Simulation System for Analog and Digital Transmissions

GIULIANO BENELLI, VITO CAPPELLINI, MEMBER, IEEE, AND ENRICO DEL RE, MEMBER, IEEE

Abstract—In this paper a complete communication system simulation package is described, which includes digital as well as analog operations to be performed on the signal to be transmitted. The package simulates the transmitter, the communication channel, and the receiver.

The digital transmission simulation considers data communication systems with digital filtering, source and channel coding performing the operations of transmitter coding and receiver decoding. The two coding operations can be realized both separately and independently and in a strictly connected or "integrated" form.

The analog transmission simulation includes the modulator, the transmitter pulse shaping filter, the communication channel, the receiving filter, and the demodulator. This section is very general, so that it allows the simulation of both analog and digital transmissions.

Simulation results are presented regarding some combinations of the following techniques: predictive source coding, block codes able to correct random and burst errors, and modulation techniques such as AM, M-level PSK, FSK, and MSK. In particular, the paper presents the results of the analyses of an integrated source-channel coding applied to digital transmissions and of a system transmitting both voice and data for a VHF communication link between ground and aircraft for air traffic control applications.

I. Introduction

THE design and performance evaluation of a complete communication system is very difficult because of the number of different parameters and subsystems that can be chosen and varied. Although powerful methods are available to the design and system engineer for a theoretical analysis, in many cases the computer simulation of a complete communication system as well as some of its parts is a convenient practical approach—and sometimes the only practical approach. As a few examples we can mention the analysis and design of digital communication systems where finite-precision arithmetic operations are to be taken into account at various points, or where performance evaluation parameters such as the probability of error and the redundancy reduction factor (compression ratio) must be determined. Computer simulation is also useful in the analysis and design of analog communication systems when analytical methods become too cumbersome to yield numerical results, or theoretical solutions are of limited use in practice.

In this paper a complete communication system simulation package is described for both analog and digital transmissions. The package simulates the transmitter, the communication channel, and the receiver. The input signal can be either analog or digitized analog or digital data. The simulation of data communication systems includes the operations of digital filtering, source and channel coding and decoding. In particular, source and channel coding and decoding can be realized both separately and independently and in a strictly connected or "integrated" form. The integration of the two techniques in a unique operation permits one to match the channel coding to the source coding, such that the overall sensitivity to errors introduced by the communication channel is greatly reduced. The package includes the simulation of the modulator, the transmitter pulse shaping filter, the physical transmission channel, the receiving filter, and the demodulator. This part is very general, so that it allows the simulation of both analog and digital transmissions. It operates in both the time domain and the frequency domain, through the appropriate use of the fast Fourier transform algorithm. Section II describes the general organization of the simulation system with its input and output capabilities. Section III describes in detail the processing section of the simulation package, which includes the digital and analog transmissions. Finally, as illustrative applications of the simulation package, Section IV shows the results obtained for two different communication schemes: an entirely digital communication system including source and channel coding, and a hybrid communication system able to transmit both voice and data in the same bandwidth. In particular, the latter scheme has been considered for aircraft-to-ground and ground-to-aircraft communications in an air traffic control environment.

II. General Structure of the Simulation System

The general organization of the simulation system is shown in Fig. 1. The system consists of an input section, a processing section, and an output section.

The input section can deal with both analog signals and digital data. Analog signals are first sampled and quantized. The sampling frequency and the number of quantization bits as well as the quantization law (e.g., linear or nonlinear quantization) can be arbitrarily chosen according to the characteristics of the signal and the transmission system to be analyzed. The digital signal is stored in an intermediate memory (generally the computer magnetic

Manuscript received November 15, 1982; revised July 10, 1983. This work was supported by the National Research Council of Italy (CNR).

G. Benelli and E. Del Re are with the Dipartimento di Ingegneria Elettronica, Università di Firenze, Florence, Italy.

V. Cappellini is with the Dipartimento di Ingegneria Elettronica, Università di Firenze, Florence, Italy, and IROE-CNR, Florence, Italy.

0733-8716/84/0100-0077$01.00 ©1984 IEEE

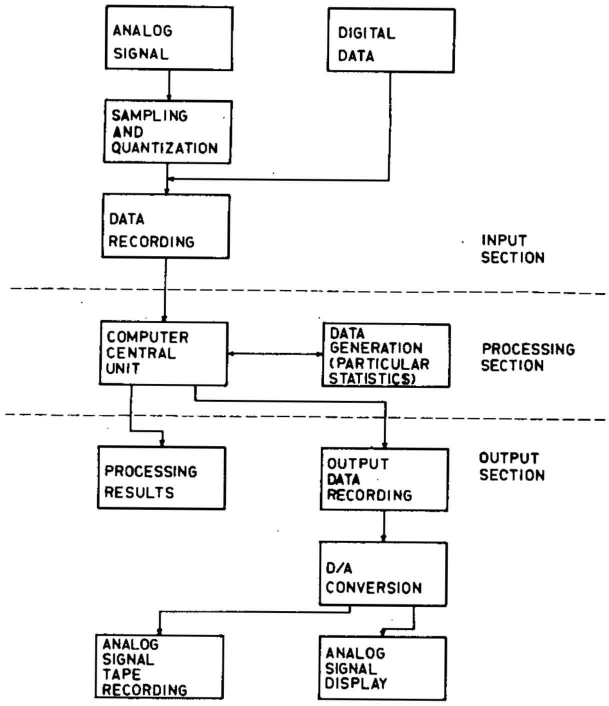

Fig. 1. General organization of the simulation system.

