

AD-A151 276

INTERACTIVE COMMUNICATIONS SIMULATOR (ICS)(U) PAR
TECHNOLOGY CORP NEW HARTFORD NY K R MATIS NOV 84
PAR-84-34 RADC-TR-84-227 F30602-82-C-0026

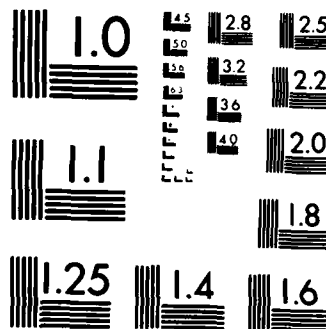
1/1

UNCLASSIFIED

F/G 9/2

NL

END



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



(2)

RADC-TR-84-227
Final Technical Report
November 1984

INTERACTIVE COMMUNICATIONS SIMULATOR (ICS)

AD-A151 276

PAR Technology Corporation

Kurt R. Matis

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DTIC
ELECTE
MAR 12 1985
S D
B

DTIC FILE COPY

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

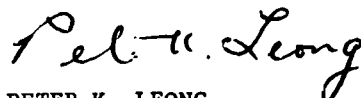
85 02 27 002

AD A 151 276

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-84-227 has been reviewed and is approved for publication.

APPROVED:



PETER K. LEONG
Project Engineer

APPROVED:



BRUNO BEEK, Technical Director
Communications Division

FOR THE COMMANDER:



DONALD A. BRANTINGHAM
Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (DCLF) Griffiss AFB NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AL A151 076

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b. RESTRICTIVE MARKINGS N/A										
2a. SECURITY CLASSIFICATION AUTHORITY N/A		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited										
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A		5. MONITORING ORGANIZATION REPORT NUMBER(S) RADC-TR-84-227										
4. PERFORMING ORGANIZATION REPORT NUMBER(S) 34-34		7a. NAME OF MONITORING ORGANIZATION Rome Air Development Center (DCLF)										
6a. NAME OF PERFORMING ORGANIZATION PAR Technology Corporation		7b. ADDRESS (City, State and ZIP Code) Griffiss AFB NY 13441-5700										
6b. ADDRESS (City, State and ZIP Code) Route 5, Seneca Plaza New Hartford NY 13413		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F30602-82-C-0026										
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Rome Air Development Center		8b. OFFICE SYMBOL (If applicable) DCLF										
8c. ADDRESS (City, State and ZIP Code) Griffiss AFB NY 13441-5700		10. SOURCE OF FUNDING NOS. <table border="1"><tr><td>PROGRAM ELEMENT NO.</td><td>PROJECT NO.</td><td>TASK NO.</td><td>WORK UNIT NO.</td></tr><tr><td>35885G</td><td>1014</td><td>02</td><td>12</td></tr></table>		PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.	35885G	1014	02	12	
PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.									
35885G	1014	02	12									
11. TITLE (Include Security Classification) INTERACTIVE COMMUNICATIONS SIMULATOR (ICS)												
12. PERSONAL AUTHOR(S) Kurt R. Matis												
13a. TYPE OF REPORT Final		13b. TIME COVERED FROM Jan 82 TO Jun 83										
14. DATE OF REPORT (Yr., Mo., Day) November 1984		15. PAGE COUNT 100										
16. SUPPLEMENTARY NOTATION N/A												
17. COSATI CODES <table border="1"><tr><td>FIELD</td><td>GROUP</td><td>SUB. GR.</td></tr><tr><td>17</td><td>02</td><td></td></tr><tr><td>09</td><td>04</td><td></td></tr></table>		FIELD	GROUP	SUB. GR.	17	02		09	04		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Communication Link Modeling and Simulation, Communications Signal Processing Techniques, Computer-aided Design and Analysis Tool.	
FIELD	GROUP	SUB. GR.										
17	02											
09	04											
19. ABSTRACT (Continue on reverse if necessary and identify by block number) This report describes the current capabilities of the Interactive Communications Simulator (ICS) and the models employed in it. The ICS is a computer-aided modeling, simulation, design and analysis tool for a wide variety of point-to-point digital communication systems. The simulator has been used to evaluate existing modem performance, and to explore new modulation/coding and signal processing concepts pertinent to military, commercial, and space applications. The ICS is a flexible, graphics oriented, high speed, high resolution, and user friendly hardware/software system consisting of a PDP 11 series or VAX minicomputer acting as host to a peripheral array processor, the Floating Point Systems AP-120B or FPS-100. It employs a fixed modeling topology to represent communication links; their signal processing functions (source/sink, source encoding/decoding, channel encoding/decoding, modulation/demodulation, transmitter/receiver), propagation channels and electronics interferences. (Cont'd on reverse)												
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS <input type="checkbox"/>		21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED										
22a. NAME OF RESPONSIBLE INDIVIDUAL Peter K. Leong		22b. TELEPHONE NUMBER (Include Area Code) (315) 330-4567										
		22c. OFFICE SYMBOL RADC (DCLF)										

DD FORM 1473, 83 APR

EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

It has an extensive graphics package, channel and filter design packages, and many powerful analysis subroutines, to facilitate user interactions, link design, analyses, and output displays. All simulation modules representing signal processing functions and some analysis subroutines are written in AP Assembly Language, all other software are written in DEC FORTRAN. *— see beyond index, 20, 21*

DTIC
ELECTE
MAR 12 1985
S B D

Accession For	
NTIS	<input checked="checked" type="checkbox"/>
DTIC	<input type="checkbox"/>
Unpublished	<input type="checkbox"/>
Distribution	
Availability Codes	
Dist	Avail and/or Special
A-1	



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

TABLE OF CONTENTS

SECTION	DESCRIPTION	PAGE
1.	INTRODUCTION	1-1
2.	GENERAL BACKGROUND.....	2-1
3.	FUNCTIONAL DESCRIPTION OF THE ICS.....	3-1
3.1	SYSTEM DESCRIPTION.....	3-1
3.2	SYSTEMS MODEL.....	3-2
	Channel Encoder.....	3-6
	Modulator/Transmitter.....	3-9
	Spread-Spectrum Modulation.....	3-13
	Channel.....	3-18
	Generic-Channel Design.....	3-29
	Explicit Diversity.....	3-33
	Electronic Countermeasure (ECM) Simulation.....	3-36
	Demodulator/Receiver.....	3-39
	Channel Decoder.....	3-43
3.3	OPERATION OF THE ICS.....	3-43
3.4	TYPICAL GRAPHICAL OUTPUT.....	3-48
4.	CONCLUSIONS AND RECOMMENDATIONS.....	4-1
	REFERENCES.....	R-1

APPENDIX A, BLOCK CODES.....	A-1
APPENDIX B, NARROWBAND FILTERING.....	B-1
Design of IIR Filters.....	B-2
Digital Butterworth Filters.....	B-5
Digital Chebyshev Filters.....	B-10
Design of FIR Filters.....	B-11
Appendix B References.....	B-19

LIST OF FIGURES

FIGURE	DESCRIPTION	PAGE
2-1	Hardware Configuration Supporting Interactive Communications Simulator.....	2-3
3-1A	Initialization.....	3-3
3-1B	Simulation Mode.....	3-4
3-2	Block Diagram of General Communication System.....	3-5
3-3	Illustration of $K=3$, $R=1/2$ Convolutional Encoder.....	3-8
3-4	Direct-Sequence Spread-Spectrum System.....	3-15
3-5	Frequency-Hopped Spread-Spectrum System.....	3-17
3-6	General Narrowband-Filtering Operation.....	3-19
3-7	Overall Block Diagram of Channel Model.....	3-20
3-8	Tapped-Delay-Line Model of Diffuse Multipath Generator.....	3-22
3-9	Illustration of Impulsive Channel Noise Model.....	3-24
3-10	Simple Pulse Waveform Model for Low-Density Shot Noise.....	3-28
3-11	Equipment Configuration for (a) Frequency-Diversity and (b) Space-Diversity Systems.....	3-34
3-12	Predetection Portion of Receiver.....	3-40
3-13	Model for Zero-Memory Nonlinearities of the Limiting Variety.....	3-42
3-14	Software Organization of Interactive Communication Simulator.....	3-47
3-15	ICS Waveform Analysis.....	3-50
3-16	ICS Waveform Analysis.....	3-51
3-17a	ICS Error Analysis, Symbol Error Probability.....	3-53
3-17b	ICS Error Analysis, Bit Error Probability.....	3-54
3-18	Channel-Scattering Function.....	3-55

B-1	Block-Diagram Representation for a General Nth-Order Difference Equation.....	B-3
B-2	Mapping of the Analog Frequency Axis onto the Unit Circle Using the Bilinear Transformation.....	B-6
B-3	Mapping of the S-Plane into the Z-Plane Using the Bilinear Transformation.....	B-7
B-4	Dependence of Butterworth Magnitude Characteristic on the Order N.....	B-9
B-5	Tolerance Scheme.....	B-16
B-6	First Transmitter Filter.....	B-18

LIST OF TABLES

TABLE	DESCRIPTION	PAGE
3-1	Description of Some Typical Modulation Capabilities Available in Interactive Communication Simulator.....	3-12
3-2	Pulse Waveforms Available in Impulse Noise Generator.....	3-30

1. INTRODUCTION

As the sophistication and complexity of modern digital communication systems have increased, so has grown the need for realistic and timely simulation of these systems. This need has provided the impetus for the development of the Interactive Communications Simulator (ICS), a flexible, graphics-oriented, and highly interactive hardware/software system consisting of a typical minicomputer acting as host to a fast peripheral array processor.

The ICS is continually under development to reflect new trends in the development of communications algorithms and technology. The system has been used both to evaluate existing modem performance and to explore new modulation/coding and signal-processing concepts pertinent to military, commercial, and space applications.

The purpose of this document is to describe the models employed in the ICS and to provide a cursory tutorial on the general concepts of digital communications.

2. GENERAL BACKGROUND

Digital simulation provides a useful and effective adjunct to direct analytical evaluation of communication-system performance. Indeed, there are situations in which explicit performance evaluation defies analysis, and results can be obtained only through either actual hardware evaluation or digital simulation. The speed and flexibility associated with digital simulation generally provide compelling reasons for adopting this approach.

Unfortunately, an accurate simulation of most communication systems of even moderate complexity can be terribly time consuming on present-day general-purpose machines. This is due to the large number of repetitive signal-processing operations that must be performed in order to obtain a statistically valid measure of system performance. Recently, a class of fast floating-point array processors has become available; this greatly facilitates these repetitive signal-processing operations. The use of the word "array" in this context is intended to indicate a processor optimized for handling large data arrays in distinction to the large machines organized as parallel arrays of independent processing elements. These array processors are intended to be employed as peripheral floating-point processors in conjunction with a general-purpose host computer which provides overall system control. In the ICS, the general-purpose host provides an interface to the user for graphics input and display functions. All signal-processing tasks are relegated to the array processor. This hardware configuration provides a unique opportunity for accurate and statistically meaningful digital simulation of existing and future communication systems.

The ICS has exploited the potential of this hardware configuration. This has resulted in an extensive hardware/software system for the digital simulation of arbitrary point-to-point communication systems. While the system can be extended to include analog communications, the primary interest is in digital communication systems. This system makes extensive use of interactive computer graphics and includes a Digital Equipment Corporation VAX-11/780 or PDP-11/40 acting as host to a Floating Point Systems, Inc., AP-120B floating-point array processor. A block diagram of the hardware

configuration supporting the ICS software is illustrated in Figure 2-1.

The user can configure a wide variety of digital communication systems from basic modules provided as system facilities. These facilities include: channel coders/decoders; modulators/demodulators (modems) for a number of modulation strategies; receiver front-end filtering; and various bit-synchronization and phase-tracking algorithms. In addition to additive white Gaussian noise (AWGN) channels, the simulator includes the ability to model impulsive noise channels as well as fading-dispersive channels typical of, say, HF or tropospheric-scatter channels. ICS also provides an extensive ECM signal-simulation facility as well as an explicit diversity capability.

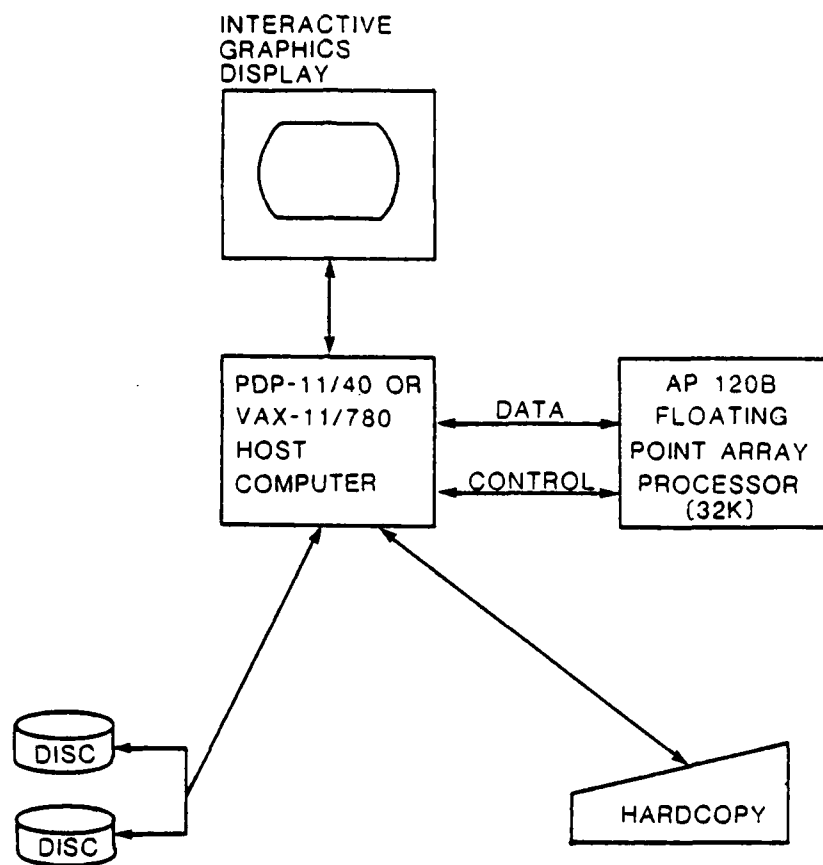


FIGURE 2-1
HARDWARE CONFIGURATION SUPPORTING INTERACTIVE
COMMUNICATIONS SIMULATOR

3. FUNCTIONAL DESCRIPTION OF THE ICS

In this section a complete functional description of the ICS is provided. The general systems model is discussed, after which the various usage modes of the ICS are described. Finally, some typical graphical outputs from the ICS are discussed.

3.1 SYSTEM DESCRIPTION

The use of an attached processor in the configuration of Figure 2-1 has become very popular for a general-purpose signal-processing system. Such a configuration can provide an excellent computation/cost ratio in a variety of applications. The availability of a comprehensive set of library routines for such array processors allows programmers versed only in FORTRAN or some other high-level language to take advantage of the peripheral processor's computational capabilities. The use of an array processor in such an environment involves three steps. First, data which is to undergo computation is transferred to the array processor. Second, parameters are passed to the AP, along with instructions to run the appropriate library function. Finally, the resultant data is transferred back to the host. Depending on the length of the data set transferred, and the number of computations performed in the AP each time this process takes place, a substantial increase in throughput may ensue.

Unfortunately, the above procedure is unsuitable for the very specialized types of signal-processing algorithms involved in communications simulation. Many of the functions are highly bit oriented and therefore unsuitable for simulation using standard library routines. Also, the overall computational efficiency of a system used in this mode is directly proportional to the number of computations performed during each run of the AP and inversely proportional to the amount of data transferred between the two computers.

As a result of these considerations, to increase throughput, all of the specialized signal-processing operations implemented in the ICS are coded in the AP-120B assembly language, APAL. This was a decision made early in the development of the ICS, and it has resulted in a throughput increase over a comparable host-based system of about three orders of magnitude. In the simulation mode, no data is transferred between the host and AP. Instead, all data is generated, processed, and analyzed in the AP. Once programs and data have been loaded in the initializations mode, the host supplies only the minimal amount of control necessary to monitor the progress of the simulation. The data transfer, processing, and control operations for the initialization and simulation phases are illustrated diagrammatically in Figures 3-1(a) and 3-1(b).

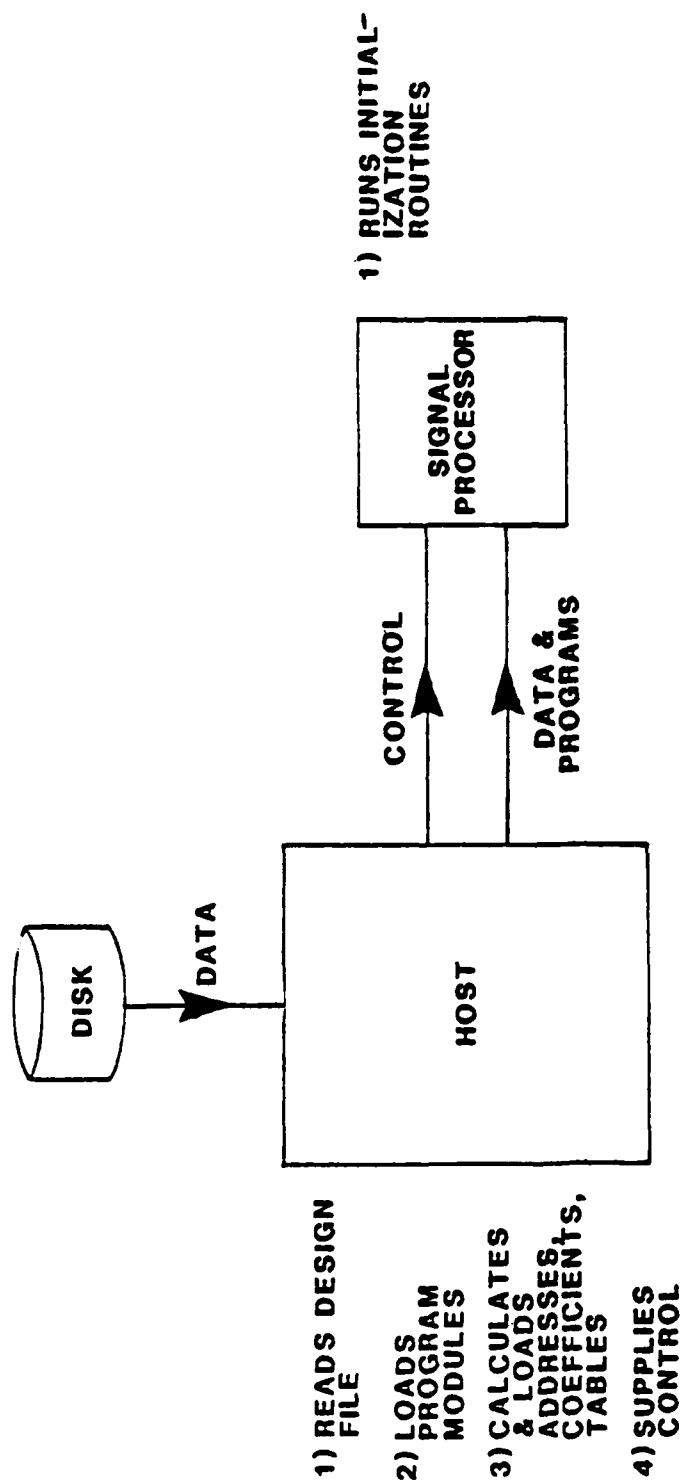
The internal processing in the AP-120B has been improved as part of the ICS enhancements with the design of an internal AP-120B executive program. This set of routines has greatly simplified the interaction of modules and submodules within the AP-120B. The executive program facilitates efficient transfer of data and parameters between various assembly language modules resident in the AP-120B. The executive is called by running programs to store or retrieve data operated upon by other programs without interaction with the host. This feature has permitted a good deal of code standardization and has increased overall system throughput.

ICS is a highly graphics-oriented system. Through graphics modularity, a high-level graphics display executive interfaces to a variety of graphics terminals including Tektronix, Ramtek and IMLAC displays. Device independence is maintained to a large degree by employing low-level translation libraries appropriate for each specific display.

