabilities for network simulation.

The paper by W. R. Braun and T. M. McKenzie describes the simulation package CLASS and its use for evaluating the performance the tracking and data relay satellite services (TDRSS). While CLASS focuses on a particular communication system, some of the models and techniques can be used for other applications.

The next two papers, by D. G. Messerschmitt, describe a general purpose time-domain simulator, BLOSIM, and a transmission line modeling program, LINEMOD. BLOSIM is written in C and is intended to provide a highly structured environment for simulations. Its features include reentrant models that are hierarchically structured and signals that may have different sampling rates. The models are interconnected by first-in first-out buffers and are automatically ordered to provide flexibility. A case study of a full duplex data transmission system is provided. LINEMOD is a modeling program for transmission lines (including bridged taps). The derived models are used as blocks in the BLOSIM library. Both programs illustrate the value of structured languages for simulation programs.

The next paper, by M. C. Jeruchim, is a study of different methods of estimating the error rate of digital communication systems. The described methods include counting technics, semianalytical methods, and variance reducing techniques.

The next paper, by D. G. Duff, reviews models and methods suitable for computer-aided design of digital lightwave systems. Models for system degradation due to source waveform, fiber dispersion, and noise are developed. Various analytical methods used to compute error rate, eye margin, and power penalty are described and compared.

The paper by R. A. Harris and P. Kristiansen describes a method of a three-dimensional representation of error rates of a digital communication system as a function of sample time and carrier phase.

In the next paper, K. J. Gladstone and J. P. McGeehan describe the application of a new simulation model for fading in the mobile radio environment to both synchronous and quasi-synchronous full carrier AM sideband diversity schemes.

B. Communication Networks

The initial paper, by C. H. Sauer, E. A. McNair, and J. F. Kurose, develops a methodology for handling complex computer communication networks. This builds on analytic results in Jackson queueing networks as well as simulation to provide a structured approach to analyzing performance.

A companion paper, by K. Bharath-Kumar and P. Kermani, goes into much greater detail in describing a package for analyzing computer communication networks. There are a great many similarities in user interfacing, preprocessing, and output display in this package for link and transport systems engineering and in the packages for transmission systems engineering, even though the questions dealt with are very different.

The paper by D. J. Baker, A. Ephremides, and J. A. Flynn deals with a new communication system technology, *cellular radio*. Geographically separated mobile stations communicate using frequency division multiplexing of bursty time division multiplexed signals. One set of problems encompasses transmission systems engineering (modulation, multipath and diversity, coding); a second set encompasses controlling an intrinsically *distributed* system, with only local state information available at any one station.

The next paper, by P. J. P. O'Reilly and J. L. Hammond, Jr., is a case study of efficient simulation of a particular type of local area network based on *carrier sense multiple access with collision detection* (CSMA/CD). The number of states in even modest systems (e.g., ten stations or less) can quickly climb into the thousands, and this in turn leads to very long execution times and a demand for storage that can tax virtually any computer simulation technique. The approach presented here involves aggregating the states in order to reduce the storage and execution time.

A complementary paper on local networks, by J. R. Spirn, J. Chien, and W. Hawe describes a package as well as a case study in modeling and simulation of the bursty workload found in a data communications local area network. The underlying network is a bus controlled by CSMA/CD, while the workload is generated by the higher levels of the communication model described earlier.

The paper by L. F. Tolendino and M. O. Vahle is a case study in currently available techniques for simulating a large network of interactive terminals and computers. Typical issues addressed are how many terminals, terminal concentrations, computers, and disks are needed for interactive program development.

The paper by H. T. Mouftah and S. Bhatia is a case study in how simulation techniques can be used to design an interface unit to a communication system. A great deal of emphasis is placed on displaying data to allow a user to draw immediate cause-and-effect inferences, rather than indirect inferences.