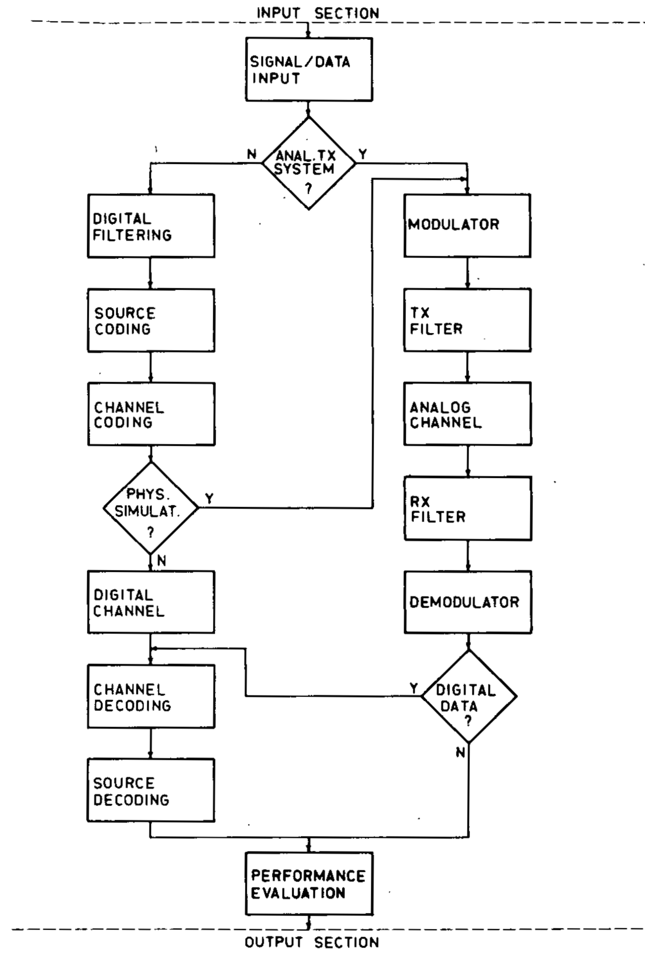

Fig. 2. Processing section.

tape or disk). In case of external digital data, they are directly stored in the intermediate computer memory.

The processing section includes the computer central unit that reads and appropriately processes the stored digital information in order to simulate the transmission system to be analyzed, as will be described in more detail in the following section. The computer may also generate particular data patterns with specified statistics and variable length in order to simulate some desired random data or signals, necessary, for example, for the modeling of noise in the analog communication channel and of random and burst errors in digital transmission.

The results of the computer processing are supplied to the output section. Here the results can be directly outputted, as those concerning the overall performance (e.g., the error statistics and/or the signal-to-noise ratios) of transmission systems, or can be stored in an intermediate memory, as is generally the case for the output signals or data from the simulated transmission system. After an appropriate digital-to-analog conversion, the output section can supply the analog signal suitable for a tape recording and for a direct display.

III. Processing Section

The flow-diagram structure of the whole processing section is shown in Fig. 2.

From the input section, the stored version of the input signal is read first.

In the case of digital transmission system simulation, the processing section can perform the operations of digital filtering, source coding, and channel coding. Afterwards two possible alternatives can occur and be simulated. In the first one (logical simulation), only the two-state (0 and 1) digital stream of data is of interest in any point of the transmission system—in other words, only the logic value of the digital data has to be considered and simulated. In this case the simulation system requires a pure digital channel. After the digital channel simulation, the channel and source decoding operations at the digital receiver are performed. In the second alternative (physical simulation), the waveform of the transmitted digital signal is of interest and therefore must be simulated. This is the case, for example, when baseband or modulated digital transmission must be analyzed. Therefore, the simulation system requires the modulator to shape the digital data, the transmission filter, the analog channel, the receiver filter, and the demodulator. At this point digital data are again obtained at the receiver and must be decoded as in the previous case (channel decoding and source decoding).

When analog transmission systems must be analyzed, after having received the digitized input signal, the processing section simulates the transmitter–channel–receiver chain by means of the same processing blocks as shown in Fig. 2. Of course, after the demodulator, in this case, no channel and source decoding operations have to be performed.

At the end of this simulation procedure, the recovered digital data or digitized analog signal at the receiver output are processed to obtain meaningful parameters for system performance evaluation, and can be compared with the system input data or signal to obtain, for example, the statistics of the error rate of digital transmission systems and the signal-to-noise ratios for analog transmission systems.

The structure shown in Fig. 2 is clearly able to simulate the most general transmission system of interest. Of course, it can be easily adapted when dealing with particular systems having a simpler structure. Any block but the decision blocks can be bypassed without affecting the operations of the other blocks.

Each block of Fig. 2 corresponds to a program package with standardized input and output interfaces to communicate with other blocks. This allows a great flexibility in the actual system configuration and use.

A detailed description of the capabilities of the blocks of Fig. 2 is given in the following subsections. From a general viewpoint we should mention here that the data generation with particular statistics shown in Fig. 1 is included in the digital and analog channel blocks of Fig. 2, as required for a correct simulation of the actual transmission channel behavior. Another general feature is the source decoding operation available in the source coding block (simply obtained by a duplication in this block of the source decoding program package). This capability is required to evaluate the separate contribution of each operation to the whole signal distortion: the comparison between the reconstructed signal at the system output and the input signal measures the overall system distortion; the comparison between the reconstructed output signal and the signal decoded at the source coding block yields the distortion only due to the transmission channel; the comparison, performed inside the source coding block, between the input signal and the signal obtained after the cascade application of the source coding and decoding gives the distortion due to the redundancy reduction operation. In this way, in addition to the overall distortions introduced by the transmission system, the separate contributions of the different operations can be easily evaluated and compared.