3.2 SYSTEMS MODEL

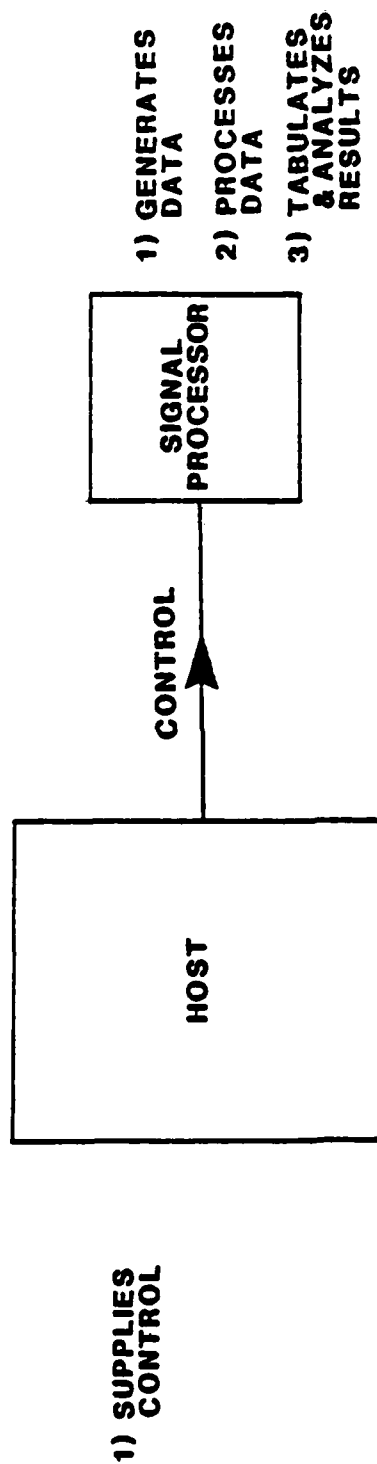
A block diagram of a generic communication system which provides a model for simulation-software development is illustrated in Figure 3-2. The information source generally produces a discrete-sequence or analog waveform which is to be encoded, transmitted over a specified channel, reconstructed, and delivered to a remote destination or information sink. In Figure 3-2, the purpose of the source encoder is to encode the source output, presumably in an efficient fashion, into a binary sequence $\{a_i\}$

PROCESSOR-HOST INTERFACES

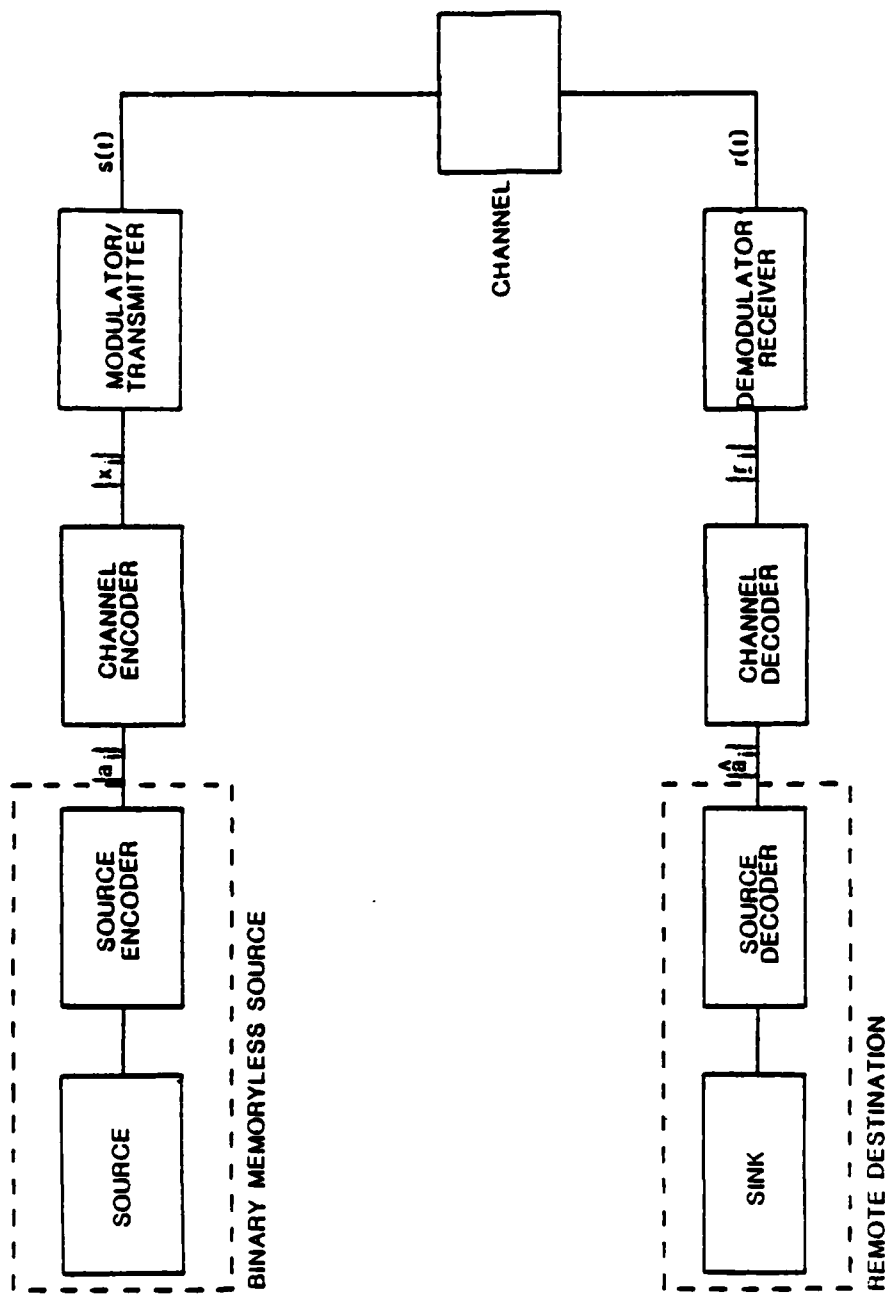


INITIALIZATION
FIGURE 3-1A

PROCESSOR-HOST INTERFACES



SIMULATION MODE
FIGURE 3-1B



BLOCK DIAGRAM OF GENERAL COMMUNICATION SYSTEM
FIGURE 3-2

for subsequent encoding and transmission. The source decoder at the remote destination provides the inverse function. That is, it employs the binary sequence $\{\hat{a}_i\}$ to reconstruct an approximation to the source output. The binary sequence $\{\hat{a}_i\}$ may differ from the actual transmitted sequence $\{a_i\}$ due to channel errors. This has an obvious effect on the accuracy with which the source decoder can reconstruct the source output.

Although source coding/decoding schemes could easily be incorporated into the simulator, they were not, because the primary purpose of the ICS is the evaluation of digital communication transmission. As a result, it is assumed that the combination of source and source encoder can be modeled as a binary memoryless source as indicated in Figure 3-2. The major criterion of system performance is then the bit error probability P_b , i.e., the probability that $\hat{a}_i \neq a_i$. This quantity can be evaluated by straightforward Monte Carlo simulation. There are, however, other modes of usage of this system. Each of the remaining elements in Figure 3-2 will be described in detail. The binary memoryless source is implemented as the modulo-2 sum of a linear feedback shift register sequence of length $2^{47} - 1$ and a mixed congruential random-number generator. It has a period exceeding $2^{47} - 1$.

Channel Encoder

The purpose of the channel encoder is to accept the binary sequence $\{a_i\}$ at its input and, through the insertion of controlled redundancy, to produce the M-ary sequence $\{x_i\}$ at its output. Generally, the encoded output sequence is such that $x_i \in 0, 1, \dots, M-1$, although in the binary case it will be convenient to assume that $x_i = \pm 1$, $i = 1, 2, \dots$. Both block and convolutional channel-encoding capabilities are provided in the simulator. In the former, the binary sequence $\{a_i\}$ is segmented into blocks of size k to which $n-k$ redundant bits are added to produce a rate $R=k/n$ code measured in units of information bits per transmitted channel symbol. There is complete independence between blocks. Convolutional codes, on the other hand, do not impose a block structure on the input-information sequence; they do produce a sliding dependency over a span of input symbols. The encoder can be implemented as a shift

register into which information bits are shifted 'b' at a time. After each shift, n modulo two sums are computed from the contents of selected shift register stages. These n modulo two sums then constitute the encoder output sequence associated with a particular branch of b-input information bits. The normalized code rate in this case is $R=b/n$. Note that each input information bit affects* $\lceil k/b \rceil n$ encoder output symbols where k is the length of the shift register. The quantity k is called the constraint length of the code. A typical $k=3$, $R=1/2$ encoder is shown in Figure 3-3. Tabulations of good convolutional code constructions are available for various k and R (cf. [1], [2], [14]) values. A description of block codes can be found in Appendix A.

The existing block and/or convolutional codes are generally effective in correcting random errors but seriously deficient in their ability to correct burst errors. In several important applications, the channel environments are known to introduce burst errors due to time-correlated fading, multipath, co-channel interference, jamming, etc. There are few constructive coding approaches for channels with memory. Indeed, the few existing approaches are extremely sensitive to channel-modeling assumptions which are often gross estimates. In the vast majority of existing systems, the approach is to employ appropriate interleaving schemes in an effort to make the channel appear memoryless. In particular, the serial channel symbol sequence $\{x_i\}$ is interleaved or scrambled in such a fashion that successive channel symbols are transmitted separated by many channel-signaling intervals. This interleaving is generally quite effective in reducing the effect of burst errors and allows use of the well-developed classes of random-error-correcting codes. The interleaver implemented in the ICS is of the convolutional type with a maximum symbol separation of 30. It is a particular implementation of a type 1 optimal interleaver described by Ramsey [20].

* The notation $\lceil x \rceil$ means the largest integer not exceeding x.

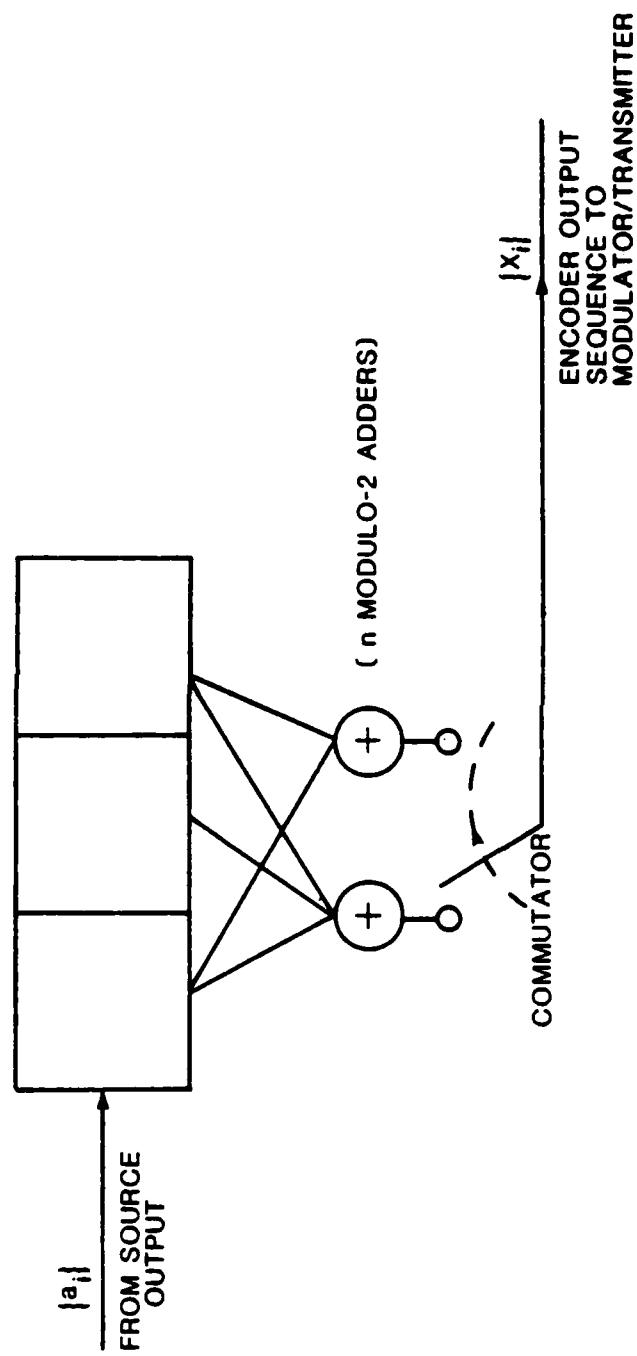


ILLUSTRATION OF $K=3$, $R=1/2$
CONVOLUTIONAL ENCODER
FIGURE 3-3

Modulator/Transmitter

The purpose of the modulator/transmitter is to map the channel encoder output sequence $\{x_i\}$ (possibly interleaved) into a waveform $s(t)$ suitable for transmission over the channel. Encoder outputs are presented to the modulator/transmitter every T_s seconds. T_s is called the baud interval, or equivalently $f_s = 1/T_s$ is the baud rate. It is convenient to normalize all frequency (time) domain quantities to the baud rate (interval). For example, the bandwidths of frequency-selective channel-filtering elements are all normalized to f_s . The filter cutoff, or 3-dB point, would then be specified in digital frequency in units of cycles per baud. This allows the user the ability to configure and execute a simulation program independent of the actual baud rate, which in turn facilitates user interaction with the communications simulator and provides simulation results in a convenient parametric form.

In simulating the channel-signaling waveform $s(t)$, complex notation is extensively used. This allows simulation of a bandpass signal in terms of its lowpass complex envelope and eliminates the need for simulating a high-frequency carrier component. Specifically, it is assumed that $s(t)$ can be expressed as

$$s(t) = \sqrt{2} \operatorname{Re} \{ \tilde{s}(t) e^{j2\pi f_c t} \} , \quad (1)$$

where f_c is an *assumed known* carrier component and $\tilde{s}(t)$ represents the *complex envelope*, which can be represented in the form

$$\tilde{s}(t) = s_c(t) - j s_s(t) , \quad (2)$$

where $s_c(t)$ and $s_s(t)$ are real quantities representing the inphase and quadrature (I/Q) components, respectively. By generating only the I/Q components, the simulator need not be concerned with the explicit details of the RF portions of the system. In particular, discrete samples of the I/Q components are generated at equally spaced intervals.

In the construction of the ICS, it has proven sufficient to consider that $\tilde{s}(t)$ can be expressed as

$$\tilde{s}(t) = \sqrt{\frac{E_s}{T_s}} \sum [C_{1,i} u_c(t - iT_s - \tau) + j C_{2,i} u_s(t - iT_s - \tau)] e^{j\Theta}, \quad (3)$$

where τ and Θ represent random timing epoch and carrier phase, respectively. The receiver will be required to estimate and track these initially unknown quantities. Here the sequences $\{C_{1,i}\}$ and $\{C_{2,i}\}$ are determined from the channel-encoder output sequence $\{x_i\}$, while $u_c(t)$ and $u_s(t)$ are elementary baseband-signaling waveforms. By appropriate choice of the mappings $\{C_{1,i}\}$ and $\{C_{2,i}\}$, together with the baseband-signaling waveforms $u_c(t)$ and $u_s(t)$, the expression for \tilde{s} is sufficiently general to include most known modulation formats. These include: coherent binary phase-shift keying (BPSK), differentially coherent phase-shift keying (DPSK), quadrature phase-shift keying (QPSK), noncoherent frequency-shift keying (FSK), and quadrature amplitude-shift keying (QASK). In the case of the offset or staggered quadrature phase-shift keying (OQPSK), and the constant envelope formats such as continuous phase frequency-shift keying (CPFSK), and the special case of minimum-shift keying (MSK) (cf. [3], [4]), this formulation must be modified slightly. Specifically, in this case

$$\tilde{s}(t) = \sqrt{\frac{E_s}{T_s}} \sum [C_{1,i} u_c(t - 2iT_s - \tau) + j C_{2,i} u_s(t - 2iT_s - \tau)] e^{j\Theta}, \quad (4)$$

where now the elementary baseband waveforms are translated by two baud intervals at a time.

In the case of coherent BPSK, for example, $C_{2,i} = 0$, $u_s(t) = 0$, while $C_{1,i} = x_i$ (assume $x_i = \pm 1$) and $u_c(t)$ is equal to the pulse-like waveform^{*}

^{*} The pulse waveform $u_0'(t)$ in Table 3-1 has value unity over the interval $[-T_s, T_s]$ and zero elsewhere.

$$\begin{aligned} u_0(t) &= i ; \quad 0 \leq t \leq T_s \\ &= 0 ; \quad \text{elsewhere} . \end{aligned} \quad (5)$$

Similarly, in the case of noncoherent binary frequency-shift keyed (BFSK) modulation it is easily shown that $C_{1,i} = \cos\Theta_i + j\sin\Theta_i$ while $C_{2,i} = x_i C_{1,i}$. Here $\{\Theta_i\}$ represents a sequence of independent and identically distributed (i.i.d.) phases uniformly distributed on $[-\pi, \pi]$. Without this latter assumption, the phase could be estimated at the receiver on the basis of past transmissions, thereby violating our assumption of noncoherent reception. The corresponding baseband signaling waveforms are given by

$$u_c(t) = \cos(\Delta\omega t/2) u_0(t) , \quad (6a)$$

and

$$u_s(t) = \sin(\Delta\omega t/2) u_0(t) , \quad (6b)$$

where $\Delta\omega$ is some integer multiple of $2\pi/T_s$ in order to ensure orthogonality between transmitted tones. Additional choices of $\{C_{k,i}\}$ $k=1,2$, and associated baseband pulse shapes $u_c(t)$ and $u_s(t)$, are provided in Table 3-1.

It should be noted that, in constructing Table 3-1, the constraint has been applied:

$$\int_{(i-1)T_s}^{iT_s} |\bar{s}(t)|^2 dt = E_s ; \quad i = 1, 2, \dots , \quad (7)$$

so that E_s represents the constant-signal-energy-per-channel signaling element or baud. Indeed, since the simulation is to be performed digitally, replace (7) by

$$\sum_{k=Ni-1}^{Ni} |\bar{s}(k\Delta T_s)|^2 = E_s ; \quad i = 1, 2, \dots , \quad (8)$$

where N is the number of samples per baud and is under user control. Typical choices

TABLE 3-1
DESCRIPTION OF SOME TYPICAL MODULATION CAPABILITIES
AVAILABLE IN INTERACTIVE COMMUNICATION SIMULATOR

MODULATION	$C_{1,i}$	$C_{2,i}$	$u_c(t)$	$u_s(t)$	REMARKS
BPSK	X_1	0	$u_0(t)$	0	$X_1 = \pm 1$
DPSK	$X_1 C_{1,i-1}$	0	$u_0(t)$	0	$X_1 = \pm 1$ $C_{1,0}$ ASSUMED KNOWN
BFSK	$e^{j\theta_i}$	$X_1 e^{j\theta_i}$	$\cos(\Delta\omega t/2) u_0(t)$	$\sin(\Delta\omega t/2) u_0(t)$	$X_1 = \pm 1$ $\Delta\omega = 2\pi k/T$ $\theta_{i-1}, d.s$ UNIFORM
OPSK	X_{2i}	X_{2i+1}	$(1/\sqrt{2}) u_0(t)$	$(1/\sqrt{2}) u_0(t)$	$X_1 = \pm 1$
MSK	X_{2i}	X_{2i+1}	$\cos(\pi/2T_s) u'_0(t)$	$\sin(\pi/2T_s) u'_0(t)$	$X_1 = \pm 1$ I/O WAVEFORMS REPEATED EVERY $2T_s$ SEC.

NOTE: $u_0(t)$ AND $u'_0(t)$ DEFINED IN TEXT

are $N=8, 16$, or 32 . The quantity $\Delta T_s = T_s/N$ is, of course, the sampling period. Observe that (7) is indeed satisfied for each of the entries in Table 3-1.

The communications simulator then generates the lowpass I/Q samples $\{s_c(k\Delta T_s)\}$ and $\{s_s(k\Delta T_s)\}$. For example, in the case of coherent BPSK described above,

$$s_c(k\Delta T_s) = x_i \sqrt{2E_s/T_s} \cos \Theta ; (i-1)N \leq k < iN , \quad (9a)$$

while

$$s_s(k\Delta T_s) = x_i \sqrt{2E_s/T_s} \sin \Theta ; (i-1)N \leq k < iN \quad (9b)$$

corresponds to the i th successive baud interval, $i=1,2,\dots$.

Spread-Spectrum Modulation

Spread-Spectrum Modulation is a modulation technique which spreads the output spectrum of a transmitted signal over a bandwidth much greater than that required for the transmission of the underlying information signal. Spread-Spectrum modulation is used for purposes of jamming resistance, low probability of intercept, and multiple access in modern digital communications systems.

By far the two most popular types of spread-spectrum modulation for digital communications are direct sequence (DS) and frequency hopping (FH). The DS technique causes changes in the phase of the transmitted signal at a chip rate $f_c = 1/T_c$ which is typically much faster than the symbol rate $f_s = 1/T_s$. The factor T_s/T_c or f_c/f_s is usually referred to as the "processing gain" of the system and is denoted as G . When normalized at the baud or symbol rate, f_c becomes the processing gain of the system. The FH technique hops the center frequency of the transmitted signal over G hop slots or frequencies. In this case G is also referred to as the processing gain of the system.