V. Looking Forward

As you read the papers in this issue, it will become clear that simulation is widely used in the communication industry as an analysis and design tool. It will also be obvious that each organization has developed its own simulation package for transmission systems and there is no industry-wide standard, such as GPSS or Simscript for discrete event simulation or the SPICE package used for circuit simulation.

Simulation languages such as GPSS, Simscript, and Simula have been developed to describe realistic models of a wide variety of systems. General purpose computer programming languages such as Fortran, PL/1, Pascal, or C have been developed to enforce a structured, systematic approach to problem solving. In times to come, we should see the ideas and trends in simulation and general purpose computer language blur and merge. Both languages will allow *concurrency* to be readily expressed, for example, by means of queueing systems, block diagrams, coroutines, and so forth. The notion of *time* is apparently lacking in many general purpose computer programming languages, but is readily expressed in many different simulation languages; we expect this to become more widespread than at present.

Not only will the language become a closer match to the ideas we want to express, but the environment for doing so will become richer: color graphics input–output, symbolic (pictorial) debugging, with a variety of data display capabilities ranging from tables to charts to graphs available singly or in combinations, all will be available in times to come.

In addition to the development of software simulation packages, some efforts are currently underway towards the development of firmware simulation systems using a hybrid hardware/software computing approach. It is anticipated that hybrid computers using programmable digital signal processor chips will find increasing use in communication systems simulations before too long. Moreover, the application of fast digital signal processors to moderate data rates will bring us to the stage where the software simulation models of receivers and transmitters will be used *as is* in a firmware implementation of the actual system. Put differently, *the system simulation will be the system*.

Microprocessor-controlled workstations are becoming widely available, and we can expect to see simulations and computer-aided design tools available for running on these environments, as well as on *distributed* multiple processor configurations. Here the advantage of being able to specify what operations can take place *concurrently* will be a plus.

The coupling between computer-aided design and product development and manufacturing and on out to installation, service, diagnostics, and support will become closer: the design or simulation will specify each subsystem that will in turn feed a variety of other computer-controlled systems for building prototype communication systems, analyzing and measuring their behavior, and, after enough iteration, leading into manufacturing. In fact, relatively more effort than at present will be spent at the initial design stage on testing, service, diagnostics, and support because of its importance in profitability later in the life cycle of a product, and computer-aided design tools will start to reflect this. The precursor of this is seen today in the great effort devoted to custom design of very large scale integrated circuitry, with one or a few chips appropriately packaged taking the place of entire circuit packs, shelves, or bays of equipment.

Indeed, this is the challenge of the future: to use computer-aided design tools to get on with the revolution that is upon us all in information transmission and processing.

Acknowledgment

We want to thank all the reviewers whose efforts ensure the quality of the papers published in this issue. We also want to express our gratitude to Z. Straniero and D.

Shoeger who kept track of all the papers, reviews, and correspondences.