In the following two subsections the blocks performing the simulation of digital and analog transmission systems are described, respectively, pointing out the capabilities of the program simulation package.

A. Digital Transmission Simulation

The simulation of a digital transmission system includes the operations of digital filtering, source coding and decoding (data compression), channel coding and decoding, and the effects of the digital transmission channel when the simulation of the physical channel is not of interest.

Digital Filtering: Generally, both finite-impulse-response (FIR) and infinite-impulse-response (IIR) digital filters are of interest. From an implementation point of view, several structures can be simulated for either FIR or IIR digital filters [1], namely:

1) the direct form, which implements directly the digital convolution for FIR filters or the finite difference equation for IIR filters;

2) the cascade form, which corresponds to writing the filter transfer function as a product of second-order factors;

3) for FIR filters, the frequency-sampling structure, where equally spaced samples of the filter frequency response are directly used;

4) for FIR filters, the structure based on an appropriate use of the fast Fourier transform (FFT) algorithm (overlay-add or overlay-save methods), where the filtered signal is obtained through the inverse transformation of the product of the discrete Fourier transforms (DFT's) of the input signal and the filter impulse response;

5) for IIR filters, the parallel form, which implements the transfer function through a parallel connection of IIR second-order sections.

Any of the previous realization structures can be simulated in the digital filtering block. Once the filter structure has been chosen, the program reads the structure parameters as external data. They are supplied by an ancillary program that designs the filter to be used in the simulation and determines the finite-precision representation of its parameters, in order to meet the filter specifications. This software organization allows a convenient separation of the filter design and implementation phases. Of course, the filter implementation software also includes the simulation of the finite-precision arithmetic operations of the actual filter: floating-point or fixed-point arithmetic with truncation or rounding.

Source Coding and Decoding: Many algorithms for data compression or source encoding are available in the literature. We briefly describe some of the better known algorithms to underline the principal characteristics and problems, which are encountered in the simulation of these algorithms.

Algorithms using prediction or interpolation are widely used in data compression for their efficiency and simple implementation [2], [3]. In algorithms using prediction, at the nth sampling instant a prediction of the value of the present sample is performed based on M previous samples [2]. If the difference between the present sample y_n and the predicted sample $y_{p,n}$ is lower than a prefixed tolerance value Δ, then the prediction is considered correct and the sample y_n is not transmitted; otherwise it is transmitted.

The predicted sample $y_{p,n}$ is obtained by a linear weighting of M previous samples:

$$y_{p,n} = \sum_{j=1}^{M} a_j y_{n-j}. \tag{1}$$

By varying M and the coefficients a_j, many different algorithms can be obtained. For example, when $M=1$ we obtain the zero-order predictor (ZOP) ($a_1 = 1$), while for

$M=2$ we obtain the first-order predictor (FOP) ($a_1=2$ and $a_2=1$) and so on.

Data compression algorithms using interpolation utilize both some past and future samples to predict the present sample and are very similar to predictor algorithms [2]. Also, in this case the most utilized algorithms are zero-order interpolator (ZOI) and first-order interpolator (FOI).

Another important class of algorithms are the adaptive algorithms, which adapt themselves to the signal time evolution [3], [4]. In such algorithms, the prediction of the sample at the nth instant is still obtained from (1), but the a_j are varied to follow the time signal activity. The coefficients a_j are determined in such a way as to minimize a specified error criterion as, for example, the mean square error. In this case they are chosen in order to minimize the mean square prediction error

$$\sigma^2(M,N)=\frac{1}{N}\left\{\sum_{k=1}^{N}\left(y_{n-k}-\sum_{j=1}^{M}a_j y_{n-k-j}\right)^2\right\} \quad (2)$$

where N is the number of preceding samples which the predictor utilized to learn the signal time evolution. Many other modifications of this technique, linear and nonlinear, are also considered in the literature [3], [4].

In data compression algorithms, time information symbols must be transmitted together with nonpredicted samples, to give the exact time position of each received sample. The most commonly used methods of time information insertion are [3]

1) time information representing the number of nontransmitted samples;

2) time information identifying the absolute position in the frame of the transmitted samples.

In general, the second method requires a higher number of bits than the first method, but it is less sensitive to channel errors [5].

The simulation structure of the previously described algorithms is quite similar. The subroutine, which performs the compression operation, has as input data the signal samples, the time duration of the signal, the coefficients a_j utilized for the prediction, and the tolerance value Δ. The output data of the subroutine are the compressed vector, which contains nonpredicted samples and the time information symbols, and the compression ratio (see Section III-D on performance evaluation). At the receiver a decompression subroutine is required to reconstruct the signal from the compressed vector. Detailed examples of subroutines for the implementation of the compression and decompression operations are shown in [3] for the ZOP algorithm. In the adaptive algorithms, it is necessary to implement, together with the compression algorithm, a subroutine which computes the coefficients a_j utilized in the prediction of the samples.

Other important data compression methods, which have found a wide application, are differential pulse code modulation (DPCM) and delta modulation (DM) [3], [4]. The structure of the simulation program for DPCM and DM algorithms is similar to that of predictor and interpolation algorithms. The structure of DPCM algorithms depends strictly on the particular methods utilized for the prediction of the actual sample and for encoding the difference e_n between the present and predicted samples. For example, encoding e_n by a Huffman code, it is necessary to implement a table which gives, for each value of e_n, the corresponding codeword.