The spread spectrum function is implemented as a multiplication of the transmitter output by a spreading function $\tilde{d}_1(t)$. Despreading at the receiver is accomplished by multiplying by a complementary function $\tilde{d}_2(t)$. These two functions have the property that

$$\int_0^{T_s} \tilde{d}_1(t) \tilde{d}_2(t) dt = 1 . \quad (10)$$

For the direct sequence system,

$$\tilde{d}_1(t) = \sum_{j=0}^{\infty} p(j) U_{-1}(t - T_c - t_j) \quad (11a)$$

and

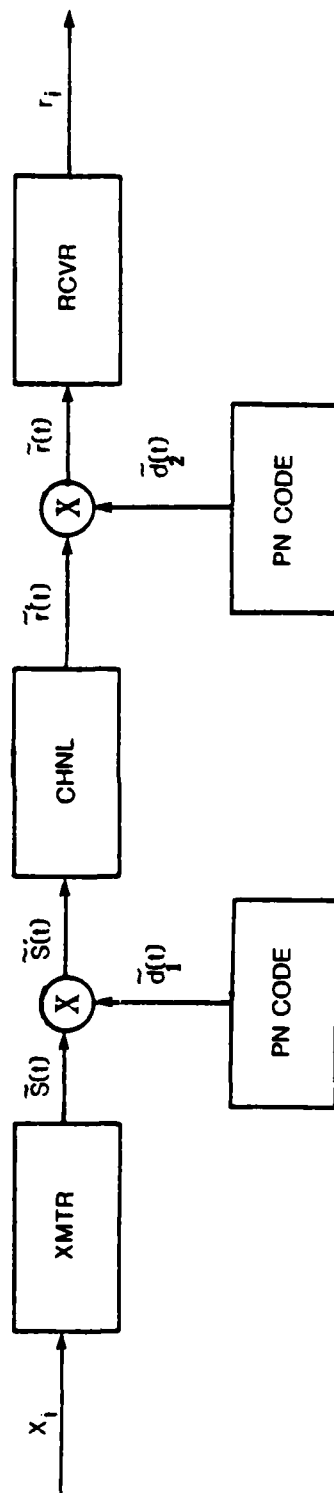
$$\tilde{d}_2(t) = \sum_{j=0}^{\infty} -p(j) U_{-1}(t - T_c - t_j) , \quad (11b)$$

where $p(j)$ represents a pseudonoise sequence of $+1, -1$. The term $p(j)$ is generated by a linear feedback shift register sequence (LFSR), the precise nature of which is under user control. The spreading function thus applies a biphase modulation at the chip rate. The direct-sequence system is illustrated in Figure 3-4.

In the FH system,

$$\tilde{d}_1(t) = \sum_{j=0}^{\infty} U_{-1}(t - t_j - T_H) e^{-j\omega_c t} \quad (12a)$$

and



DIRECT-SEQUENCE
SPREAD-SPECTRUM
SYSTEM

FIGURE 3-4

$$\tilde{d}_2(t) = \sum_{j=0}^{\infty} U_{-1}(t - t_j - T_H) e^{+j\omega_j t} . \quad (12b)$$

Here ω_j is a pseudorandom frequency which is chosen at the start of each hop interval t_j and lasts for the duration of the hop, T_H .

The processing gain G and the hop time T_H are under user control. A block diagram of the frequency-hopped system is shown in Figure 3-5.

Because of array-processor memory limitations of 32K, the spread-spectrum processing gain is limited to $G = 512$.

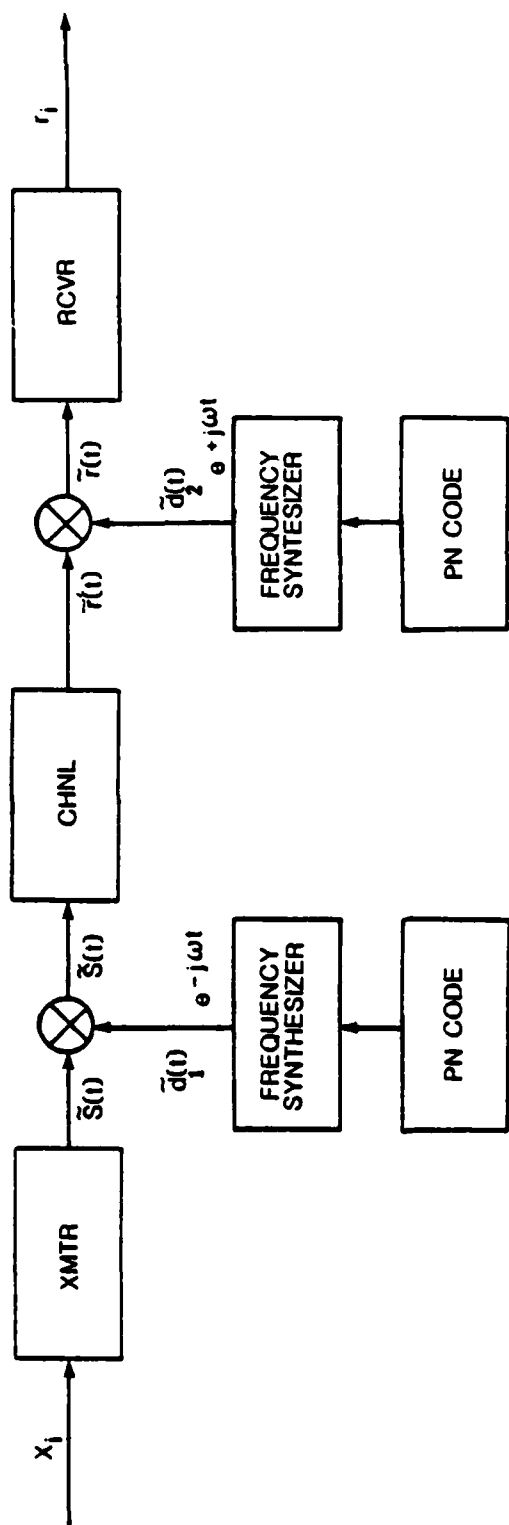
Finally, one last function provided in the modulator/transmitter module needs to be mentioned. In particular, any narrowband filtering or nonlinear amplifier distortion effects will be simulated as part of the modulator/transmitter. For example, if $h(t)$ represents the impulse response of a narrowband or bandpass filter, it can be expressed in the form

$$h(t) = 2\text{Re}\{\tilde{h}(t)e^{j2\pi f_c t}\} , \quad (13)$$

where $\tilde{h}(t) = h_c(t) - jh_s(t)$ represents the baseband equivalent impulse response. Here $h_c(t)$ and $h_s(t)$ are real lowpass functions representing inphase and quadrature filtering effects, respectively. Let $\tilde{s}_0(t)$ represent the complex envelope at the output of this filter, excited by $\tilde{S}(t)$ at its input. That is, $\tilde{s}_0(t) = \tilde{s}(t) * \tilde{h}(t)$ where $*$ represents convolution. Expressing $\tilde{s}_0(t)$ in terms of its inphase and quadrature components according to

$$\tilde{s}_0(t) = s_{0c}(t) - js_{0s}(t) , \quad (14)$$

produces



FREQUENCY-HOPPED
SPREAD-SPECTRUM SYSTEM

FIGURE 3-5

$$s_{0c}(t) = s_c(t) * h_c(t) - s_s(t) * h_s(t) , \quad (15a)$$

and

$$s_{0s}(t) = s_c(t) * h_s(t) + s_s(t) * h_c(t) , \quad (15b)$$

which is illustrated more conveniently in Figure 3-8. A variety of filtering options are available to the user in the ICS. These filters are implemented as digital filters and include such options as linear-phase finite impulse response (FIR) filters and a variety of infinite impulse response (IIR) filters such as maximally flat Butterworth response and Chebyshev equiripple response filters. Filter parameters such as cutoff frequency (normalized to the baud rate), roll-off characteristics, passband phase, and amplitude characteristics are chosen by the user from an appropriately formatted menu list. While it would be possible to provide various nonlinear saturation effects associated with the RF power amplifier in the modulator/transmitter module, such a capability is not present in the ICS at this time.

Channel

In choosing a channel model for the ICS, it was important to provide a fairly general approach which would encompass the entire range of potential applications. A useful channel model felt to satisfy this requirement consists of a three-component fading-multipath channel plus additive noise. In particular, the fading-multipath channel is assumed to consist of a single direct path, a specular multipath component, and a diffuse multipath component, as illustrated in Figure 3-7. Here $\tilde{s}(t)$ represents the complex envelope of the transmitted signal component, which is assumed to be expressible in terms of its inphase and quadrature components according to $\tilde{s}(t) = s_c(t) - js_s(t)$. The quantities ζ_s and ζ_d in Figure 3-7 represent, respectively, the specular and diffuse signal energies normalized to the energy in the direct path so that energies in the three components sum to 1.0. These parameters are under direct user control. In addition, the user is allowed to adjust the relative phase ϕ of the specular multipath component relative to the direct path. Other parameter choices

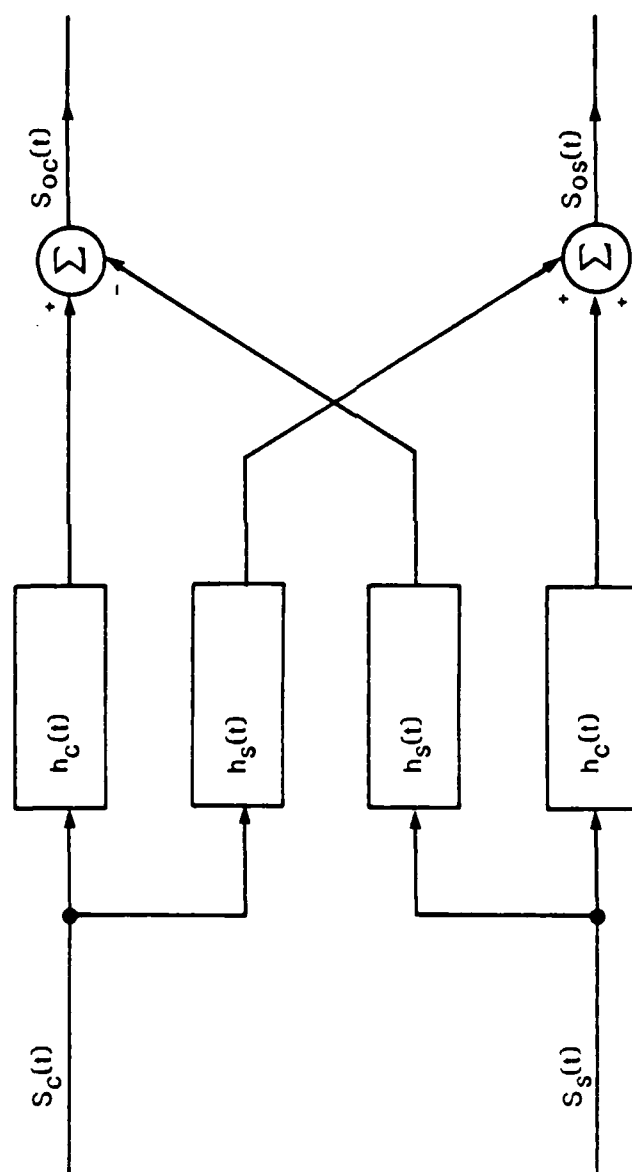


FIGURE 3-6
GENERAL NARROWBAND-FILTERING OPERATION

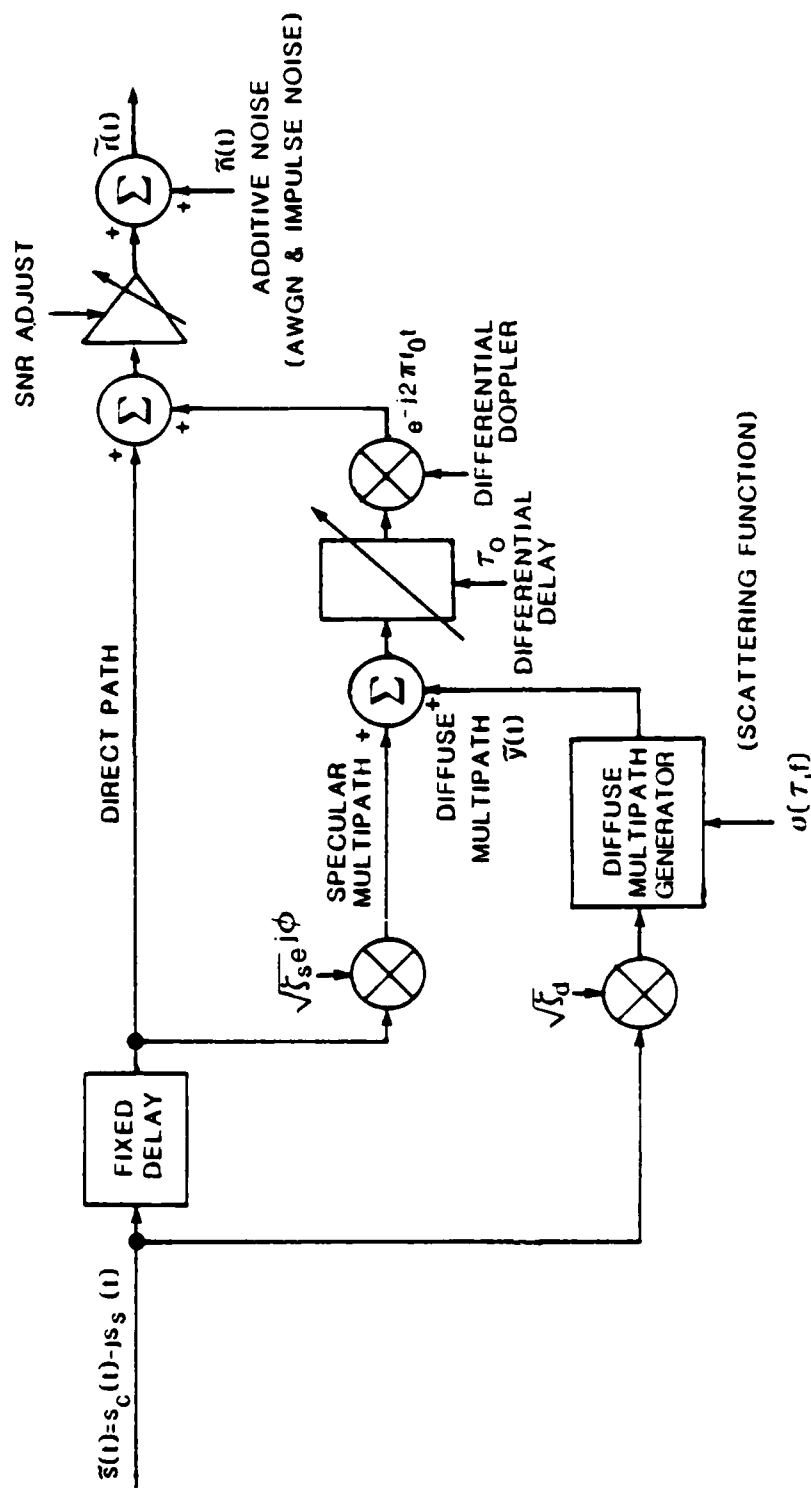


FIGURE 3-7
OVERALL BLOCK DIAGRAM OF CHANNEL MODEL

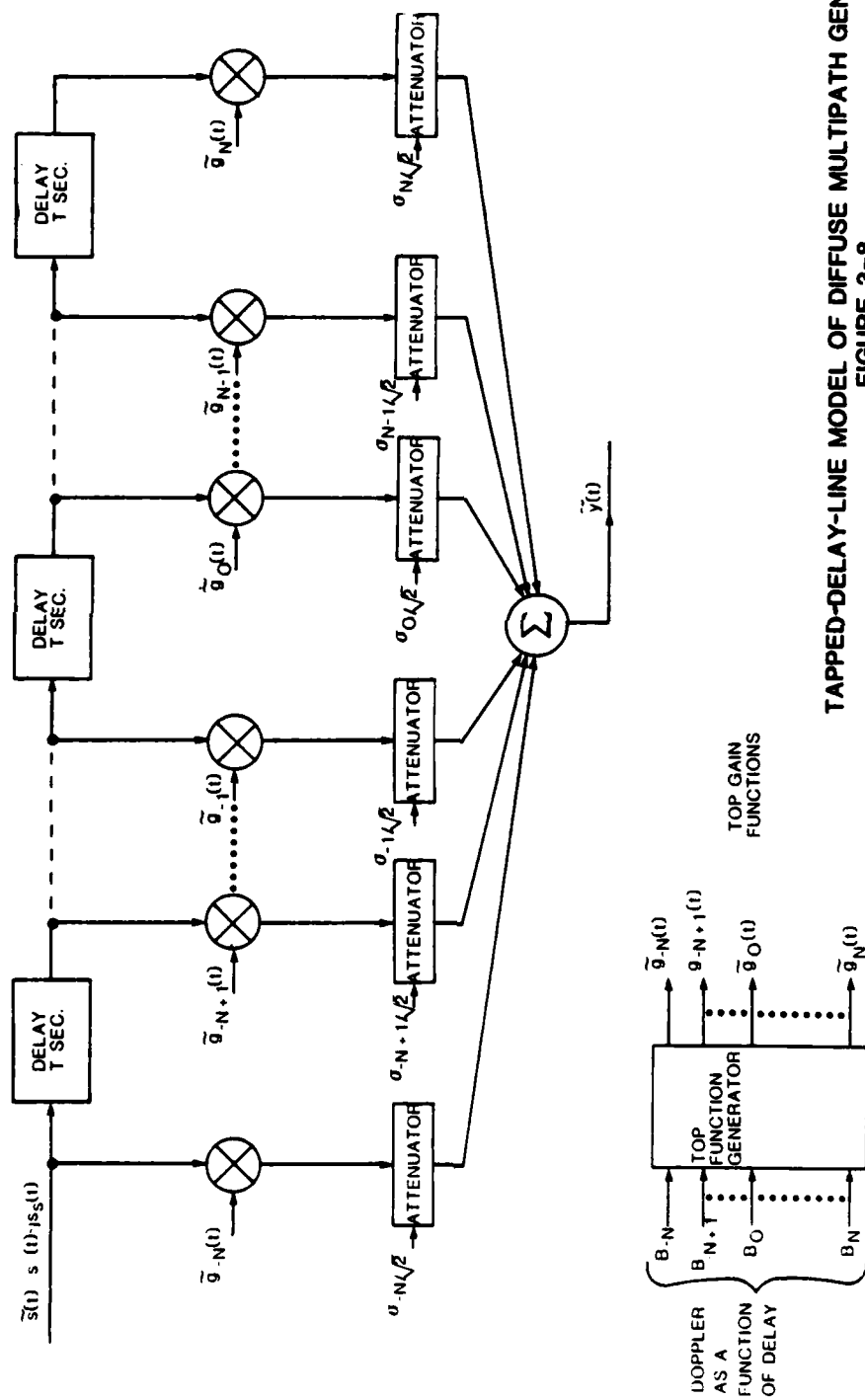
include the nominal differential delay τ_0 and the differential doppler f_0 associated with the two multipath components. The channel model assumes the same nominal differential delay and differential doppler for the two multipath components. The delay and doppler spreads about these nominal values will be determined by choice of the channel scattering-function $\sigma(\tau, f)$ associated with the diffuse-multipath component. In this context, the quantity $\sigma(\tau, f)$ represents the energy associated with delays in the range $(\tau, \tau + d\tau)$ and doppler shifts in the range $(f, f + df)$.

The diffuse multipath component has been implemented by means of the tapped delay-line model illustrated in Figure 3-8. This is a fairly general model for fading dispersive channels under the so-called wide-sense stationary uncorrelated scattering (WSSUS) assumption (cf. [5], [6]). The delay T in Figure 3-8 is assumed to be equal to the Nyquist interval $T=1/W$, where W is the transmitted signal bandwidth. In practice, a sampling interval somewhat shorter than this may be desirable in some applications.

The quantities $\tilde{g}_i(t)$, $i=0, \pm 1, \pm 2, \dots, \pm N$, in Figure 3-8 represent tap-gain functions which are modeled as mutually independent complex zero-mean wide-sense stationary Gaussian processes. In the communications simulator, all the tap-gain functions are restricted to possess second-order Butterworth power spectral densities of the form

$$S_i(f) = \frac{2}{\pi} \frac{B_i^3}{B_i^4 + f^4} ; \quad i = 0, \pm 1, \dots, \pm N, \quad (16)$$

where B_i represents the 3-dB bandwidth in Hz. Specification of the 3-dB bandwidths B_i string, then, effectively determines the doppler as a function of delay profile as illustrated in Figure 3-8. Similarly, specification of the multiplying factors σ_i , $i = 0, \pm 1, \pm 2, \dots, \pm N$ determines the delay profile. The parameters N , B_i , and σ_i , $i = 0, \pm 1, \pm 2, \dots, \pm N$ are specified by the user. The tap gain functions are then obtained as the output of appropriately defined digital filters excited by white Gaussian noise inputs. While other, more general doppler-as-a-function-of-delay profiles are easily



TAPPED-DELAY-LINE MODEL OF DIFFUSE MULTIPATH GENERATOR
FIGURE 3-8

incorporated into the ICS, the approach described here is expected to prove adequate for the intended applications.