PHILIP BALABAN, *Guest Editor*
Bell Laboratories
Holmdel, NJ 07733

K. SAM SHANMUGAN, *Guest Editor*
University of Kansas
Lawrence, KS 66045

BARTON W. STUCK, *Guest Editor*
Bell Laboratories
Murray Hill, NJ 07974

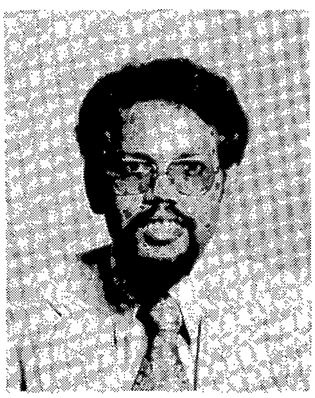
REFERENCES

[1] J. G. Proakis, *Digital Communications.* New York: McGraw-Hill, 1983.
[2] K. S. Shanmugan, *Digital and Analog Communication Systems.* New York: Wiley, 1979.
[3] R. W. Lucky, J. Salz, and E. J. Weldon, *Principles of Data Communications.* New York: McGraw-Hill, 1968.
[4] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1975.
[5] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1975.
[6] S. A. Tretter, *Discrete Time Signal Processing.* New York: Wiley, 1976.
[7] A. Papoulis, *Probability, Random Variables, and Stochastic Processes.* New York: McGraw-Hill, 1965.
[8] F. P. Kelly, *Reversibility and Stochastic Networks.* New York: Wiley, 1979.
[9] L. Kleinrock, *Queueing Systems, Vol. I: Theory; Vol. II: Applications.* New York: Wiley, 1976.
[10] M. Schwartz, *Computer Communication Network Design and Analysis.* Englewood Cliffs, NJ: Prentice-Hall, 1977.
[11] A. Tanenbaum, *Computer Networks.* Englewood Cliffs, NJ: Prentice-Hall, 1981.
[12] D. E. Knuth, *The Art of Computer Programming*, vols. 1–3. Reading, MA: Addison-Wesley, 1968–1973.
[13] R. I. Churchhouse, Ed., *Numerical Methods*, vol. 3 in *Handbook of Applicable Mathematics*, W. Ledermann, Ed. New York: Wiley, 1981.
[14] A. Ralston and P. Rabinowitz, *A First Course in Numerical Analysis.* New York: McGraw-Hill, 1978.
[15] R. Y. Rubinstein, *Simulation and the Monte Carlo Method.* New York: Wiley, 1981.
[16] J. P. C. Kleinjen, *Statistical Techniques in Simulation.* New York: Dekker, 1974.
[17] W. G. Bulgren, *Discrete System Simulation.* Englewood Cliffs, NJ: Prentice-Hall, 1982.
[18] G. S. Fishman, *Concepts and Methods in Discrete Event Digital Simulation.* New York: Wiley, 1973.
[19] H. Kobayashi, *Modeling and Analysis: An Introduction to System Performance Evaluation Methodology.* Reading, MA: Addison-Wesley, 1978.

Philip Balaban (A'52–M'55–SM'71) received the Dipl. Ing. degree from the Technical University, Munich, Germany, in 1950 and the Ph.D. degree from the Polytechnic Institute of Brooklyn, Brooklyn, NY, in 1966, both in electrical engineering.

He was a Senior Research Engineer at the Scientific Department, Ministry of Defense, Israel, from 1950 to 1956, where he was engaged in developing a general purpose analog computer. From 1956 to 1958 he was at Contraves, Zurich, Switzerland, where he supervised a group engaged in early work of developing a transistorized fire control computer. From 1958 to 1962 as Director of Research and Development at Computer Systems, Inc., New York, NY, he managed a team developing a hybrid computer system, DYSTAC. At the Polytechnic Institute of Brooklyn from 1963 to 1967, as Research Associate and Assistant Professor, he conducted research in communications, including video bandwidth compression. In 1970–1971 as a Visiting Professor at the Technion, Haifa, Israel, he organized and taught courses in CAD. He joined Bell Laboratories, Holmdel, NJ, in 1967, initially working in statistical modeling, optimization, and analysis of circuits. He also developed programs to evaluate capacitances for LSI circuits. He is currently responsible for the modeling, analysis, and simulation of fiber optic and satellite communication systems and has published extensively in this field. His most recent interests have been exploratory development of CAD tools for VLSI circuits for communication systems.

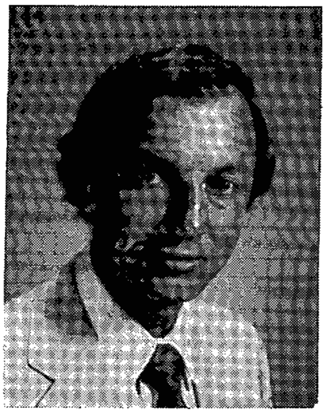
Dr. Balaban is a member of Sigma Xi. He received the Distinguished Member of Staff Award from Bell Laboratories in 1982.



K. S. Shanmugan (M'70–SM'74) received the B.E. degree from Madras University, Madras, India, in 1964, the M.E. degree from the Indian Institute of Science, Bangalore, India, in 1966, and the Ph.D. degree from Oklahoma State University, Stillwater, in 1970, all in electrical engineering.