Other codes that can be implemented by the source coding block are tree codes using a fidelity criterion to adapt their strategy to the source behavior.

The efficiency of a compression algorithm is often given through the compression ratio, the mean square error, and the peak error as explained in Section III-D.

Channel Coding and Decoding: After data compression or source encoding, a channel coding operation is generally introduced to reduce the effects of channel noise and disturbances. The methods used for channel coding are strictly dependent on the characteristics of the noise introduced by the communication channel.

The memoryless channel is modeled as a binary symmetric channel (BSC), shown in Fig. 3, where p represents the bit error probability and $q=1-p$. The channel with memory is simulated using a model proposed by Gilbert [6]. In this model, the channel is described using a Markov chain with two states (Fig. 4). State B (burst) simulates the behavior of the communication channel during the burst periods, while state G (good) simulates the behavior of the communication channel during periods between bursts. We have denoted with p_G and p_B the probability of transition $B\to G$ and $G\to B$, respectively, with q_G and q_B the probability of remaining in state G and B, respectively. Of course, $q_G=1-p_B$ and $q_B=1-p_G$. To simulate the burst behavior of the channel, the transition probability p_G and p_B are generally assumed small. In the states B and G, an error probability equal to h and k, respectively, is present. In general $h\gg k$, to take into account that in the burst state the error probability is high. An error probability is included in the state G to simulate random errors in the time between bursts [7].

The time in which the model remains in a state is assumed to have a geometric distribution with mean $1/p_B$ for state G and $1/p_G$ for state B. These hypotheses seem reasonable in many cases.

Some of the more interesting classes of block codes, utilized in our simulation program, are briefly described in the following.

A block code, having a codeword length equal to n and k information symbols, is defined by a matrix G, called generator matrix, having dimension $k\times n$ [8]. If $\boldsymbol{d}$ is the information vector with k components, then the corresponding codeword $\boldsymbol{c}$ is obtained by

$$\boldsymbol{c}=\boldsymbol{d}G. \quad (3)$$

The code can be defined also through the parity-check matrix H, having dimension $(m-k)\times m$. If $\boldsymbol{c}$ is a codeword, then

$$\boldsymbol{c}\cdot H^T=\sum_{i=1}^{n}c_i\boldsymbol{h}_i=0 \quad (4)$$

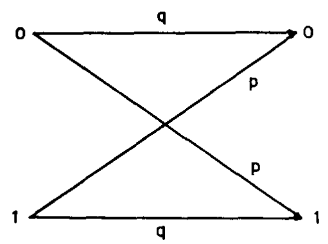

Fig. 3. Model for a binary symmetric channel.

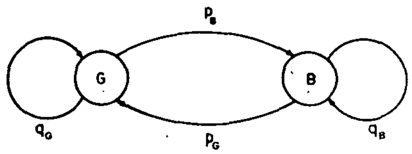

Fig. 4. Gilbert model for a digital channel with memory.

where T indicates matrix transposition and $\boldsymbol{h}_i$ is the ith column of H. If we denote with $\boldsymbol{r}$ the received vector, then $\boldsymbol{r}=\boldsymbol{c}+\boldsymbol{e}$ where $\boldsymbol{e}$ is the error vector. The syndrome $\boldsymbol{s}$ is a vector with $m=n-k$ components and is defined as

$$\boldsymbol{s}=\boldsymbol{r}\cdot H^T=\boldsymbol{e}\cdot H^T=\sum_{i=1}^{n} e_i\boldsymbol{h}_i \tag{5}$$

where e_i is the ith component of the vector $\boldsymbol{e}$.

Channel encoding and decoding operations for a block code or a convolution code can be performed by using the generator matrix G or the parity-check matrix H. In our simulation program we utilize the parity-check matrix because when $k>m$ (as in most cases) the matrix H has lower dimensions than the matrix G. The codes considered in our simulation are in a systematic form, where the first k symbols of every codeword are the information symbols to be encoded and the last m symbols are the redundancy symbols.

In order to simplify the structure of the simulation program, the channel encoding operation (4) and the syndrome computation (5) use the same subroutine with different input parameters. The block diagram of this subroutine, called CODDEC, is shown in Fig. 5. If the parameter IDEC is equal to 0, this subroutine is used for the channel coding operation, while if IDEC $=1$, it is used for the computation of the syndrome vector. The parameter L is defined in the main program and is set equal to k when the subroutine is used for simulating the channel coding operation, while it is set equal to n when the subroutine is used for computing the syndrome. The vector IVET(I) represents the information symbols when the channel coding is performed, while it represents the received vector when the syndrome is computed. Analogously, $S(I)$ contains the redundancy symbols or the syndrome symbols according to whether the channel coding or the syndrome computation is simulated.