Note that the channel model as illustrated in Figure 3-8 includes provision for an additive noise component. An adequate model for a wide range of applications, including ELF/VLF and HF, can be represented as the linear combination of a low-density shot noise process and additive white Gaussian noise (AWGN). That is, the complex envelope $\tilde{n}(t)$ of the additive bandpass noise can be expressed as

$$\tilde{n} = \tilde{y}(t) + \tilde{w}(t) . \quad (17)$$

Here $\tilde{y}(t)$ represents a complex low-density shot noise process while $\tilde{w}(t)$ represents complex AWGN with double-sided noise spectral density $N_0/2$ watts/Hz. Low-density means that the interarrival times of the impulses are relatively long compared to typical impulse durations. The low-density shot noise component $\tilde{y}(t)$ accounts for the relatively infrequent high-level and time-resolved impulse noise hits due to atmospheric discharge phenomenon or various contributions due to man-made noise. The Gaussian noise component $\tilde{w}(t)$, on the other hand, is due to a combination of front-end noise and the large number of low-level and overlapping impulse hits. The latter contribution is expected to exhibit Gaussian behavior from central limit theorem considerations.

The low-density shot noise component can be modeled as the output of a linear time-invariant filter excited by an amplitude-modulated impulse train or a point process at its input. The filter generating the complex low-density shot process $\tilde{y}(t)$ is assumed to possess impulse response $\tilde{h}(t)$, or, equivalently, system transfer function $H(s)$. A block diagram of the channel-noise model is illustrated in Figure 3-9. The impulse response $\tilde{h}(t)$ will in general be complex, possessing an inphase component $h_c(t)$ and a quadrature component $h_s(t)$. As indicated in Figure 3-9, the input to the linear filter generating the low-density shot noise component $\tilde{y}(t)$ is $\tilde{u}(t)$, assumed to be of the form

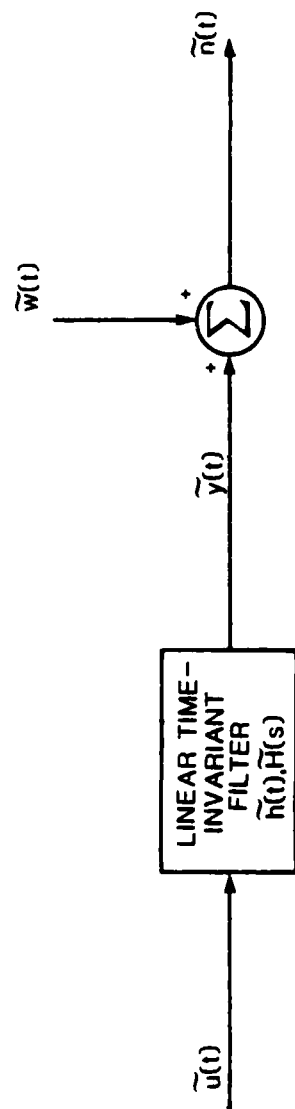


ILLUSTRATION OF IMPULSIVE CHANNEL NOISE MODEL

FIGURE 3-9

$$\tilde{u}(t) = \sum_{i=0}^{M(t)} \tilde{u}_i \delta(t-t_i) . \quad (18)$$

Here $\{\tilde{u}_i\}$ is a sequence of complex independent and identically distributed (i.i.d.) random variables which can be expressed in terms of I/Q components according to

$$\tilde{u}_i = u_{ci} - ju_{si} ; \quad i=0,1,\dots . \quad (19)$$

The process $\{N(t), t \geq 0\}$ in (18) is a point process, in particular a counting process, whose event times are described by the sequence $\{t_i\}$. More specifically, $\{N(t), t \geq 0\}$ undergoes a unit step at each of the event times $\{t_i\}$. The Poisson process is a good example of a counting process. The communications simulator possesses the ability to simulate stationary renewal counting processes possessing Gamma-distributed interarrival times with p.d.f.

$$f(x) = \frac{x^{\nu-1}}{\Gamma(\nu)\beta^\nu} \exp\{-x/\beta\} ; \quad x \geq 0 , \quad (20)$$

where $\nu \geq 1$, and the parameter β is defined according to $\beta = 1/\nu\lambda$ with $\lambda \geq 0$ fixed. The quantity λ represents the average number of events per unit time. This quantity is specified by the user in units normalized to the baud duration, T_b seconds. Specification of the two parameters ν and λ , then, completely defines the point process $\{N(t), t \geq 0\}$.

For example, if $\nu=1$, then

$$f(x) = \lambda e^{-\lambda x} ; \quad x \geq 0 . \quad (21)$$

This is the exponential distribution. In this case, $\{N(t), t \geq 0\}$ is Poisson, with λ the average number of hits per unit time. Another special case is obtained by letting $\nu \rightarrow \infty$ while holding λ fixed. The result is

$$f(x) = \delta(x-1/\lambda) , \quad (22)$$

where $\delta(\cdot)$ is the delta function. In this case, the impulses are periodic with rate λ per unit time. By introduction of the class of driving point processes described here, a parameterized impulsive noise model is provided which includes these two extremes as special cases. The class of point processes $N(t), t \geq 0$, and hence $\tilde{u}(t)$, is easily generated on a digital machine.

It is assumed that the I/Q components $\{u_{ci}\}$ and $\{u_{si}\}$ can be described by

$$u_{ci} = R_i \cos \Theta_i , \quad (23a)$$

and

$$u_{si} = R_i \sin \Theta_i , \quad (23b)$$

with $\{R_i\}$ an i.i.d. sequence possessing common probability density function (p.d.f.) $f_R(\cdot)$ while $\{\Theta_i\}$ is likewise an i.i.d. sequence uniformly distributed over $[-\pi, \pi]$. Related work, [7] - [8], on modeling impulse noise phenomenon indicates that an appropriate choice of p.d.f. $f_R(\cdot)$ is the lognormal distribution

$$f_R(R) = \frac{1}{\sqrt{2\pi}\sigma R} \exp\left\{-\frac{(\ln R - \mu)^2}{2\sigma^2}\right\} ; \quad R \geq 0 , \quad (24)$$

where μ and σ^2 are the mean and variance, respectively, of a Gaussian variate g for which $r = e^g$. Another useful choice is the power-Rayleigh distribution defined by

$$f_R(R) = \frac{\alpha}{R_0^\alpha} R^{\alpha-1} \exp\{-(R/R_0)^\alpha\} ; \quad R \geq 0 , \quad (25)$$

where the characteristic exponent α lies in the range $0 < \alpha \leq 2$, and R_0 is a scale

parameter to be specified. The sequence $\{\tilde{u}_i\}$ is thus easily generated digitally. Some guidance on choosing parameter values to match observed impulse noise characteristics is provided in [7] - [8]. These studies indicate that values of α between 0.2 and 0.7 for the power-Rayleigh case allow the impulse statistics to match observed impulse phenomena.

The low-density shot noise process $\tilde{y}(t)$ appearing at the output of the linear filter in response to $\tilde{u}(t)$ at its input can be expressed as

$$\tilde{y}(t) = y_c(t) - jy_s(t), \quad (26)$$

where the I/Q components $y_c(t)$ and $y_s(t)$ can be generated according to

$$y_c(t) = \sum_{i=0}^{M(t)} R_i * [h_c(t-t_i)\cos\Theta_i - h_s(t-t_i)\sin\Theta_i], \quad (27a)$$

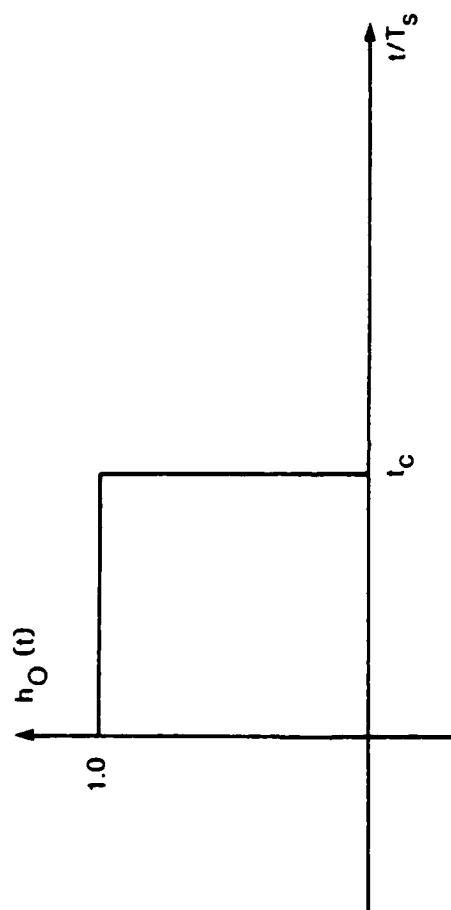
and

$$y_s(t) = \sum_{i=0}^{M(t)} R_i * [h_c(t-t_i)\cos\Theta_i + h_s(t-t_i)\sin\Theta_i]. \quad (27b)$$

Here $h_c(t)$ and $h_s(t)$ are, as described previously, the I/Q components of the complex impulse response function $\tilde{h}(t)$. The terms $h_c(t)$ and $h_s(t)$ are both assumed to be equal to the same impulse response functions $h_0(t)$. Several choices for $h_0(t)$ are possible. For example, the simplest choice is the pulse waveform of duration t_c seconds normalized to the baud duration as illustrated in Figure 3-10. The corresponding impulse response is

$$h_0(t) = u_{-1}(t) - u_{-1}(t-t_c), \quad (28)$$

where $u_{-1}(t)$ is the unit step function



SIMPLE PULSE WAVEFORM MODEL FOR LOW-DENSITY SHOT NOISE

FIGURE 3-10

$$u_{-1}(t) = \begin{cases} 1 & ; t \geq 0 \\ 0 & ; t < 0 \end{cases} \quad (29)$$

The duration t_c is specified by the user such that $t_c \ll 1/\lambda$, so that the impulse process is truly low density. Other choices of pulse waveform available are illustrated in Table 3-2.

Finally, the AWGN component which is added to the low-density shot noise component to complete the channel noise model is described in terms of its I/Q components according to

$$\tilde{w}(t) = w_c(t) - jw_s(t) \quad (30)$$

The components $w_c(t)$ and $w_s(t)$ will be modeled as mutually independent zero-mean white Gaussian noise processes, each possessing double-sided noise spectral density $N_0/2$ watts/Hz. These components are easily simulated. Gaussian variates are accurately generated in the ICS by generating a Rayleigh random variable R and a uniform variate θ on the interval $[-\pi, \pi]$. The transformations $g_1 = R \sin \theta$ and $g_2 = R \cos \theta$ then produce two independent Gaussian variates. This procedure is much more accurate than approximating the Gaussian distribution with a sum of uniform variates.

Generic-Channel Design

The ICS provides the ability to design high-frequency (HF), tropospheric-scatter (TROPO), or line-of-sight (LOS) generic transmission channels in terms of parameters which are physically meaningful for each channel. A translation program accepts generic link information for each of the three links and generates the appropriate fundamental parameters for the ICS fading/impulsive channel models. Each of the channel design programs is described.

TABLE 3-2
PULSE WAVEFORMS AVAILABLE IN IMPULSE
NOISE GENERATOR

PULSE TYPE	IMPULSE RESPONSE $h_o(t)$	SYSTEM TRANSFER FUNCTION $H_o(s)$
UNIPOLAR PULSE	$u_{-1}(t) - u_{-1}(t - t_c)$	$\frac{1 - e^{-st_c}}{s}$
EXPONENTIAL PULSE	$e^{-at} u_{-1}(t)$	$\frac{1}{s + a}$
DECAYING SINE	$\frac{1}{\omega_c} e^{-at} \sin \omega_c t u_{-1}(t)$	$\frac{1}{(s + a)^2 + \omega_c^2}$
DECAYING COSINE	$e^{-at} \cos \omega_c t u_{-1}(t)$	$\frac{s + a}{(s + a)^2 + \omega_c^2}$

The LOS channel is usually modeled as a direct path with additive noise. The pertinent parameter to be determined is the received SNR, which is calculated as:

$$E_s/N_o = P_{T_{dBm}} - LOSS_{dB} - N_{dBm} - 10LOG B_R, \quad (31)$$

where the path loss $LOSS_{dB}$ can be expressed as

$$LOSS_{dB} = -G_{T_{dB}} - G_{R_{dB}} + 36.6 + 20LOG(F_{MHz}) + 20LOG(D_{MILES}). \quad (32)$$

These equations depend on the following parameters:

Transmitted Power/dBm	- $P_{T_{dBm}}$
Path Distance (Stature Miles)	- D_{MILES}
Receiver Noise Floor (dBm)	- N_{dBm}
Transmitter Antenna Gain (dB)	- $G_{T_{dB}}$
Receiver Antenna Gain (dB)	- $G_{R_{dB}}$
Baud Rate	- B_R (bits/sec)
Frequency (MHz)	- F_{MHz}

The E_s/N_o thus calculated may then be entered in either the VALIDATION or SIMULATION modes.

The TROPO interface program is based on a model developed by the Defense Communication Engineering Center (DCEC) and is based on the tapped-delay-line model described in [19]. The first pass of the program computes a power-versus-delay profile after the user inputs parameters related to transmission distance, antenna beamwidths, and transmission frequency.

The resulting powers at discrete delay intervals become the tap variances for the ICS fading dispersive channel. The second pass of the TROPO design routine calculates received average E_s/N_o .

In addition to the parameters needed to compute the tap gain variances, the user must also input:

Transmitted Power (dBm)	- P_{TdBm}
Receiver Noise Floor (dBm)	- N_{dBm}
Transmitter Antenna Gain (dB)	- G_{TdB}
Receiver Antenna Gain (dB)	- G_{RdB}
Frequency (MHz)	- F_{MHz}
$F(\theta d)$ attenuation function (dB)	- $F(\theta d)$ see [3]
Atmospheric absorption factor [4]	- A_a .

The attenuation function $F(\theta d)$ determines the path loss as a function of the product of the scatter angle θ and great circle path distance d . The atmospheric absorption factor A_a is a function of frequency and path length. Both of these quantities are usually read from tables or graphs.

E_s/N_o is then calculated as follows:

$$E_s/N_o = P_{TdBm} - Loss_{dBm} - N_{dBm} - 10\log BR, \quad (33)$$

where

$$Loss_{dBm} = 30\log F_{mhz} - 20\log_{dBm} + F(d) + Aa - G_T G_r. \quad (34)$$

The HF channel is typically modeled as a one-path, two-path, or three-path dispersive channel with additive impulsive noise. The parameter translation program employed in ICS is based on the DICEF *HF MEDIA SIMULATOR* developed by SIGNATRON, Inc. [20]. The user specifies the following information:

o Baud rate	- BR bits/sec
o Number of paths	- 1, 2, or 3
o Delay per path	- $D_{\mu s}$

- o RMS doppler spread per path
- o Power per path
- o Mean doppler shift

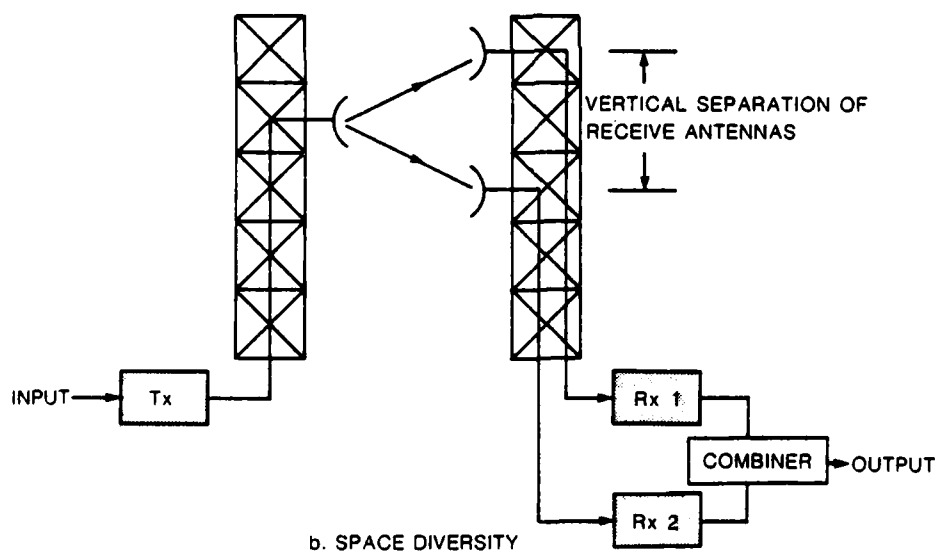
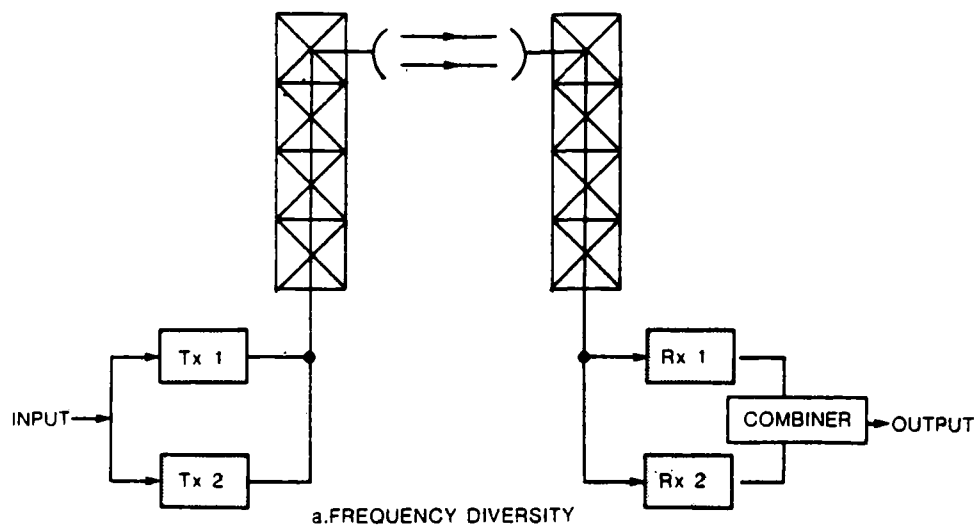
For the impulse noise (assumed rectangular) the user specifies:

- o number of impulses per minute
- o width of each pulse
- o periodic or random arrivals
- o power in impulsive component.

Explicit Diversity

The effect of fading is often minimized by the use of space-diversity or frequency-diversity techniques as illustrated in Figure 3-11. These techniques are based on the hypothesis that simultaneous fading on both radio transmission channels is unlikely. In a frequency-diversity configuration, the same information is fed to two (or more) transmitters, TX 1 and TX 2. A wide frequency separation ensures less correlation between the fades of the individual signals; therefore, improved performance may be realized. In the space-diversity system, the same frequency band is used in transmission; however, the two receiver antennas are separated vertically. This ensures that the signals travel through different transmission paths and are, thus, not likely to be affected by fading in the same way. Combinations of both space diversity and frequency diversity have also been used to achieve increased performance. An explicit diversity-simulation facility has been incorporated into the ICS.

Two popular methods of combining the outputs of diversity channels, selection combining and maximal-ratio combining, have been implemented. Each method uses SNR information in a different way to combine the channel outputs. A signal-to-noise estimate is derived as:



EQUIPMENT CONFIGURATION FOR (a) FREQUENCY-DIVERSITY AND (b) SPACE-DIVERSITY SYSTEMS

FIGURE 3-11

$$SNR = \frac{\hat{r}(n)}{\hat{\sigma}^2(n)}, \quad (35)$$

where $\hat{r}(n)$ is the short-term average signal level and $\hat{\sigma}^2(n)$ is the noise variance. Both quantities are evaluated at sample time (n) . The signal level estimate is given by:

$$\hat{r}(n) = \frac{1}{n_r} \sum_{i=n-n_r+1}^n |r(i)|, \quad (36)$$

where n_r represents the number of channel output samples $r(i)$ that are to be used in the estimate. The variance estimate is coupled to the signal estimate and is given by:

$$\hat{\sigma}^2(n) = \frac{1}{n_\sigma} \sum_{i=n-n_\sigma+1}^n (|r(i)| - \hat{r}(n))^2, \quad (37)$$

where n_σ is the number of samples used in the estimate.

The selection diversity algorithm accepts data from an arbitrary channel among the D diversity channels at the start of transmission and switches to a different diversity path if the new path has the highest SNR estimate of D paths and if this SNR exceeds the SNR of the current path by an amount Δ .