Since 1970, he has held academic appointments at Oklahoma State University, Wichita State University, Wichita, KS, and the University of Kansas, Lawrence. His industrial experience includes visiting appointments at Bell Laboratories, Holmdel, NJ, and Hughes Aircraft Company, Los Angeles, CA. At the University of Kansas he is in charge of the undergraduate and graduate programs in communication systems engineering and he is the Director of the newly formed Telecommunications and Information Sciences Laboratory. His current research interests are in the areas of image processing and satellite communication systems. He has published over 50 research articles and is the author of a widely used textbook, *Digital and Analog Communication Systems* (New York: Wiley, 1979).

Dr. Shanmugan is a member of the American Society for Engineering Education and the Society for Automative Engineers. He serves as a reviewer for several journals and is an Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS. He received the Outstanding Young Engineering Faculty Award from the Society of Automative Engineers in 1979, the 1982 Henry E. Gould Award for distinguished service to undergraduate engineering education, and the 1982 Amoco Foundation Award for Outstanding Teaching at the University of Kansas.



Barton W. Stuck (S'67–M'72) was born in Detroit, MI, on October 25, 1946. He received the S.B., S.M., and Sc.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1969, 1969, and 1972, respectively.

Since joining Bell Laboratories, Murray Hill, NJ, in 1972, he has worked on a variety of computer and computer communication systems, encompassing telephone call processing, network management of private line voice and DDD networks, automatic call distributors, directory assistance, repair service bureau, billing order support, satellite video conferencing, remote PBX testing, voice storage services, packet switching, local area networks, and integrated voice and data switching.

Dr. Stuck is a member of the Association for Computing Machinery, the Operations Research Society of America, and the Mathematical Association of America. He is a Member of the Board of Governors of the IEEE Communications Society for 1984–1986, and is Chairman of the IEEE Computer Society Project 802 Local Area Network Standards Technical Advisory Group on Traffic Handling Characteristics. He was Co-Guest Editor (with K. Kuemmerle and F. A. Tobagi) of the November 1983 IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Special Issue on Local Area Networks.

Communication System Simulation and Analysis with SYSTID

MICHAEL FASHANO AND ANDREW L. STRODTBECK, MEMBER, IEEE

***Abstract*—The design of satellite and deep space communications systems frequently involves problems that cannot be solved through pure analysis. In these cases simulation is an invaluable system design tool. This paper describes a flexible communication system simulation package, SYSTID, which has been under continual development and use at Hughes Space and Communications Group for over 15 years. We discuss the history of SYSTID's development as well as its internal workings. Both the software structure and analytic techniques are described. The paper includes several examples, one of which demonstrates SYSTID's topological input language. Hardware comparisons which show the accuracy and limitations of the simulation approach are also presented.**

I. INTRODUCTION

A communication system simulation package, SYSTID, has been extensively used at Hughes Aircraft Company for several years. Based on work that commenced in the mid-1960's, SYSTID has evolved into a comprehensive software system that is currently hosted on a minicomputer/array processor facility known as the communications systems simulator (CSS). SYSTID has been developed through an interdisciplinary approach using an appropriate blend of communication analysis, computer sciences, numerical methods, statistics, and digital signal processing. The implementation of SYSTID on the CSS provides flexible topology simulations at speeds comparable to the best fixed topology simulation packages. Thus, new simulations that execute quickly can be developed quickly as well.

SYSTID and the CSS are used to simulate a large variety of signal processing and communications systems. Some typical applications include:

- bit error rate sensitivity
- adjacent and cochannel interference
- multicarrier FDMA frequency planning

Manuscript received September 1, 1983; revised October 1, 1983.
The authors are with the Space and Communications Group, Hughes Aircraft Company, Los Angeles, CA 90009.

0733-8716/84/0100-0008$01.00 ©1984 IEEE