An important subclass of block codes are the cyclic codes, which contain many good classes of codes and are particularly simple to implement. A cyclic code (n,k) can be defined through a polynomial $g(x)$, having degree equal to $n-k$. If $d(x)$ is a polynomial associated with the

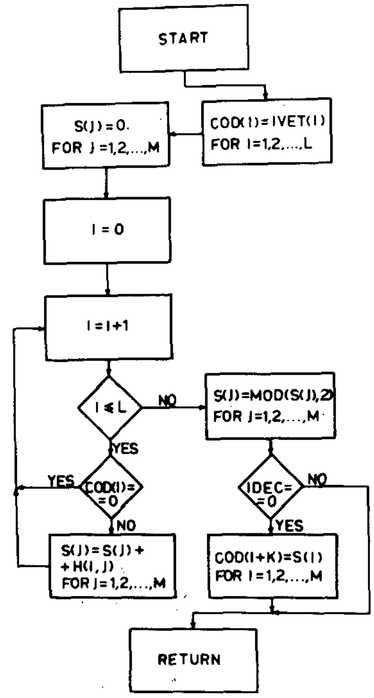

Fig. 5. Block diagram of the channel coding and decoding for a code using the parity-check matrix.

information symbols, then this polynomial is divided by $g(x)$:

$$d(x)=q(x)g(x)+r(x) \tag{6}$$

where $q(x)$ and $r(x)$ are the quotient and the remainder of the division of $d(x)$ by $g(x)$, respectively. The corresponding codeword $f(x)$ is then

$$f(x)=d(x)-r(x)=q(x)g(x). \tag{7}$$

Cyclic codes for random error correction, implemented in our simulation system, are BCH codes, Hamming codes, and Golay codes, while for burst error correction, Fire codes and Burton codes have been considered [8].

Coding and decoding operations for cyclic codes are performed on Galois fields, GF(q^m), where q is a prime number and m is an integer; therefore, in the simulation of cyclic codes, it is necessary to establish and store a table, which gives, for each element of GF(q^m), its binary representation. This table is obtained through a subroutine which simulates the behavior of the shift-register corresponding to the primitive polynomial which generates the Galois field [8]. A similar subroutine is also utilized for the computation of the syndrome, during the decoding operation.

The simulation of Galois fields, particularly of high orders, often requires high computer processing time. Codes defined on the field GF(p) of the integers modulo a prime number p are, on the contrary, computationally simpler for implementation and simulation on a general computer. Some good classes of codes defined on these fields are known. A typical example is the binoid code [9], [10], which can be used for burst-error correction. If $a_0, a_1, \cdots, a_{p-1}$ are the p elements of GF(p), a binoid code able to correct a burst with length b or less is defined by a parity-check matrix H:

$$H = \begin{vmatrix} I_b & I_b & \cdots & I_b & I_b & 0 \\ a_0 I_b & a_1 I_b & \cdots & a_{p-1} I_b & 0 & I_b \end{vmatrix} \quad (8)$$

where I_b is the $b \times b$ identity matrix.

Other important burst-error-correcting codes defines on the field of integers modulo p are generalized Hamming codes [11].

Integration of Data Compression and Channel Coding: Data compression algorithms are often very sensitive to the channel errors; indeed, the signal is reconstructed using a lower number of bits with respect to the input bits; therefore, an error can introduce a greater distortion. This is particularly true, for example, for predictor and interpolator algorithms [12]–[14]. For these algorithms, the errors tend to propagate for many samples and can even destroy the shape of the signal for a long time interval. In many cases, it is convenient to strictly integrate the channel coding operation within the structure of the compressed data to optimize the two operations. We describe a typical example of the integration of data compression and channel coding, when predictors or interpolators are utilized [6].

As we have previously outlined, in data compression algorithms using predictors or interpolators, a time information symbol $\{t_i\}$ is transmitted after each nonpredicted sample y_i. Therefore, the transmitted vector is of the form

$$(y_1, t_1, y_2, t_2, \cdots, y_i, t_i, \cdots). \quad (9)$$

If the symbol t_i represents the time position in a frame of the nonpredicted sample y_i (second method), then the sequence t_i is strictly increasing. This property can be utilized to detect and correct some error patterns and, therefore, to aid the channel decoding. If errors change some t_i' such that the sequence is not increasing, the errors are detected. If the ith received symbol t_i' is such that $t_i' > t_{i+2}$ or $t_i' < t_{i-2}$, then we can suppose with high probability that the ith received symbol is wrong. Then t_i' can be replaced by a value equal to the mean t_{i-1} and t_{i+1}. This procedure enables us to correct, approximately, the errors in many situations, particularly when the transmission channel is without memory. In this case, the errors are often isolated and two or more consecutive symbols rarely are simultaneously altered by a channel. Then the conditions for an exact identification of erroneous symbols are often met.

Of course, this procedure is not suitable in the presence of burst errors. In fact, many consecutive symbols t_i can be altered by a burst and the previous conditions are often not satisfied. Now we illustrate a new strategy using the second time information method, which also presents a good performance for burst-type errors. The general structure of this algorithm is shown in Fig. 6. The data compression operation is performed so that a fixed number of samples greater than $N_{\max}$ is not eliminated consecutively. At the receiver, first, the t_i succession is examined to see if the sequence of t_i is increasing and obeys the previous restriction on $N_{\max}$. Until this is verified, either no error or some undetected errors occurred and therefore nothing is modified. When some consecutive t_i are detected in error following the previous criterion, they are replaced by new t_i equidistant between themselves and with values included between the last exact value which precedes them and the first exact one found after the wrong one. The values of the samples relative to the wrong t_i and those immediately following are modified because they are generally in error: the values that are now assigned to the wrong samples are obtained through a weighted mean of the values of the exact samples which precede them and of those which follow them. The weights given to the exact samples are inversely proportional to the distance between the sample to be replaced and the exact one. Of course, this procedure can also be utilized on memoryless channels, excluding the part relative to the interpolation of samples because errors are uncorrelated. An improvement is obtained in this way, particularly for high bit error probability. Nevertheless, errors on samples are not detected. The problem can be solved using a slight error protection of samples. In Section IV, as examples, we will consider each sample y_i encoded through a parity-check code with one redundancy bit able to detect an odd number of errors. When a sample y_j is