Maximal-ratio combining accepts data from each of the D channels and weights the data of each channel by its corresponding SNR estimate before adding to produce a final output $r'(i)$:

$$r'(i) = \sum_{d=1}^D r_d(i) / SNR_d(i). \quad (38)$$

Electronic Countermeasure (ECM) Simulation

A variety of Electronic Countermeasure Simulation modules have been included in the ICS. The jamming waveform capabilities include multitone-modulated AM, noise-modulated AM, multitone-modulated FM, noise-modulated FM, continuous wave (CW) interference, linear-swept CW, log-swept CW, and a versatile frequency-agile pulsed-jamming facility. A general expression for the complex envelope of the jamming waveform is given in terms of inphase $I_c(t)$ and quadrature $I_s(t)$ components:

$\tilde{I}(t) = I_c(t) - jI_s(t)$ is developed as:

$$I_c(t) = A[1 + M \sum_i b_i \cos(2\pi f_{AMi} t)] \cdot \cos\left[\frac{(\Delta W + W_{step})}{e^{i_{(0,1)} t}} t\right] + \frac{\Delta f}{f_M} \sum_j (a_j \sin(2\pi f_{FMj} t) + \theta), \quad (39a)$$

and

$$I_s(t) = A[1 + M \sum_i b_i \cos(2\pi f_{AMi} t)] \cdot \sin\left[\frac{(\Delta W + W_{step})}{e^{i_{(0,1)} t}} t\right] + \frac{\Delta f}{f_M} \sum_j (a_j \sin(2\pi f_{FMj} t) + \theta). \quad (39b)$$

- A - total normalized gain
- M - modulation index of AM signal
- b_i - weighting term
- f_{AMi} - AM tone frequency
- ΔW - offset from carrier frequency
- W_{step} - sweep step
- $i_{(0,1)}$ - on/off, logarithmic/linear sweep (1/0)
- $\Delta f/f_m$ - modulation index FM wave

- a_j - weighting term
- f_{FMj} - FM tone frequency
- θ - phase offset

The desired jamming scenarios can be derived from the above equations by substituting zeros for the unrelated variables.

For the simple case of CW interference, the following variables can be substituted with the constant zero, M , b_i , f_{AMi} , W_{step} , L , f/f_m , A_j , and f_{FMj} . This leaves:

$$I_c(t) = A \cos(\Delta W t + \theta), \quad (40a)$$

and

$$I_s(t) = A \sin(\Delta W t + \theta), \quad (40b)$$

which are the formulas for CW interference. For swept CW terms, W_{step} and $i_{(0,1)}$ remain in the original equations to produce:

$$I_c(t) = A \cos\left[\frac{(\Delta W + W_{step})}{e^{i_{(0,1)} t}} t + \theta\right] \quad (41a)$$

and

$$I_s(t) = A \sin\left[\frac{(\Delta W + W_{step})}{e^{i_{(0,1)} t}} t + \theta\right]. \quad (41b)$$

For amplitude-modulated interference, the following variables are substituted with zeros: W_{step} , $i_{(0,1)}$, f/f_m , A_j , and f_{FMj} . This breaks down (39) into the AM standard form:

$$I_c(t) = A[1 + M \sum_i b_i \cos(2\pi f_{AMi} t)] \cos(\Delta W t + \theta) \quad (42a)$$

and

$$I_s(t) = A[1 + M \sum_i b_i \cos(2\pi f_{AMi} t)] \sin(\Delta W t + \theta) \quad (42b)$$

For the case of AM noise, the term $\sum_i b_i \cos(2\pi f_{AMi} t)$ is substituted with a Gaussian random variable.

FM interference can be simulated when M , b_i , f_{AMi} , W_{step} , and $\alpha(0,1)$ are substituted with zeros. The equations:

$$I_c(t) = A \cos[\Delta W t + \frac{\Delta f}{f_m} \sum_j a_j \sin(2\pi f_{FMj} t) + \theta] \quad (43a)$$

and

$$I_s(t) = A \sin[\Delta W t + \frac{\Delta f}{f_m} \sum_j a_j \sin(2\pi f_{FMj} t) + \theta] \quad (43b)$$

result.

For noise-modulated FM, the quantity $\frac{\Delta f}{f_m} \sum_j a_j \sin(2\pi f_{FMj} t) + \theta$ is replaced by a Gaussian process.

The pulsed frequency-agile jammer is constructed by replacing A in (39a and b) with

$$A(t) = \sum_{k=0}^{N(t)} a_k [U_{-1}(t - t_k) - U_{-1}(t - T_B - t_k)] e^{-jW_k t} \quad (44)$$

Here $N(t)$ is the same counting process employed in the impulsive noise channel with an additional provision for perfectly periodic arrivals. The quantity W_k represents a random frequency for the k th jamming burst and is drawn from a distribution under user control. The random jamming amplitude a_k is chosen from either a power-Rayleigh or uniform distribution.

Although only these five distinct choices of jamming waveforms are available as generic jamming options, the basic jamming module allows any combination of the above modulations to be employed.

Demodulator/Receiver

The main purpose of the demodulator/receiver function module is to perform data demodulation of the channel output $r(t)$. That is, this function module produces the sequence $\{\underline{r}_i\}$ indicated in Figure 3-2, which is applied as input to the channel decoder. The demodulator/receiver produces an output once per baud interval. The term $\{\underline{r}_i\}$ has been purposefully allowed to be a vector sequence to accommodate various soft-decision decoding schemes. For example, in an M -ary system, the receiver might well produce an ordered list of the l largest outputs ($l < M$) during successive channel signaling intervals. This is the so-called list-of- l quantization scheme [9] which provides a reasonably efficient and computationally effective means for implementing soft-decision decoding in a variety of M -ary signaling situations.

The demodulator/receiver module contains a predetection front end which precedes the data demodulator and/or other portions of this module. This predetection section, illustrated in Figure 3-12, consists of a zero-memory nonlinear (ZMNL) device sandwiched between two narrowband filters. The filters are implemented as described previously for the output filtering provided in the modulator/transmitted module. A ZMNL processing capability has been provided in the ICS to allow simulation of intentional nonlinear processing, such as might be used to mitigate against the effects of

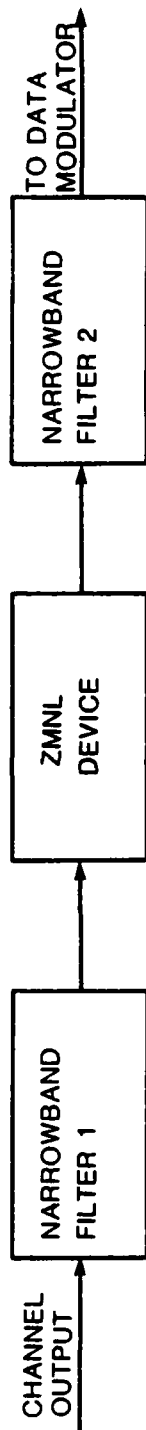


FIGURE 3-12
PREDETECTION PORTION OF RECEIVER

impulse noise.

The various ZMNL processing schemes are generally of the limiting variety and have been implemented in the ICS as illustrated in Figure 3-13. Here the envelope $R_0(t) = \sqrt{s_{0c}^2(t) + s_{0s}^2(t)}$ of the complex envelope $\tilde{s}_0(t)$ is formed and passed through an instantaneous or zero-memory nonlinear (ZMNL) device with input/output characteristic $g[R_0(t)]$ depending only upon the instantaneous envelope $R_0(t)$. Forming the product

$$\tilde{s}'_0(t) = g[R_0(t)]\tilde{s}_0(t) \quad (46)$$

allows synthesis of a variety of nonlinearities of the limiting or saturation type by appropriate choice of the function $g(R)$. These nonlinearities are all characterized by the fact that they depend upon the envelope and not upon the inphase and quadrature components individually. For example, the choice $g(R) = 1/R$ can easily be shown to correspond to an ideal bandpass limiter. The choice

$$g(R) = \begin{cases} 1 & ; 0 \leq R \leq 1 \\ 1/R & ; R > 1 \end{cases} \quad (47)$$

on the other hand, represents a soft limiter or clipper characteristic. Butterworth-type nonlinear characteristics of various cutoffs and orders are provided under user control in an attempt to provide a realistic and yet computationally efficient model for demodulator/receiver nonlinear effects.

It should be noted that in order to perform its basic data demodulation functions, the demodulator/receiver module must perform a variety of ancillary functions. These ancillary functions include symbol synchronization and phase tracking, which have been implemented for BPSK and DPSK modulation strategies. In addition, data symbol deinterleaving is performed within the demodulator/receiver module. It is these ancillary functions which seriously complicate the design of this function module. The ICS also has the capability of patching symbol synchronization and carrier phase and/or

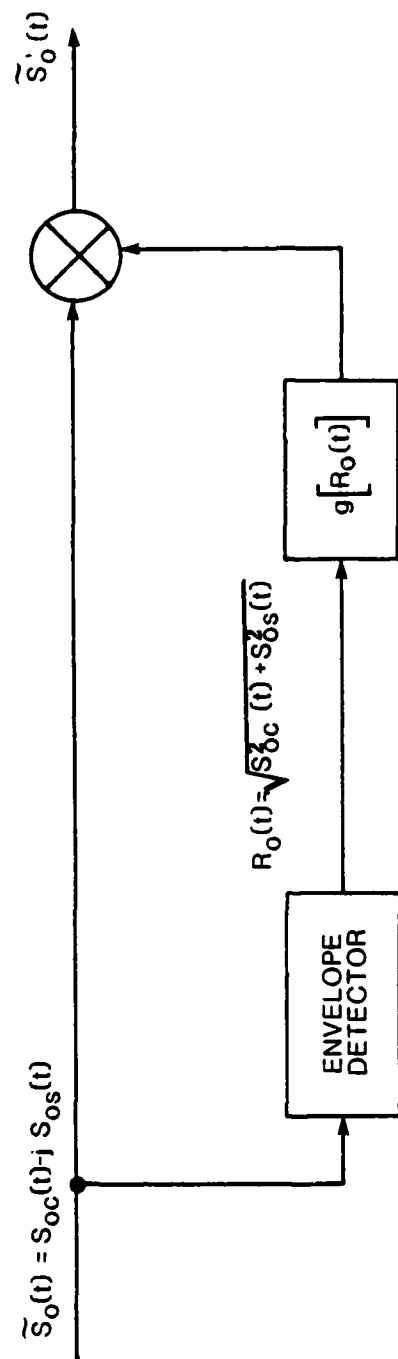


FIGURE 3-13
MODEL FOR ZERO-MEMORY NONLINEARITIES OF THE LIMITING VARIETY

frequency from transmitter to receiver. This facility is useful, for example, when one would like to isolate and eliminate the various synchronization loop tracking dynamics from consideration.

Channel Decoder

The channel decoder will accept the sequence $\{\tilde{r}_i\}$ appearing at the output of the demodulator/receiver and, through appropriate decoding algorithms, will produce the output sequence $\{\hat{a}_i\}$ delivered to the remote destination. Here several decoding options are provided depending upon whether block or convolutional codes are used. In the latter case, the Viterbi Algorithm (VA) (cf. [14], [15]) is employed for decoding arbitrary convolutional codes with constraint lengths in the range $3 \leq K \leq 9$. For block codes, several hard-decisions decoders, using essentially table-look-up techniques, have been implemented. These have included the Hamming (8,4) and the Golay (24,12) codes which are typical choices in a wide variety of applications. In addition, a fairly general soft-decision decoding capability for arbitrary block codes was provided using the Wolf algorithm [16], which resembles, in many ways, the VA. Use of this decoding capability requires explicit knowledge of the code-parity check matrix. For selected codes this information is stored in tables on the host computer. The user then specifies a code (e.g., the Golay (24,12) code), and the appropriate parity-check matrix is loaded with the decoder program.

3.3 OPERATION OF THE ICS

The overall technical design of the ICS has been described in some detail in the preceding section. In this section, the interactive use of this system is described. Careful consideration has been given to this issue to allow flexible and efficient usage. The general philosophy is that simulation should be used as an adjunct to, and not as a substitute for, analysis. The communications-simulator development has been consistent with this philosophy. While the basic approach is Monte Carlo simulation, every effort has been made to incorporate analytical and graphical techniques where and when appropriate to enhance and extend the capabilities of this system.

The ICS has been designed to be used in four distinct modes of operation. In the first, called the DESIGN mode, the user assembles a simulation model in building-block fashion from basic function modules provided as part of the system. Appropriate user protocols have been provided to guide even the casual user through the DESIGN stage. Specifically, the block diagram of a general communications system (illustrated in Figure 3-2) is displayed on the graphics CRT. Using the cross-hair curser, the user selects successive blocks, and a menu list appears describing each of the available options for each functional element. The user chooses from this list by entering the appropriate keyboard response, eventually creating a DESIGN file representing an executable simulation model of a complete end-to-end communications system. This DESIGN file is suitably named and entered into the system library. At this point, the user can either exit from the DESIGN mode or continue creating alternate executable DESIGN files. This is the most straightforward of the four usage modes and requires no further elaboration.

The second usage mode is called the VALIDATION mode. Here the user is allowed to interactively exercise a simulation model which has previously been developed in a DESIGN session. The purpose of the VALIDATION mode is to allow the user to verify a simulation model before a commitment to time-consuming Monte Carlo simulation. More specifically, the user is allowed to view time waveforms and/or frequency-domain quantities such as power spectral densities, system transfer functions, etc., at critical points in the system to be simulated. This should enable the user to verify the end-to-end behavior of the simulation model for the communication system under study or the input/output behavior of any specific sub-module. Situated at the graphics terminal, the user can by cross-hair keyboard entry view time-domain or frequency-domain quantities as if a scopeprobe was connected to the corresponding point in an actual communications system. This ability is termed the SCOPEPROBE concept. This capability of the communication simulator should not be considered as merely a procedural verification stage preceding an actual simulation. Rather, the VALIDATION mode can itself be employed as a highly flexible and useful tool in conceptual communication-system studies. That is, by employment of the VALIDATION mode, a useful and realistic appreciation of actual system behavior can be developed. Such an appreciation can often serve as a useful adjunct to detailed

analysis. For example, the effects of various filtering operations can be observed directly, degradations due to intersymbol interference or multipath can be assessed visually, etc. Some typical graphical outputs are provided in the next subsection.

In order to modify or otherwise change a simulation model, the user must leave the VALIDATION stage and re-enter the DESIGN mode. Appropriate editing capabilities have been provided to allow flexible modification of previously developed simulation models. At any rate, the user is at some point ready to commit a simulation model to extensive Monte Carlo simulation. It is at this point that the user enters the third stage, called the SIMULATION mode. In this mode, an executable DESIGN file corresponding to a simulation model must be identified together with appropriate simulation parameters. Primary interest is in the evaluation of bit-error probability P_b as a function of the quantity E_b/N_0 . Here E_b represents the signal energy per bit while $N_0/2$ represents the double-sided noise spectral density in watts/Hz. As a result, the important parameters to be specified are both an initial and a final value of E_b/N_0 for which simulation results are desired as well as an increment in E_b/N_0 . In addition, parameters such as the number of errors to be collected for each value and the maximum number of iterations must be specified. These latter quantities are necessary in order to ensure statistical validity of the results. The communications simulation will then loop through the specified range of E_b/N_0 . This is the *only* parameter for which looped operation is provided in the SIMULATION mode. For other parameters it is required that an appropriate DESIGN file representing a simulation model with specific parameter choices has been identified for subsequent execution.

In the SIMULATION mode, appropriate data-logging and bookkeeping software is provided to allow tabulation of bit-error histories and evaluation of bit-error probabilities as a function of E_b/N_0 . This software is considered an important part of the communications simulator.

The last and final mode of operation is called the ANALYSIS mode. This mode provides two basic functions. The primary function is to provide the graphical display interpretation of results obtained during a typical SIMULATION session. Appropriate display drivers and associated software have been provided to allow results to be

displayed in a format which allows easy interpretation by communication systems engineers. For example, a basic requirement is to display curves representing simulated bit-error probability P_b as a function of E_b/N_0 . In those cases where closed-form analytical expressions or bounds on P_b vs. E_b/N_0 are available, these are included as user-selected options. The user protocol in the ANALYSIS mode will provide guidance to the user concerning any such facilities available for the system configuration and/or parameter choices under study. For example, fairly tight upper bounds are available [14], [17] on P_b vs. E_b/N_0 for convolutional codes in conjunction with Viterbi decoding and employing a variety of modulation formats on the AWGN channel. Similar bounds have been derived [18] for soft-decision decoding of block codes employing the Wolf algorithm. These bounds, for the most part, neglect the effects of narrowband filtering, imperfect phase and/or symbol synchronization, intersymbol interference, etc. Nevertheless, the readily computed upper bounds are useful in providing a context or perspective in which to assess the degradations due to these effects. For this reason, the ability to compute and display these idealized performance bounds together with results obtained by simulation has been incorporated as a useful feature of the ICS.

In the course of developing the ICS, it was found to be useful to access the computational and display capabilities associated with the ANALYSIS mode on a stand-alone basis. For example, the ability to compute and plot upper bounds, and in some cases, actual theoretical performance of selected communication systems, has proven useful in a number of related system studies. As a result, the ANALYSIS mode has been designed to allow dual entry, thus providing a secondary function as a stand-alone computational and display resource. The library associated with this mode can be easily updated to provide an expanding analysis tool quite independent of the simulation capabilities associated with the overall system.

A block diagram illustrating the overall software organization of the communications simulator is presented in Figure 3-14.

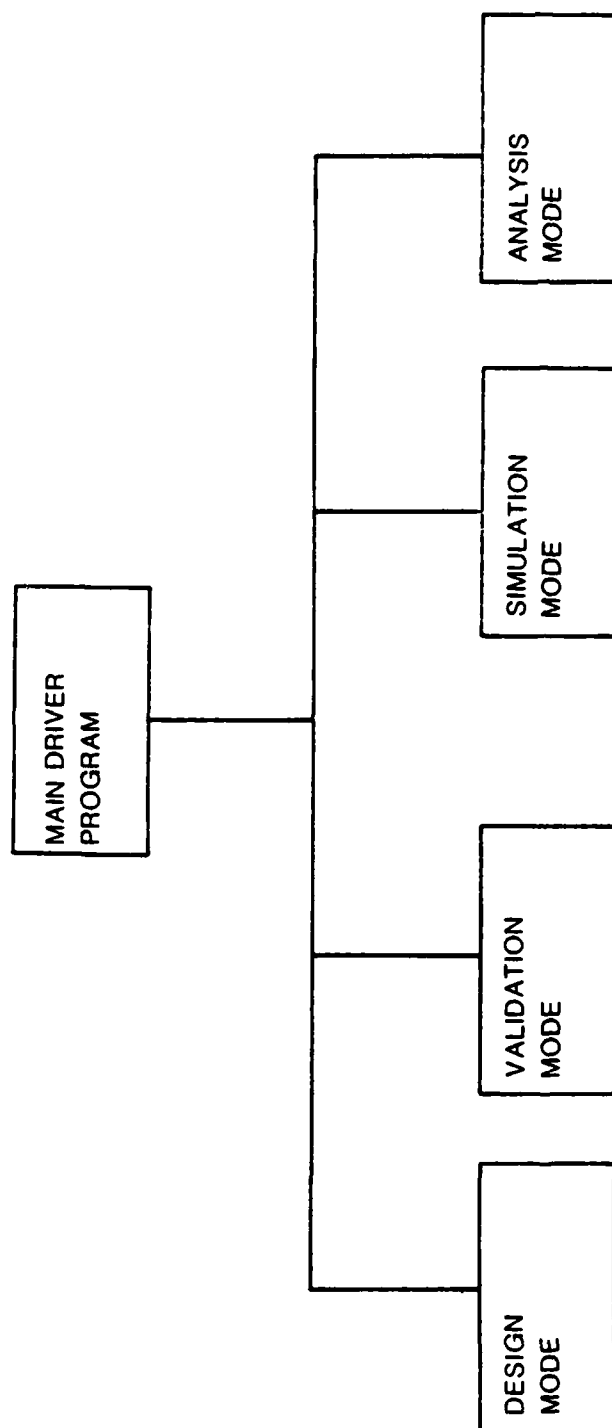


FIGURE 3-14
SOFTWARE ORGANIZATION OF INTERACTIVE
COMMUNICATION SIMULATOR

3.4 TYPICAL GRAPHICAL OUTPUT

In this section, some of the hard-copy graphical outputs obtained with the ICS are discussed. First considered are the time-domain waveforms obtainable in the VALIDATION mode. Frequency-domain graphical display and/or output capabilities are available and will be described below. Virtually any time waveform can be observed individually or in combination with other related waveforms. The user must first specify a time or viewing window consisting of a fixed number of baud. A segment of a waveform is then displayed within the viewing window, and the contents can be indexed through a waveform file either in fixed increments or continuously scrolled through a large file. The scrolling can be stopped and restarted under keyboard control.

It was found convenient to classify time-domain waveforms into five broad categories:

1. BASEBAND Waveforms
2. CHANNEL Waveforms
3. Eye Diagrams
4. Continuous I/Q Plots
5. Sampled I/Q Plots.

The BASEBAND Waveforms include all time-domain quantities which allow lowpass signal representations and are in some sense synchronous with the baud rate. This would include source outputs, source/channel encoder and decoder outputs, and matched filter outputs sampled at the baud rate. By comparison, CHANNEL waveforms include all time-domain waveforms which require complex representations, either in polar coordinates or in terms of I/Q components. Actually, the user protocol allows the user to specify either representation. Frequency-domain plotting and display capabilities are available for any CHANNEL waveforms appearing within the viewing

window and using a wide variety of possible window functions. Detailed information on the various window options is provided in the user protocol. Again, the frequency-domain representation can be displayed in either polar or rectangular (i.e. I/Q) format under user control. The eye diagram represents the continuous running sum of the "undumped" matched filter output. The continuous I/Q plot is a polar representation of sampled complex data which is available at the output of any module generating complex data. Finally, the sampled I/Q plot is a polar representation of the I versus Q matched filter output statistic.

Some typical BASEBAND signaling waveforms are illustrated in Figure 3-15 for a $K=6$, $R=1/2$ convolutionally encoded QPSK system on the AWGN channel with $E_b/N_0 = 20dB$. In this case the observation window consists of 8 baud, and a MULTIPLE WAVEFORM plotting option is used. More specifically, while the user can look at any individual BASEBAND waveform in the COMPOSE plotting and display mode, the ICS provides the capability of plotting certain logical combinations in the MULTIPLE WAVEFORM plotting option. These figures illustrate the source output and Viterbi decoder output in time alignment in the right-hand column, while the encoder output and matched filter output are in time alignment along the left-hand column. This is a logical arrangement of multiple BASEBAND waveforms and is provided to the user as a display option.

Typical CHANNEL signaling waveforms are illustrated in Figure 3-16. The waveforms in Figure 3-16 correspond to a coherent QPSK system operating on the AWGN channel at the relatively large value $E_b/N_0 = 20dB$. Here again, use is made of a MULTIPLE WAVEFORM plotting capability with the left-hand column associated with the modulator/transmitter output $s(t)$, while the right-hand column is associated with the channel output $r(t)$. Furthermore, the CHANNEL waveforms in this case are plotted in polar coordinates in terms of envelope and phase. Similar plots in rectangular coordinates using I/Q components could have been chosen under user control. Likewise, individual CHANNEL waveforms could have been selected under the COMPOSE plotting and display option. In this case, frequency-domain representations could be computed and displayed.

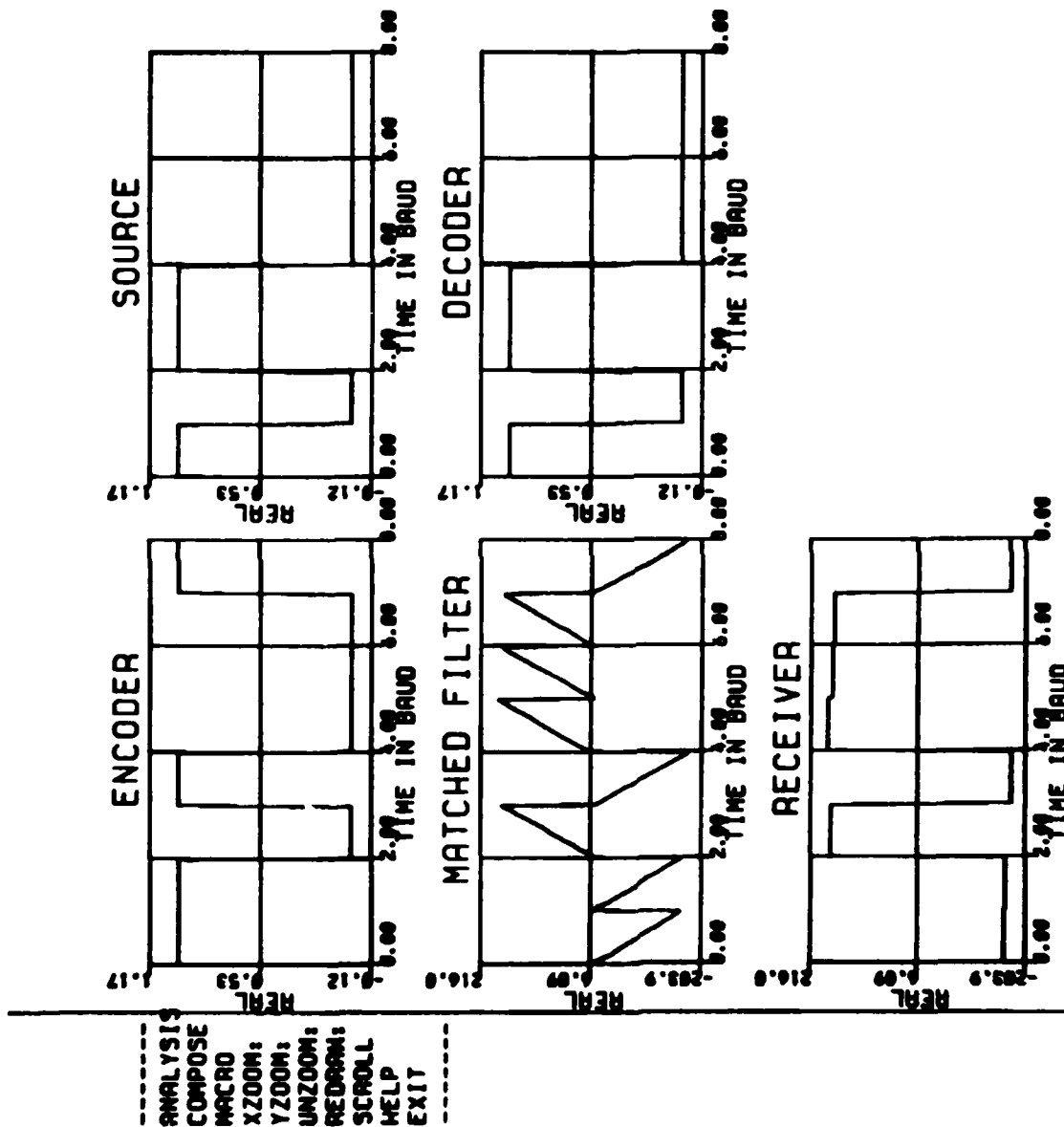


FIGURE 3-15 ICS WAVEFORM ANALYSIS

 ANALYSIS
 COMPOSE
 MACRO
 XZOOM:
 YZOOM:
 UNZOOM:
 REDRAW:
 SCROLL
 HELP
 EXIT

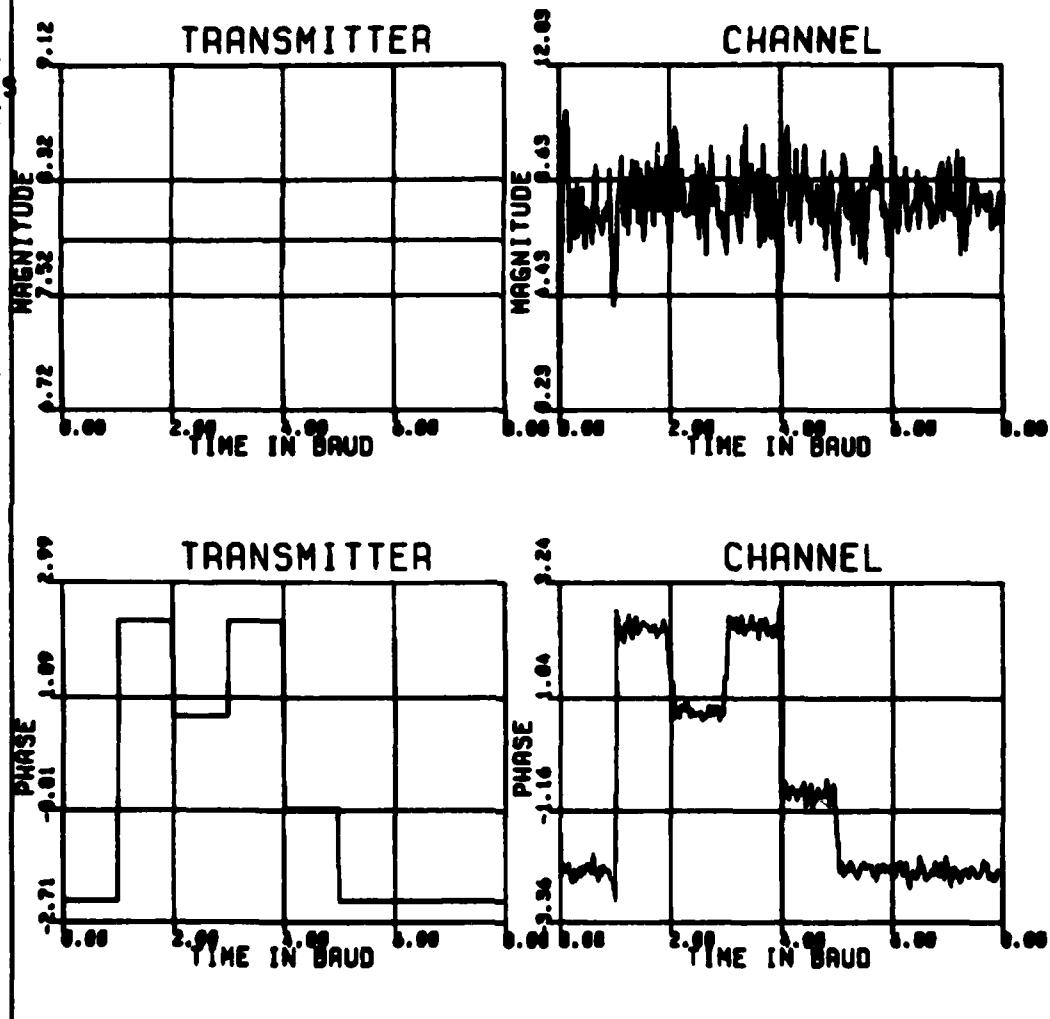


FIGURE 3-16 ICS WAVEFORM ANALYSIS

For a complete description of the remaining graphical display options, see the ICS User's Manual.

A number of additional graphical plotting and display capabilities exist in the ANALYSIS mode. For the most part, these consist of accessing, formatting, and displaying error statistics or noise amplitude probability distributions (APDs) generated in executing an appropriate DESIGN file in the SIMULATION mode. For example, a DESIGN file, representing a coherent QPSK system operating on the AWGN channel and employing a $K=6$, $R=1/2$ convolutional code, was executed in the SIMULATION mode, and both raw and decoded error statistics were collected. Figure 3-17 illustrates the measured symbol-error probability P_e and the decoded bit-error probability P_b . The decoded bit-error probability is plotted as a function of E_b/N_0 , while the symbol-error probability is plotted as a function of E_s/N_0 . Here, E_s is the signal energy on a per-baud basis and is related to E_b according to $E_s = RE_b$, where R is the code rate (in this case $R=1/2$). Also included in the respective plots in Figure 3-17 are the theoretical symbol-error probability P_e and the computed upper bound on P_b . The simulation results compare favorably to these computed results. These results provide a complete picture of the performance of the coded coherent QPSK system on the AWGN channel. The total elapsed time to execute the SIMULATION mode (in this case over the range $0dB \leq E_b/N_0 \leq 6dB$ in steps of 0.5 dB) was 2 hours. This is significant when one observes that error events as rare as one in 10^7 have been accurately measured. Comparable simulation capabilities would be impossible in existing general-purpose machines. Indeed, it is questionable whether such an extensive simulation would be undertaken in the first place.

Other graphical output available in this case would include histograms of the additive channel noise amplitude and phase, two-dimensional histograms of the phase and time synchronization errors, and three-dimensional plots of the fading Channel scattering function. This scattering function is shown in Figure 3-18 for an 21-tap delay line model.

LEGEND

□ - QD0004(R)
○ - QTEST(R)

SYMBOL ERROR PROBABILITY

MENU

MENU
XZOOM
YZOOM
UNZOOM
REDRAW
EXIT

COMMAND

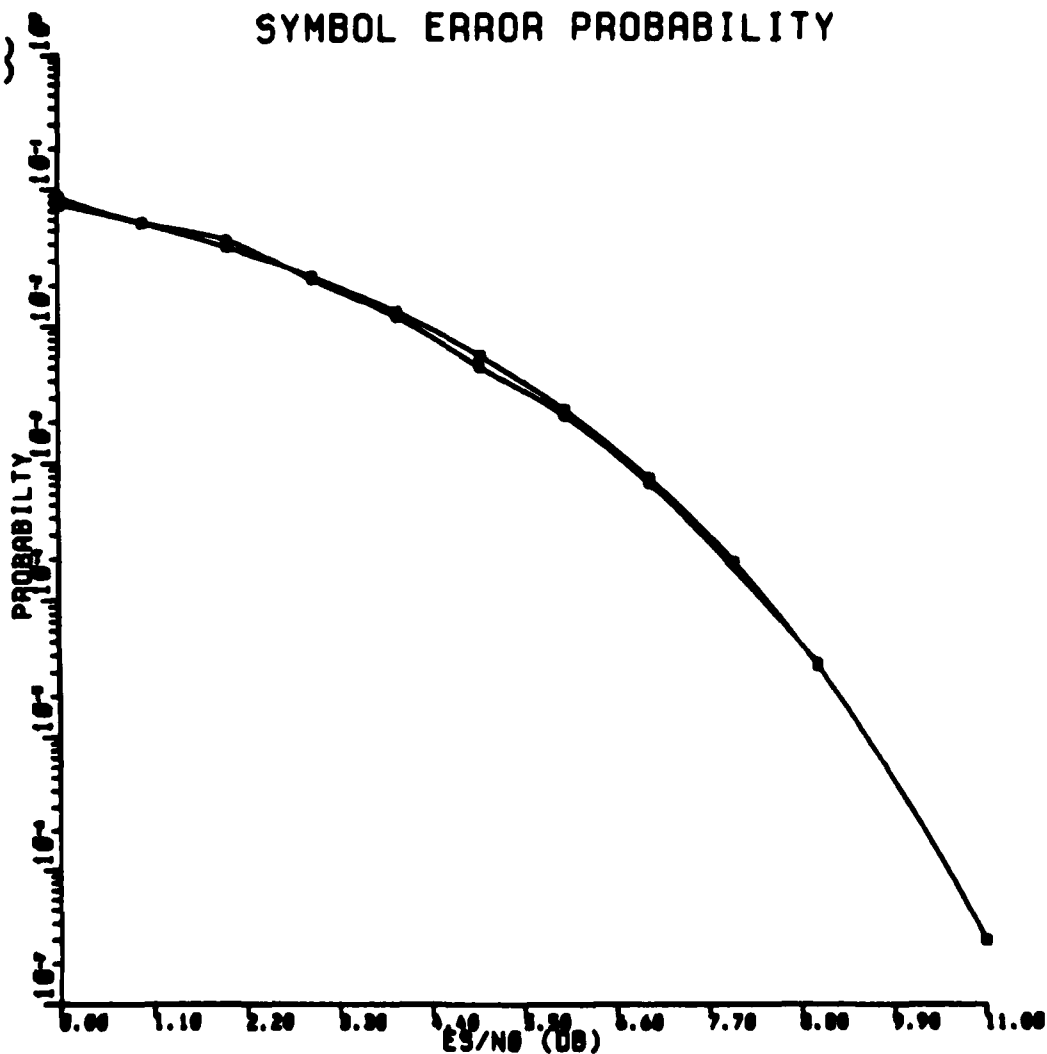


FIGURE 3-17a ICS ERROR ANALYSIS, SYMBOL ERROR PROBABILITY

LEGEND

□ - BTEST(D)
 ○ - BPRE009(R)

MENU

MENU
 XZOOM
 YZOOM
 UNZOOM
 REDRAW
 EXIT

COMMAND

BIT ERROR PROBABILITY

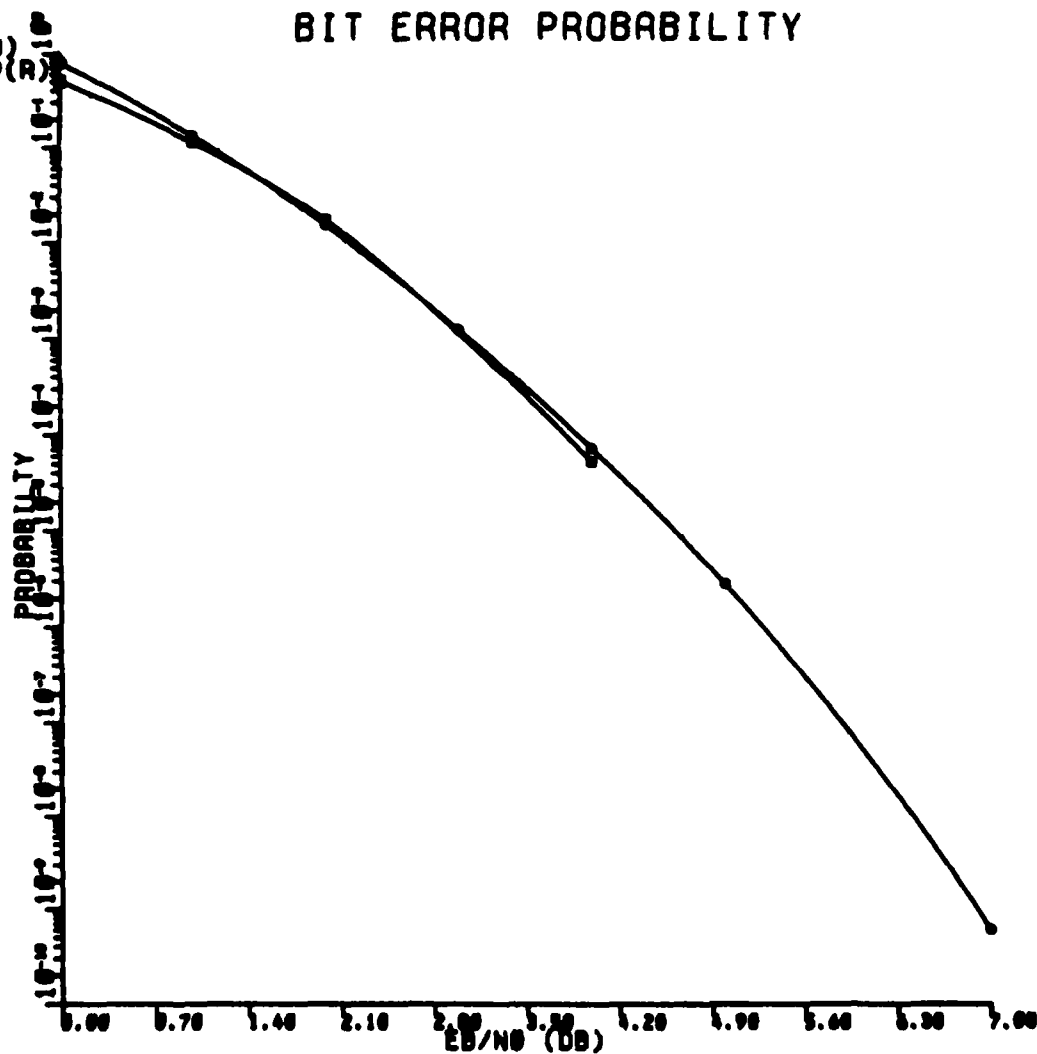


FIGURE 3-17b ICS ERROR ANALYSIS, BIT ERROR PROBABILITY

DESIGN FILE: 0

ICS DESIGN PHASE

11-NOV-83 09:53:49

FADING DISPERSIVE CHANNEL--DIFFUSE COMPONENT
BUTTERWORTH APPROX ORDER=2
NUMBER OF TAPS=21

CLIP;
FLATTEN;
DISTANCE;
REDRAW
UNCLIP
HELP
EXIT

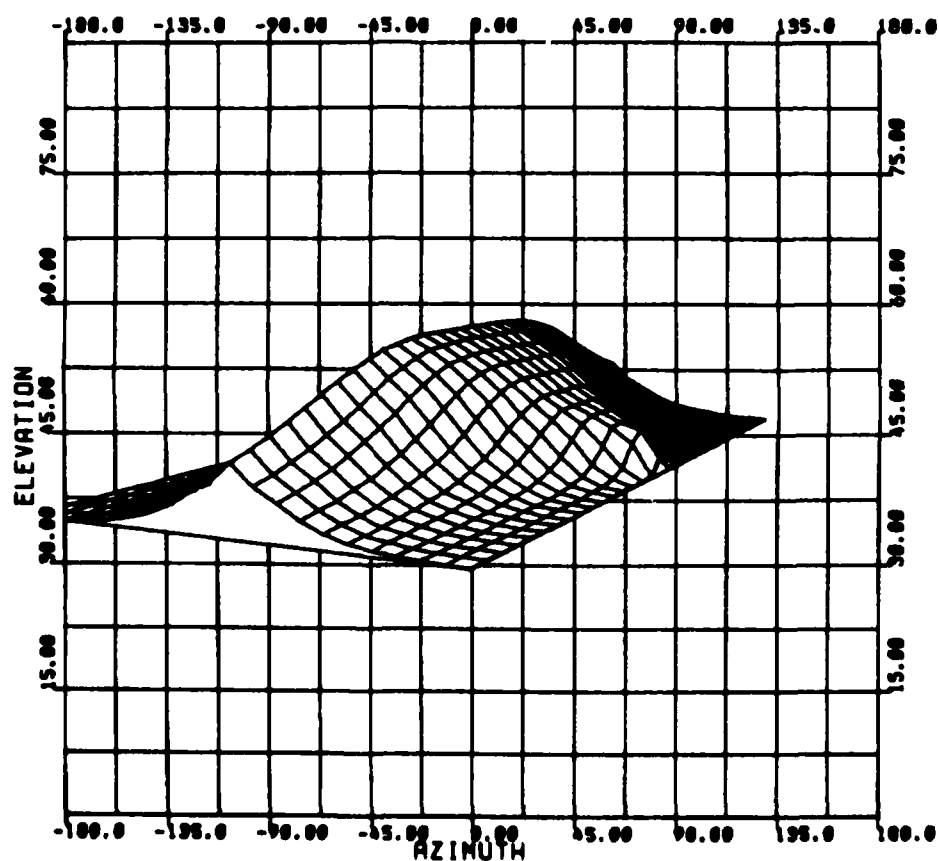


FIGURE 3-18 CHANNEL-SCATTERING FUNCTION

4. CONCLUSIONS AND RECOMMENDATIONS

The enhanced ICS provides a comprehensive facility for the modeling, simulation, and analysis of point-to-point digital communications systems. Applications modules available in ICS include modulators/demodulators, coders/decoders, filters, nonlinearities, and a comprehensive set of modules for the simulation of realistic transmission impairments. These include fading dispersive phenomena as well as ECM interference. The ICS is being employed in the evaluation of single-link communication systems and in the development of state-of-the-art signal processing concepts for military, space, and commercial applications.