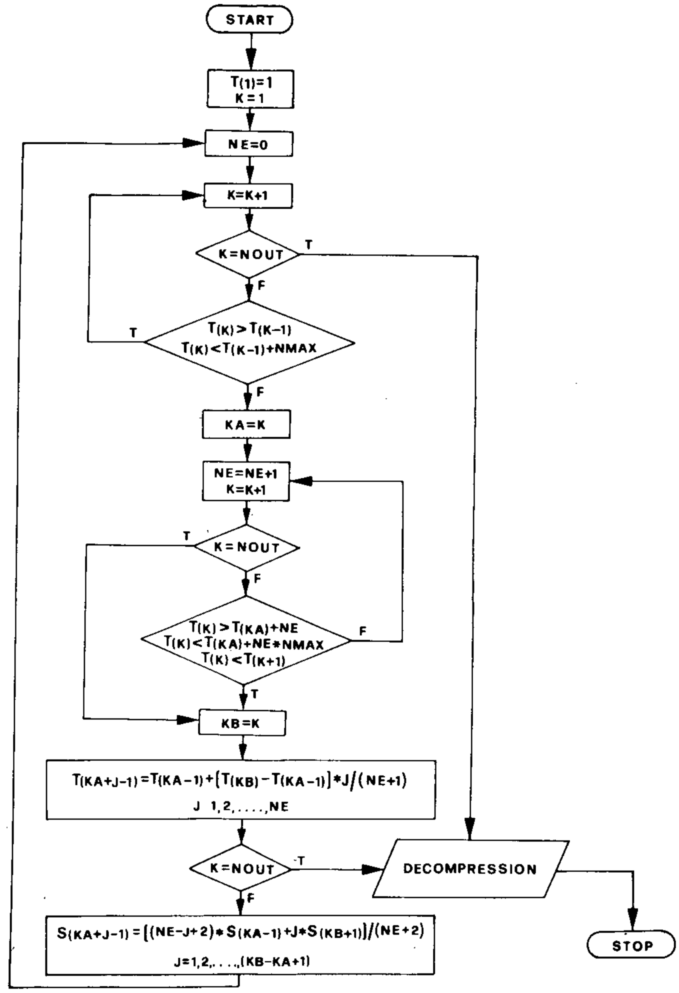

Fig. 6. Block diagram of an integrated data decompression and channel decoding algorithm.

detected in error, its received value is replaced by a weighted mean between the adjacent samples which are correctly received. This procedure permits us to obtain a net improvement in the performance while the compression ratio is not significantly reduced.

B. Analog Transmission Simulation

In the simulation of modulated signals, it is often convenient to deal with an equivalent baseband model to reduce computation time and sample number. In this respect, if $s(t)$ is a modulated signal having a frequency spectrum band limited in the band $(f_0 - B, f_0 + B)$, where f_0 is the carrier frequency and $2B < f_0$, it can be written either in the form

$$s(t) = A(t)\cos[\omega_0 t + \phi(t)] \tag{10}$$

where $\omega_0 = 2\pi f_0$, or in the form

$$s(t) = x(t)\cos\omega_0 t - y(t)\sin\omega_0 t \tag{11}$$

with

$$\begin{aligned} x(t) &= A(t)\cos\phi(t) \\ &= s(t)\cos\omega_0 t + \hat{s}(t)\sin\omega_0 t \\ y(t) &= A(t)\sin\phi(t) \\ &= \hat{s}(t)\cos\omega_0 t - s(t)\sin\omega_0 t \end{aligned} \tag{12}$$

where $\hat{s}(t)$ is the Hilbert transform of $s(t)$. The signals $x(t)$ and $y(t)$ are the in-phase and quadrature components of $s(t)$, respectively, and are low-pass signals in the frequency range up to B [15].

As an example, a PSK modulated signal can be written as

$$s(t) = \sqrt{2E}\cos(\omega_0 t + \phi_k), \qquad kT \leqslant t \leqslant (k+1)T \tag{13}$$

where E is the average power of the signal, ϕ_k is 0 or π if the kth data d_k to be transmitted is 0 or 1, respectively, and T is the symbol interval.

An MSK modulated signal can be written as [16]

$$s(t) = \sqrt{2E}\cos\left(\omega_0 t + \frac{\pi}{2T}d_k t + \alpha_k\right) \tag{14}$$

where the constants α_k are determined in such a way as to obtain a phase continuity at the end of the signaling intervals:

$$\begin{cases} \alpha_k = \alpha_{k-1} + (d_{k-1} - d_k)\dfrac{k\pi}{2} \\ \alpha_0 = 0. \end{cases} \tag{15}$$

The MSK signal can also be written in the form

$$s(t) = \sqrt{2E}\sum_n [y_{2n-1}g(t-2nT)\cos\omega_0 t - y_{2n}g(t-2nT-T)\sin\omega_0 t] \tag{16}$$

where

$$g(t) = \begin{cases} \cos\dfrac{\pi}{2T}t & \text{for } -T \leqslant t \leqslant T \\ 0 & \text{elsewhere} \end{cases} \tag{17}$$

and the sequence of binary symbols $y_n = \pm 1$ is obtained recursively from the d_k [16]. Therefore, the MSK signal (14) is expressed in the form (16). The form (16) is also valid for the PSK case, setting $y_{2n} = 0$, $T = T_s/2$, with T_s the PSK symbol time interval and

$$g(t) = \begin{cases} 1 & \text{for } -T \leqslant t < T \\ 0 & \text{elsewhere.} \end{cases} \tag{18}$$

In this way the same subroutine can be utilized to simulate both PSK and MSK modulations.