The major difficulties encountered in the ICS ENHANCEMENTS efforts were caused by hardware/software limitations of the target computer configuration, a PDP-11/40 computer residing at DICEF. ICS has become a large-scale software system and, as such, will run efficiently only on a virtual memory machine such as a VAX-11 series computer. The limited task size (32K) of PDP-11 series computers severely complicates program development by requiring complex overlay structures. Since overlays run much less efficiently than single image tasks, the overall speed and, hence, usefulness of a PDP-11 based ICS is also severely compromised. The replacement of DICEF PDP-11/40 computer with a VAX-11 series equivalent computer is recommended as the highest priority for future system improvements.

The applications modules provided in the current enhancement effort addressed channel modeling, spread-spectrum techniques, ECM waveform generation, and enhanced graphics capabilities. The major enhancement which was planned but not implemented was an adaptive array simulation capability. The usefulness of the ICS would be greatly enhanced by the inclusion of such a capability. Other improvements which would be useful include improved tutorial interaction, adaptive equalizers, and more comprehensive block and concatenated coding schemes. The capability to collect bit-error and symbol-error statistics other than average error probability would prove quite useful on fading and/or burst noise channels. Any of these applications/enhancements should be accomplished only after an upgrade to a virtual

memory machine, since further growth of the software system will be difficult and can only exacerbate current inefficiencies on the PDP-11/40 based target system.

REFERENCES

- [1] J. P. Odenwalder, "Optimum Decoding of Convolutional Codes," Ph.D. Dissertation, Syst. Sci. Dept., Univ. California, Los Angeles, 1970.
- [2] K. J. Larsen, "Short Convolutional Codes with Maximal Free Distance for Rates $1/2$, $1/3$ and $1/4$," IEEE Trans. Inform. Theory, Vol. IT-19, pp. 371-372, May 1973.
- [3] W. C. Linsey and M. K. Simon, *Telecommunication Systems Engineering*, Prentice-Hall, Englewood Cliffs, NY, 1973.
- [4] J. J. Spilker, Jr., *Digital Communications by Satellite*, Prentice-Hall, Englewood Cliffs, NJ, 1977.
- [5] P. A. Bello, "Characterization of Randomly Time-Variant Linear Channels," IEEE Trans. Commun. Syst.; Vol. CS-11, pp. 360-293, December 1963.
- [6] R. S. Kennedy, *Fading Dispersive Communication Channels*, Wiley-Interscience, NY, 1969.
- [7] J. W. Modestino and B. Sankur, "Modeling and Simulation of ELF/VLF Noise," Proc. Seventh Annual Pittsburgh Conf. on Modeling and Simulation, Pittsburgh, PA, pp. 189-194, April 1976.
- [8] J. W. Modestino and B. Sankur, "Modeling and Analysis of Impulsive Noise," Proc. of NATO Conf. on Communication and Signal Processing, Darlington, England, pp. 619-636, August 1977.
- [9] G. D. Forney, Jr., "Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," IEEE Trans. Information Theory, Vol. IT-18, pp. 363-377, May 1972.
- [10] J. G. Proakis, "Advances in Equalization of Intersymbol Interference," Chapt. 3 in *Advances in Communication*, Ed. by A. J. Viterbi, Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [11] P. Monsen, "Feedback Equalization for Fading Dispersive Channels," IEEE Trans. Inform. Theory, Vol. IT-17, pp. 56-64, January 1971.

- [12] P. Monsen, "Digital Transmission Performance on Fading Dispersive Diversity Channels," IEEE Trans. on Commun., Vol. COM-21, pp. 33-39, January 1973.
- [13] P. Monsen, "Theoretical and Measured Performance of a DFE Modem on a Fading Multipath Channel," IEEE Trans. on Commun., Vol. COM-25, pp. 1149-1153, October 1971.
- [14] A. J. Viterbi, "Convolutional Codes and Their Performance in Communication systems," IEEE Trans. Commun. Tech., Vol. COM-19, pp. 751-772, October 1971.
- [15] G. D. Forney, Jr., "The Viterbi Algorithm," Proc. of IEEE, Vol. 61, pp. 268-278, March 1973.
- [16] J. K. Wolf, "Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis," IEEE Trans. Inform. Theory, Vol. IT-24, pp. 76-81, January 1978.
- [17] J. A. Heller and I. M. Jacobs, "Viterbi Decoding for Satellite and Space Communication," IEEE Trans. Commun. Tech., Vol. COM-19, pp. 835-848, October 1971.
- [18] J. W. Modestino, "Error Probability Bounds for Soft-Decision Decoding of Block Codes," TM 78-30, unpublished RPI report, November 1978.
- [19] P. A. Bello, "A Troposcatter Channel Model," IEEE Trans. Commun. Tech., April, 1969.
- [20] J. L. Ramsey "Realization of Optimum Interleavers," IEEE Trans. Info. Th., Vol. IT-16, pp. 338-345.
- [21] Technical Order, "High Frequency Channel Simulator", Signatron, Inc.
report number A250-19, Signatron, Inc., MA, 13 March 1980, (not published).

APPENDIX A

BLOCK CODES

In a block encoder, the binary sequence $\{a_i\}$ is segmented into blocks of size k to which $n-k$ redundant bits are added to produce a rate $R=k/n$ code measured in units of information bits per transmitted channel symbol. There is complete independence between blocks. The set of all binary n -tuples is a vector space. A set of these vectors of length n is called a linear block code if and only if it is a subspace of the vector space of n -tuples. In the binary case, and in fact over a field of p elements where p is prime, every group of vectors is also a subspace. The term group code is common terminology for binary linear codes.

The Hamming weight of a vector \underline{x} , denoted $w(\underline{x})$ is defined to be the number of nonzero components. Since the Hamming distance between two vectors \underline{x}_1 and \underline{x}_2 is the number of positions in which they differ, the distance between \underline{x}_1 and \underline{x}_2 is equal to $w(\underline{x}_1 - \underline{x}_2)$. If \underline{x}_1 and \underline{x}_2 are both code words of a linear block code, $\underline{x}_1 - \underline{x}_2$ must also be a code word, since the set of all code words is a vector space. Therefore, the distance between any two code vectors equals the weight of some other code vector, and the minimum distance for a linear code equals the minimum weight of its nonzero vectors. This property is extremely helpful in analyzing the error-correction capabilities of linear codes.

Nearly all known block and convolutional codes are based on the concept of Hamming distance. Any set of basis vectors for a linear block code X can be considered as rows of a matrix \mathbf{G} , called a generator matrix of X . The row space of \mathbf{G} is the linear code X , and the vector is a code vector if and only if it is a linear combination of the rows of \mathbf{G} . If the dimension of the vector space X is k , the number of rows of \mathbf{G} (which equals the rank of \mathbf{G} , since the rows must be linearly independent) is k . If any two linear combinations were equal, there would be a dependence relation among rows of \mathbf{G} . Thus, each distinct linear combination gives a distinct code vector, and, since

there are k coefficients with 2 possible values for each, there are 2^k code vectors in X . Such a code is called an (n,k) code.

Unless k is small, the matrix description is much more compact than a list of code vectors. A $(50, 30)$ code is described by a 30×50 matrix but has more than 10^9 code vectors.

Example: A typical code with $N=7$ and $K=4$ has a generator matrix given by

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

This is the Hamming (7, 4) Code with minimum distance $d=3$.

There is an alternative description by matrices. Again, if X is a sub-space of dimension k , its null space is a vector space X' of dimension $n-k$. A matrix H of rank $n - k$ whose row space is X' can be made with a basis for X' as rows. Then X is the null space of X' , and a vector \underline{x} is in X only if it is orthogonal to every row of H , that is, if and only if

$$\underline{x}H^T = 0 . \quad (A-1)$$

If $X = (X_1, X_2, X_3, \dots, X_n)$ and the element in the i th row and j th column of H is denoted h_{ij} , Equation A-1 implies that for each i (that is for each row of H),

$$\sum_j = x_j h_{ij} = 0 . \quad (A-2)$$

Thus, the meaning of Equation A-1 is that the components of X must satisfy a set of $n - k$ independent equations. Of course, any linear combination of the Equation A-2 also gives an equation that the components of X must satisfy, and this corresponds to

the fact that X is orthogonal to every vector of X' . These equations are called generalized parity checks, since in the binary case they are simply checks for even parity on certain sets of symbols in the code word. That is, for each row of H , the number of 1's in X that corresponds to 1's in that row of H is even (for the binary case) if and only if X satisfies Equation A-1. The matrix H is called a parity-check matrix of X .

Equation A-1 holds for every vector x in X . In particular, it holds for the k basis vectors of the matrix G . These k equations can be expressed as follows, with 0 denoting the $k \times (n-k)$ all-zero matrix:

Example: The null space X' of the vector space X of the previous example consists of all possible linear combinations of the rows of

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

The code X is the null space of this matrix. For each vector in X' there is an equation that the components of every code vector must satisfy. For example, corresponding to $(0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1)$ is the equation

$$0X_1 + 1X_2 + 1X_3 + 1X_4 + 1X_5 + 0X_6 + 0X_7 = 0$$

that must be satisfied by every code vector $(X_1, X_2, X_3, X_4, X_5, X_6, X_7)$. For binary codes, this is equivalent to having an even number of 1's among the last four components or an even parity check on the last four components. Note that, unlike vectors over the field of real numbers, a vector can be orthogonal to itself. For example, $(0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0)$ is in both X and X' .

APPENDIX B

NARROWBAND FILTERING

In implementing a digital filter on a digital computer or with special-purpose hardware, the input-output relation must be converted to a computational algorithm. The algorithm is essentially specified in terms of a set of basic computations or elements. For the implementation of discrete-time systems described by linear constant-coefficient difference equations, it is convenient to choose as these elements the basic operations of addition, delay, and multiplication by a constant. The computational algorithm for implementing the filter is then defined by a structure or network consisting of an interconnection of these basic operations. As an illustration, consider a system with a system function of the form

$$H(z) = \frac{\sum_{k=0}^M b_k Z^{-k}}{1 - \sum_{k=0}^N a_k z^{-k}} = \frac{Y(z)}{X(z)} . \quad (\text{B-1})$$

The difference equation relating input and output is easily written down directly from the system function and is given by

$$y(n) = \sum_{k=1}^N a_k y(n-k) + \sum_{k=0}^M b_k X(n-k) . \quad (\text{B-2})$$

We can interpret Equation B-2 directly as a computational algorithm in which the delayed values of the input are multiplied by the coefficients b_k , the delayed values of the output are multiplied by the coefficients a_k , and all the resulting products are added.

A block-diagram representation for the general difference equation of Equation B-2 is given in Figure B-1. The network of Figure B-1 is an explicit graphical representation of the difference equation B-2.

In general, a linear shift-invariant system may have a unit-sample response that is of finite duration, or it may be of infinite duration. Because of the properties of some digital processing techniques, it is useful to distinguish between these two classes. If the unit-sample response is of finite duration, it will be referred to as a finite impulse response (FIR) system; if the unit-sample response is of infinite duration, it will be referred to as an infinite impulse response (IIR) system. If $N = 0$ in Equation B-2, so that

$$y(n) = \frac{1}{a_0} \left[\sum_{r=0}^M b_r x(n-r) \right],$$

it corresponds to an FIR system. In fact, the above difference equation is identical to the convolution sum, and hence it follows directly that

$$h(n) = \begin{cases} \frac{b_n}{a_0} & n=0,1,\dots,M \\ 0, & \text{otherwise} \end{cases}.$$

An FIR system can always be described by a difference equation of the form of Equation B-2 with $N = 0$. In contrast, for an IIR system, N must be greater than zero.

Design of IIR Filters

The traditional approach to the design of IIR digital filters involves the transformation of an analog filter into a digital filter meeting prescribed specifications.

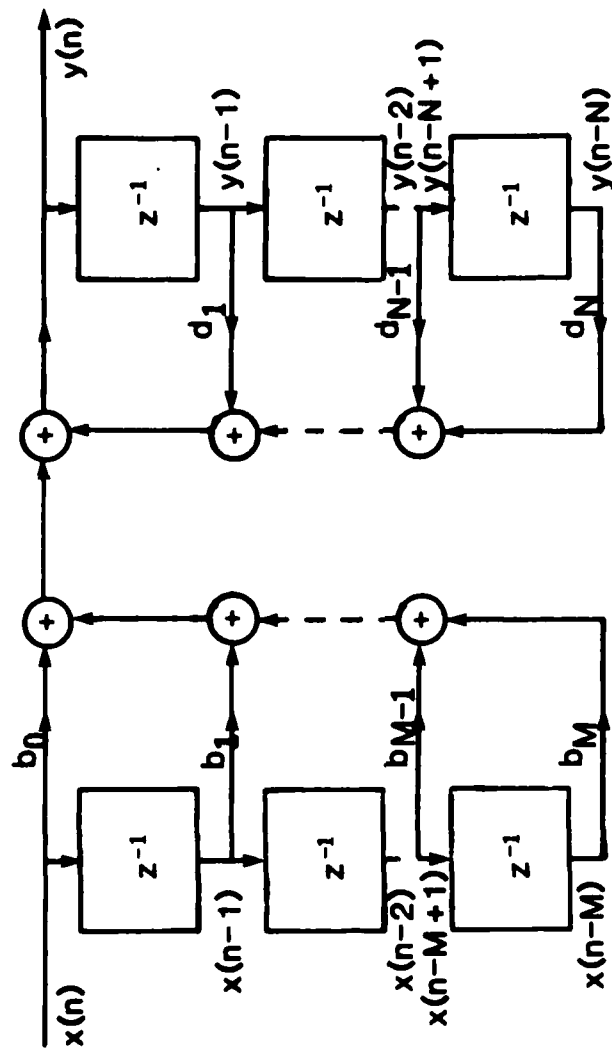


FIGURE B-1 BLOCK-DIAGRAM REPRESENTATION FOR A GENERAL
Nth-ORDER DIFFERENCE EQUATION

The most common design procedure is based on integrating the differential equation and then using a numerical approximation to the integral. This procedure leads to the so called bilinear transformation [B-1] where the z transforms of the digital filter's impulse response $H(z)$ are obtained from the Laplace transform $H(s)$ of the corresponding analog filter by the substitution

$$s = \frac{2}{T_s} \frac{1-z^{-1}}{1+z^{-1}} . \quad (\text{B-3})$$

That is,

$$H(z) = H(s) \Big|_{s = (2/T) (1-z^{-1}) (1+z^{-1})} . \quad (\text{B-4})$$

Solving Equation B-3 for Z gives:

$$z = \frac{1+(T_s/2)s}{1-(T_s/2)s} . \quad (\text{B-5})$$

To demonstrate that this mapping has the property that the imaginary axis in the s -plane maps onto the unit circle, consider $z=e^{j\omega}$. Then, from Equation B-5, S is given by

$$\begin{aligned} S &= \frac{2}{T} \frac{1-e^{-j\omega}}{1+e^{-j\omega}} \\ &= \frac{2}{T} \frac{j\sin(\omega/2)}{\cos(\omega/2)} \\ &= \frac{2}{T} j\tan(\omega/2) \end{aligned}$$

$$= \sigma + j\Omega .$$

Thus for z on the unit circle, $\sigma=0$ and Ω and ω are related by

$$T \frac{\Omega}{2} = \tan(\omega/2) .$$

This relationship is plotted in Figure B-2. From the figure it is clear that the positive and negative imaginary axes of the s -plane are mapped, respectively, into the upper and lower halves of the unit circle in the z -plane.

In addition to the fact that the imaginary axis in the s -plane maps to the unit circle in the z -plane, the left half of the s -plane maps to the inside of the unit circle and the right half of the s -plane maps to the outside of the unit circle, as depicted in Figure B-3.

This can be seen by referring to Equation B-5: for the real part of s negative, the magnitude of the factor $(1+sT/2) / (1-sT/2)$ is less than unity, corresponding to the inside of the unit circle. Conversely, for the real part of s positive, the magnitude of that ratio is greater than unity, corresponding to the outside of the unit circle. Thus we see that use of the bilinear transformation yields stable digital filters from stable analog filters.

Digital Butterworth Filters

Butterworth filters are defined by the property that the magnitude response is maximally flat in the passband. For an N th-order lowpass filter, this means that the first $2N - 1$ derivatives of the squared magnitude function are zero at $\Omega = 0$. Another property is that the approximation is monotonic in the passband and the stopband. The squared magnitude function for an analog Butterworth filter is of the form

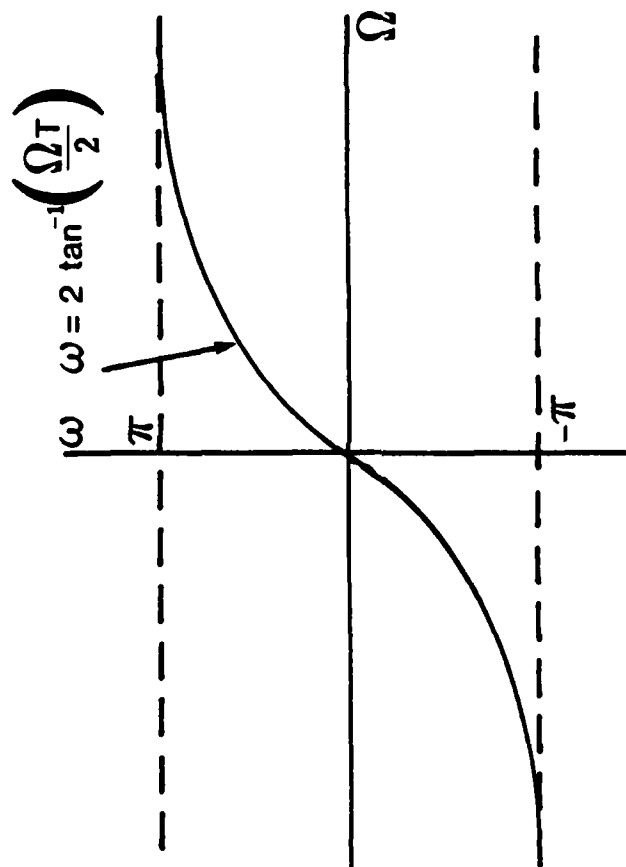


FIGURE B-2 MAPPING OF THE ANALOG FREQUENCY AXIS ONTO THE UNIT CIRCLE USING THE BILINEAR TRANSFORMATION

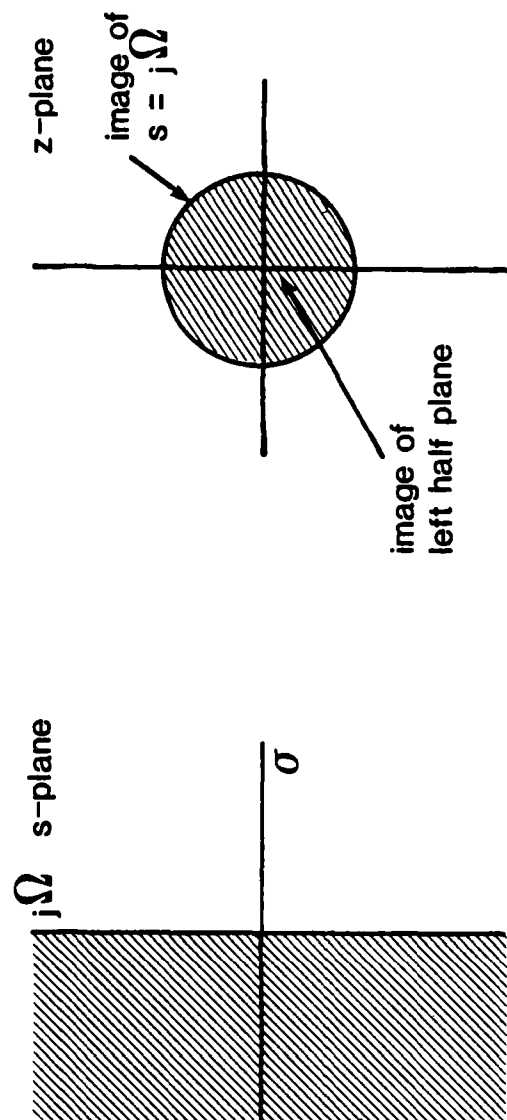


FIGURE B-3 MAPPING OF THE S-PLANE INTO THE Z-PLANE
USING THE BILINEAR TRANSFORMATION

$$\left| H_a(j\Omega) \right|^2 = \frac{1}{1 + (j\Omega/j\Omega_c)^{2N}} \quad (\text{B-6})$$

As the parameter N in Equation B-6 increases, the filter characteristics become sharper; that is, they remain closer to unity over more of the passband and become close to zero more rapidly in the stopband, although the magnitude function at the cutoff frequency Ω_c will always be $1/\sqrt{2}$ because of the nature of Equation B-6. The dependence of the Butterworth filter characteristic on the parameter N is indicated in Figure B-4.