Other modulations considered are quaternary PSK (QPSK) and FSK for data transmission and amplitude modulation for the transmission of analog signals as will be used in the example of Section IV.

In the simulation chain of Fig. 2, including the modulator, the transmission filter, the analog channel, the receiving filter, and the demodulator, the signals are generally represented by their equivalent baseband model (baseband representation). An exception to this computationally efficient approach for the signal representation is when dealing with nonlinear operations, as will be outlined in the following.

Transmission Filter, Analog Channel, and Receiving Filter: The block diagram of the program package usually employed in the simulation of analog transmission systems is shown in Fig. 7. The simulation of the transmission filter, the analog channel, and the receiving filter is carried out in the frequency domain through the appropriate use of the fast Fourier transform (FFT) algorithm. The baseband representation of the modulated signal is first transformed in the frequency domain by means of the FFT algorithm; then the operations of multiplication by the samples of the transmission filter frequency response, multiplication by the samples of the analog channel frequency response, addition of the FFT-transformed samples of the baseband representation of the additive channel noise, and multiplication by the samples of the receiving filter frequency response are performed in this order. Finally, the obtained result is transformed again in the time domain by applying an inverse FFT algorithm before supplying the signal to the demodulator input. It is well known that the correct time behavior of the signal from two or more linear systems in cascade can be obtained through the use of the FFT algorithm by applying the overlay-save or overlay-add methods [1]. Either method can be used in the simulation package at the choice of the user.

The software organization of Fig. 7 is able to simulate linear transmission systems and channels with additive-type noise. This is by far the most frequent case. In the few cases where nonlinear operations are involved in some part of the signal path (for example, in the presence of a nonlinear channel), the corresponding simulation package

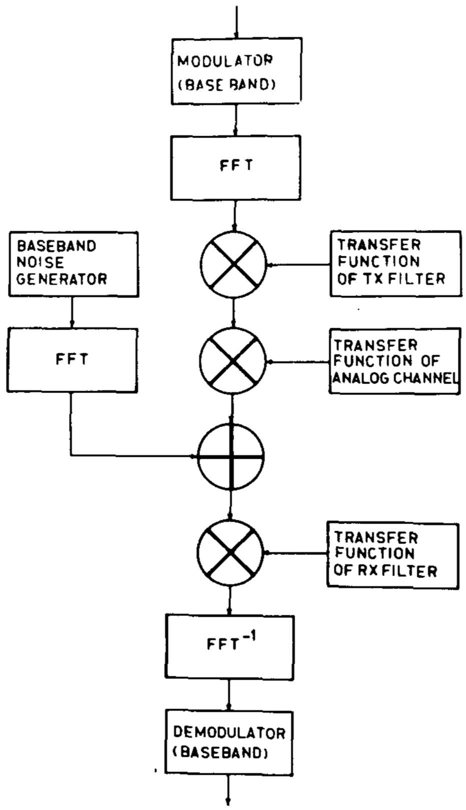

Fig. 7. Block diagram of the analog transmission simulation.

operates in the time domain. This is easily obtained including an inverse FFT transformation at the package input and a direct FFT transformation at the package output after the time domain nonlinear simulation. Generally, in this case, the simulation operations cannot be performed on the baseband signal representation and require the recovery of the signal in the correct modulated form.

C. Importance Sampling Techniques

The error probability is one of the most important parameters which characterize the performance of a communication system. In many cases such a parameter assumes very low values. To evaluate, through a computer simulation, the error probability P_e in a communication system, a number of samples $n > 10/P_e$ is required. For low P_e, the number of samples, which must be processed using standard Monte Carlo simulation techniques, can become prohibitively high. Some simulation methods are available which permit a reduction in the sample number n and therefore in the computation time [17], [18]. These techniques are essentially based on the strategy of increasing the error causes and weighting the obtained values at the output to compensate for the previous distortion. These techniques are called reduced variance or importance sampling techniques.

We now illustrate the general principles in the application of the importance sampling to compute the error probability. If $x(t)$ is the input signal, and $y(t)$ the received signal, then

$$y(t) = x(t) + n(t) \tag{19}$$

where $n(t)$ is the noise in the communication channel. In a Monte Carlo simulation, samples x_k and n_k of the input signal and noise, respectively, are generated. At the output of the communication system, the samples y_k are computed and compared with a threshold V: if y_k is greater than V, an error is detected. Then, using a standard Monte Carlo simulation, the estimated error probability after n processed samples is

$$P_e = \frac{1}{n} \sum_{k=1}^{n} u(y_k - V) \tag{20}$$

where $u(t)$ is the unit step function. The principle of the importance sampling technique is to increase the probability of the noise samples that give rise to errors at the output. This is obtained by modifying the probability density function $p(x)$ of the noise. For example, we consider the case in which the probability density function $p(x)$ is a Gaussian function with variance σ (Fig. 8). In the importance sampling a new Gaussian probability density function $p^*(x)$ is utilized having a higher variance σ^*. In such a way, noise samples from tails have a higher probability, and therefore error events are more frequent. When an error event occurs using importance sampling techniques, such an error must be weighted with a function

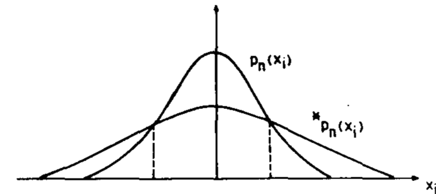

Fig. 8. Example of undistorted, $p(n)$, and distorted, $p^*(n)$, Gaussian probability density functions used in importance sampling techniques.

$$W(x) = \frac{p(x)}{p^*(x)}. \tag{21}$$

Therefore, if n^* is the number of samples processed with importance sampling techniques, then the error probability is

$$P_e^* = \frac{1}{n^*} \sum_{k=1}^{n^*} \frac{p(x_k)}{p^*(x_k)} u(y_k - V). \tag{22}$$

The problem of the choice of the distorted probability density function $p^*(x)$ is discussed by Shanmugan and Balaban [18].