From the squared magnitude function in Equation B-6, we observe that $H_a(s)H_a(-s)$ must be of the form

$$H_a(s)H_a(-s) = \frac{1}{1 + (s/j\Omega_c)^{2N}} \quad (\text{B-7})$$

The roots of the denominator polynomial (the poles of the squared magnitude function) are then at

$$s_p = (-1)^{\frac{1}{2N}} (j\Omega_c)$$

Thus, there are $2N$ poles equally spaced in angle on a circle of radius Ω_c in the s -plane. The poles are symmetrically located with respect to the imaginary axis. A pole never falls on the imaginary axis, and one occurs on the real axis for n odd but not for N even. The angular spacing between the poles on the circle is π/N radians. In designing a Butterworth filter using the bilinear transformation, the most straightforward procedure is to first determine the location of the poles in the s -plane and then to map the left-hand plane poles to the z -plane.

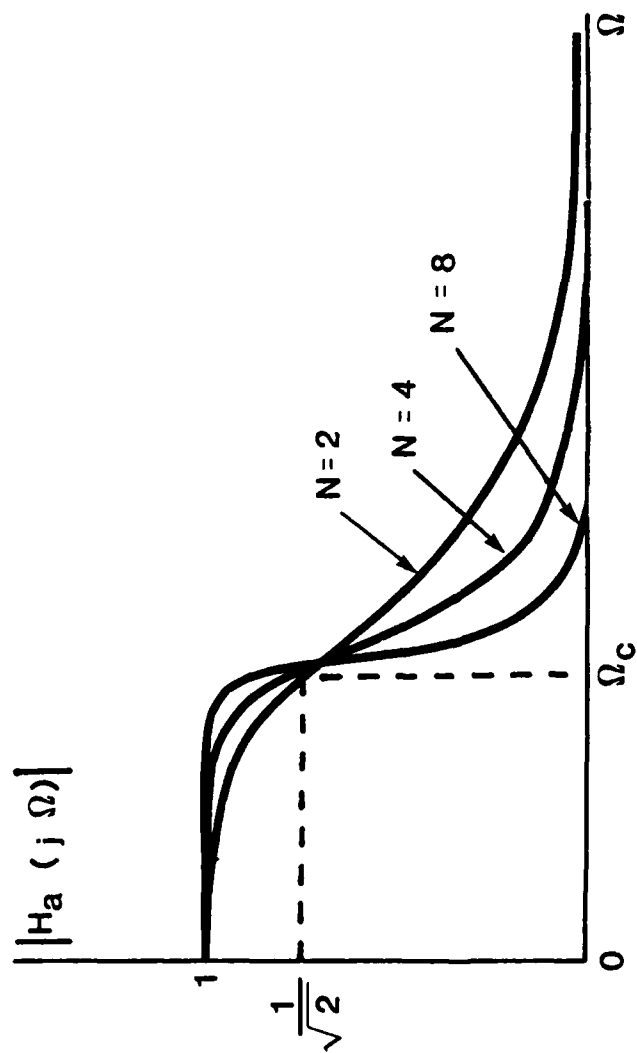


FIGURE B-4 DEPENDENCE OF BUTTERWORTH MAGNITUDE CHARACTERISTIC ON THE ORDER N

Digital Chebyshev Filters

In a Butterworth filter, the frequency characteristic is monotonic in both the passband and the stopband. Consequently, if the filter specifications are in terms of, let us say, maximum passband approximation error, the specifications are exceeded toward the low-frequency end of the passband. A more efficient approach, which usually leads to a lower-order filter, is to distribute the accuracy of the approximation uniformly over the passband or the stopband or both. This is accomplished by choosing an approximation that has an equiripple behavior rather than a monotonic behavior. The class of Chebyshev filters has the property that the magnitude of the frequency response is either equiripple in the passband and monotonic in the stopband or monotonic in the passband and equiripple in the stopband.

The analytic form of the squared magnitude function for the Chebyshev filter is

$$\left| H_a(\Omega) \right|^2 = \frac{1}{1 + \epsilon^2 V_N^2(\Omega/\Omega_c)} , \quad (\text{B-8})$$

where $V_N(x)$ is the N th-order Chebyshev polynomial defined as

$$V_N(x) = \cos(N \cos^{-1} x) . \quad (\text{B-9})$$

For example, for $N = 0$, $V_N(x) = 1$; for $N = 1$, $V_N(x) = \cos(\cos^{-1} x) = x$; for $N = 2$, $V_N(x) = \cos(2 \cos^{-1} x) = 2x^2 - 1$; etc.

From Equation B-9, which defines the Chebyshev polynomials, it is straightforward to obtain a recurrence formula from which $V_{N+1}(x)$ can be obtained from $V_N(x)$ and $V_{N-1}(x)$, by applying trigonometric identities to Equation B-9, with the result that

$$V_{N+1}(x) = 2xV_N(x) - V_{N-1}(x) . \quad (\text{B-10})$$

From Equation B-9 we note that $V_N(x)$ varies between zero and unity for x between zero and unity. For x greater than unity, $\cos^{-1} x$ is imaginary, so that $V_N(x)$ behaves as a hyperbolic cosine and consequently increases monotonically for x greater than unity. Referring to Equation B-8, then, $|H_d(\Omega)|^2$ ripples between 1 and $1/(1+\epsilon^2)$ for $0 \leq \Omega/\Omega_c \leq 1$ and decreases monotonically for $\Omega/\Omega_c > 1$. Three parameters are required to specify the filter: ϵ , Ω_c , and N . In a typical design, ϵ is specified by the allowable passband ripple, and Ω_c is specified by the desired cutoff frequency. The order N is then chosen so that the stopband specifications are met.

The poles of the Chebyshev filter lie on an ellipse in the s -plane.

Design of FIR Filters

The previous section has been concerned with the design of IIR filters using the bilinear transformation. Although such filters have many attractive features, they also have a number of disadvantages. For example, if one wishes to take advantage of the computational speed of an FFT implementation, a finite-duration impulse response is essential. Also, IIR filters generally achieve excellent amplitude response at the expense of non-linear phase. In contrast, FIR filters can have exactly linear phase. Thus, design techniques for FIR filters are of considerable interest.

The system function of a causal FIR filter is of the form

$$H(z) = \sum_{n=0}^{N-1} h(n)e^{-j\omega n} ,$$

i.e., $H(z)$ is a polynomial in z^{-1} of degree $N - 1$. Thus $H(z)$ has $N - 1$ zeros that can be located anywhere in the finite z -plane and $N - 1$ poles all of which lie at $z = 0$. The

frequency response $H(e^{j\omega})$ is the trigonometric polynomial

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n) e^{-j\omega n} \quad (B-11)$$

We recall that any finite-duration sequence is completely specified by N samples of its Fourier transform, so that the design of an FIR filter may be accomplished by finding either its impulse response coefficients or N samples of its frequency response.

If the impulse response satisfies the condition

$$h(n) = h(N-1-n), \quad (B-12)$$

the filter has linear phase. This is easily shown by substituting Equation B-12 into B-11, thereby obtaining

$$H(e^{j\omega}) = \begin{cases} e^{-j\omega((N-1)/2)} \left[h\left(\frac{N-1}{2}\right) + \sum_{n=0}^{(N-3)/2} 2h(n) \cos\left(\omega\left(n - \frac{N-1}{2}\right)\right) \right], & N \text{ odd} \quad (B-13A) \\ e^{-j\omega((N-1)/2)} \left[\sum_{n=0}^{N/2-1} 2h(n) \cos\left(\omega\left(n - \frac{N-1}{2}\right)\right) \right], & N \text{ even} \quad (B-13B) \end{cases}$$

It is seen from these equations that the condition of Equation B-12 implies a linear phase shift corresponding to a delay of $(N-1)/2$ samples. We note that for the case of N odd, the phase shift corresponds to an integer number of samples delay, while for N even, the delay is an integer plus one-half sample.

Since linear phase is generally desirable and often a necessity, and since linear phase often simplifies a design procedure, our discussion will center on linear-phase filters.

The most straightforward approach to FIR filter design is to obtain a finite-length impulse response by truncating an infinite-duration impulse response sequence. If we suppose that $H_d(e^{j\omega})$ is an ideal desired frequency response, then

$$H_d e^{j\omega} = \sum_{n=-\infty}^{\infty} h_d(n) e^{j\omega n}, \quad (\text{B-14A})$$

where $h_d(n)$ is the corresponding impulse response sequence, i.e.,

$$h_d(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega}) e^{j\omega n} d\omega. \quad (\text{B-14B})$$

In general, $H_d(e^{j\omega})$ for a frequency selective filter may be piecewise constant with discontinuities at the boundaries between bands. In such cases the sequence $h_d(n)$ is of infinite duration, and it must be truncated to obtain a finite-duration impulse response. As we have pointed out before, B-14 can be thought of as a Fourier series representation of the periodic frequency response $H_d(e^{j\omega})$, with the sequence $h_d(n)$ playing the role of the "Fourier coefficients." Thus, the approximation of an ideal filter specification by truncation of the ideal impulse response is identical to the study of the convergence of Fourier series, a subject that has received a great deal of study since the middle of the eighteenth century. The most familiar concept from this theory is the Gibbs phenomenon.

If $h_d(n)$ has infinite duration, one way to obtain a finite-duration causal impulse response is to simply truncate $h_d(n)$, i.e., define

$$h(n) = \begin{cases} h_d(n), & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases} \quad (\text{B-15})$$

In general, we can represent $h(n)$ as the product of the desired impulse response and a finite-duration "window" $w(n)$, i.e.,

$$h(n) = h_d(n)w(n) . \quad (\text{B-16})$$

By tapering the window smoothly to zero at each end, the height of the sidelobes in $H(e^{j\omega})$ can be decreased at the expense of a wider main lobe. Typical windows for this application can be found in [B-2].

Another popular method of designing FIR filters is by sampling the frequency response of the desired analog filter. A finite duration sequence can be represented by its discrete Fourier transform.

Thus an FIR filter has a representation in terms of the "frequency samples"

$$\tilde{H}(k) = H(z) \Big|_{z=e^{j2\pi/N}k} = \sum_{n=0}^{N-1} h(n) e^{-j2\pi/N} kn, k=0,1,\dots,N-1 .$$

$H(z)$ can be represented in terms of the samples $\tilde{H}(k)$ by the expression

$$H(z) = \frac{1-z^{-N}}{N} \sum_{k=0}^{N-1} \frac{\tilde{H}(k)}{1-e^{j2\pi/N}kz^{-1}} . \quad (\text{B-17})$$

Equation B-17 serves as the basis of the frequency-sampling realization of an FIR filter. If we let $z=e^{j\omega}$, the frequency response has the representation

$$\begin{aligned} H(e^{j\omega}) &= \frac{1-e^{-j\omega N}}{N} \sum_{k=0}^{N-1} \frac{\tilde{H}(k)}{1-e^{j(2\pi/N)k}e^{-j\omega}} \\ &= \frac{e^{-j\omega((N-1)/2)}}{N} \sum_{k=0}^{N-1} \tilde{H}(k) \frac{\sin [N(\omega - (2\pi/N)k)/2]}{\sin [(\omega - (2\pi/N)k)/2]} . \end{aligned} \quad (\text{B-18})$$

Equation B-18 suggests a simple but rather naive approach to filter design, i.e., to specify the filter in terms of samples of one period of the desired frequency response

$$\tilde{H}(k) = H_d(e^{j2\pi/N}k), k = 0, 1, \dots, N-1,$$

relying on the interpolation indicated in Equation B-18 to "fill in the gaps" in the frequency response.

Frequency sampling designs are particularly attractive for narrow-band frequency selective filters where only a few of the samples of the frequency response are nonzero. In such cases, a frequency sampling realization may be considerably more efficient than either direct convolution or convolution using the DFT. In general, even if more than a few samples are nonzero, the frequency-sampling design method yields excellent results.

Although the results of the frequency sampling design procedure can be very good, it is still not optimum in a mini-max sense. We also lack flexibility in specifying the stopband and passband frequencies W_B or W_P , since all samples are constrained to $2\pi K/N$. It is possible to design optimal (in a prescribed sense) FIR digital filters. These filters are called equiripple because of the form of their magnitude response. If the approximation error in the digital filter's frequency response were spread out uniformly in frequency, a given design specification could be met with a lower-order filter than if the approximation just meets the specification at certain frequencies and far exceeds it at others. Suppose a digital filter design is desired in accordance with the tolerance scheme of Figure B-5. That is, it is desired to approximate 1 in the frequency range $W_s \leq |W| \leq W_p$ within a tolerance $\delta_1 = K\delta_2$ and it is desired to approximate zero in the frequency range $W_s \leq |W| \leq \pi$. The parameter K controls the magnitude of the ripple in the passband relative to the magnitude of the ripple in the stopband. Parks and McClellan [B-3] have developed an iterative procedure to minimize the value of δ_2 given the filter order μ , the value of the stopband frequency W_s , the value of the passband cutoff frequency W_p , and the weighting ratio K . It can be shown that this procedure results in a filter with a frequency response which exactly

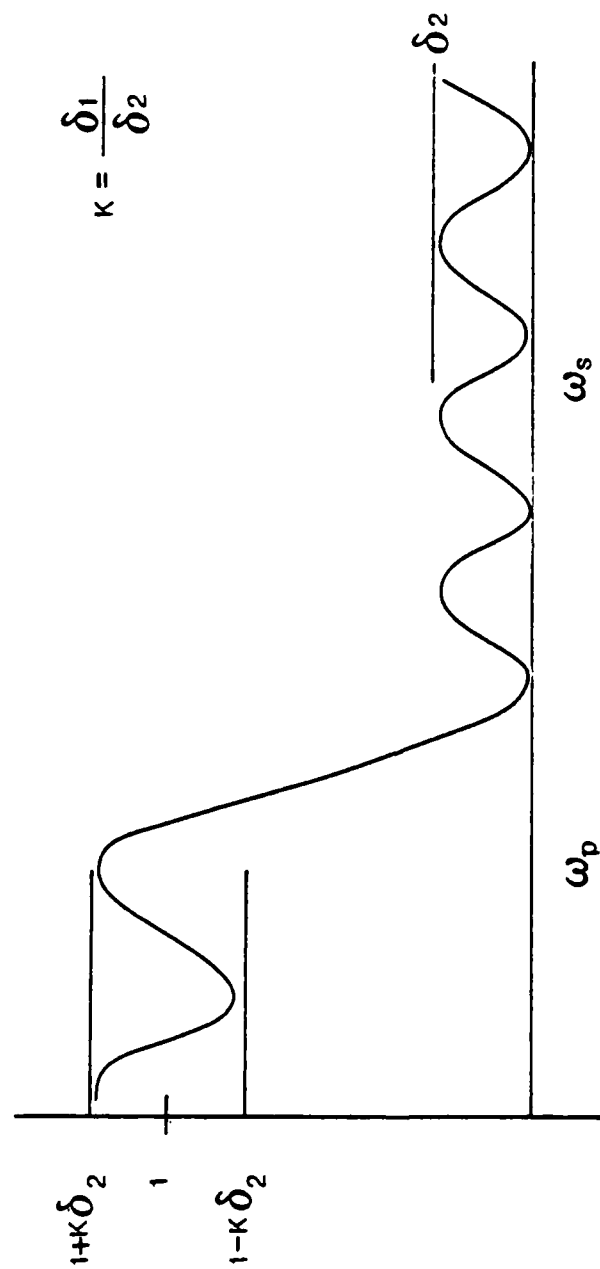


FIGURE B-5
TOLERANCE SCHEME

meets the maximum magnitude error specification of exactly $M + 2$ frequencies including W_s and W_P . This filter is optimum in the sense that no other linear phase FIR filter of the same order can have a smaller value of maximum error in the passband and stopband.

The Parks - McClellan algorithm is incorporated in the ICS design software. A typical response for the case

$$M=64, W_p=.5\pi, W_s=.8\pi$$

and $K = 1$ is illustrated in Figure B-6. In this case, the frequency axis is scaled relative to the baud rate, and thus the frequency response is plotted from $-N_s/2$ to $N_s/2$.

DESIGN FILE: B0B2

ICS DESIGN PHASE
FIRST TRANSMITTER FILTER
FIR FILTER LENGTH= 16

30-JUL-84 16:48:12

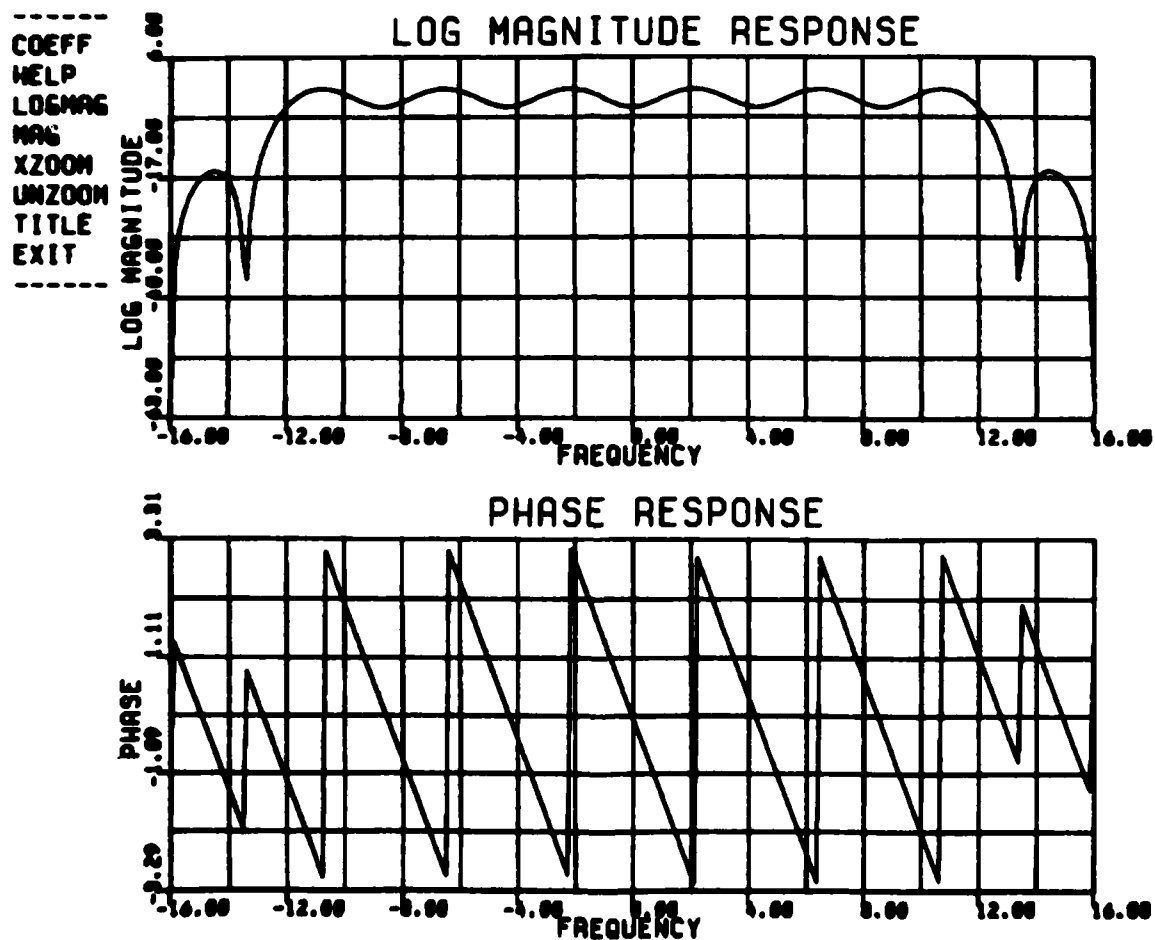


FIGURE B-6 FIRST TRANSMITTER FILTER

APPENDIX B REFERENCES

- [1] B. Gold and C. M. Rader, "Digital Processing of Signals," McGraw-Hill Book Company, New York, 1969.
- [2] R. B. Blackman, and J. W. Tukey, "The Measurement of Power Spectra," Doves Publications, Inc., New York, 1958.
- [3] T. N. Parks and J. H. McClellan, "A Program for the Design of Linear Phase Finite Impulse Response Filters," IEEE Trans. Audio Electroacoustics, Vol. AU-20, No. 3, pp. 195-199, August 1972.

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C³I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

END

FILMED

4-85

DTIC