Using these techniques, high gains in the number of samples which must be processed and, therefore, in the computation time are obtained. Of course, a loss in the accuracy of the P_e computation is often obtained with respect to the standard Monte Carlo simulation. Nevertheless, such losses are often low and not important, as depicted in [17].

D. Performance Evaluation

The performance evaluation of analog or digital transmission systems is carried out in the corresponding block of Fig. 2, after the analog signal has been demodulated or the digital data have been decoded.

For analog transmission the following parameters are generally computed:

- received signal mean power, evaluated without the addition of the channel noise shown in Fig. 7;
- received noise mean power, evaluated without transmission signal;
- mean and peak error signals, where the error signal is defined as the difference between the received and transmitted signals (at the user's choice, the system output signal in the presence or in the absence of channel noise can be considered as the received signal).

For digital transmission the following parameters are evaluated:

- probability of error of the received data;
- compression ratio C_a, defined as the ratio of number of bits of the input signal to the number of the received bits (when source coding is included);
- mean and peak error signals (defined as before), when the transmission of digitized analog signals is simulated.

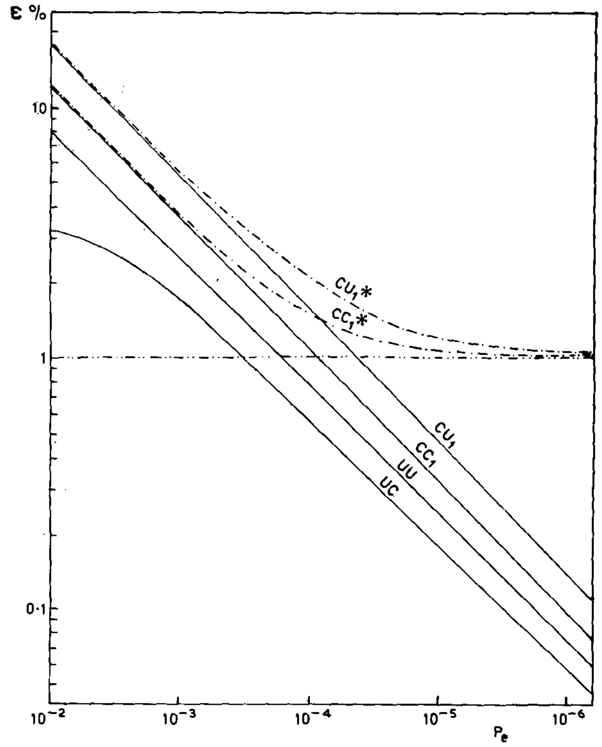

Fig. 9. rms error ϵ versus P_e for a channel with memory.

IV. Examples of Application of the Simulation System

In this section we present some interesting results obtained from the simulation package previously described. In particular, the results refer to two different communication schemes. The first scheme simulates the digital part of a communication system and includes data compression operations and channel coding. The second scheme simulates a complete communication chain, including the analog section.

The first scheme was considered to evaluate the performance of data compression in noisy conditions and their integration with the channel coding operation [13], [19]. To characterize the performance of these systems using data compression, we have simulated four structures: uncompressed-uncoded (UU), compressed-uncoded (CU), uncompressed-coded (UC), and compressed-coded (CC). For all these cases, we have obtained the compression ratio C_a and the rms error ϵ (as a percentage of the full scale signal). In the CU and CC systems, ϵ value includes the distortion introduced by the compression error and channel noise. To show the different influence of these two error types, we have also computed the rms error due only to the channel noise. It was obtained by exploiting the capabilities of the simulation system according to the distortion evaluation procedure outlined in Section III. In the following, the results considering only the distortion introduced by the channel noise are denoted without asterisk, while the total distortion error is indicated with asterisk.

In the figures, data compression algorithms using the first time information method (Section III) are denoted with index 1, algorithms using the second time information method with index 2, and the system in Fig. 6 with index 3.

In the following results, the communication channel is simulated using the Gilbert model (Fig. 4), which describes approximately the behavior of some channels with memory, as telephone channels. The algorithm utilized for data compression is the ZOP algorithm. The code for burst error correction is a Samoylenko binoid code (90, 80) defined in a Galois field GF (31). In the binary transmission the code is of the type (450, 400) and is able to correct all the bursts of length 21 bits or less. The transmitted signal, which is an electrocardiogram (ECG), is first processed using a third-order low-pass digital filter and then compressed by a ZOP algorithm with floating aperture. For a tolerance $\Delta = 2.18$ percent with respect to the full scale, we have obtained an rms error $\epsilon = 1$ percent. It is clear that in the CU and CC cases it is impossible to go below this value. The rms error ϵ versus the channel error probability P_e or signal-to-noise ratio S/N is shown in Figs. 9–11 for all the simulated systems. In Table I the compression ratio C_a for the different systems is shown. It includes the encoding bits as well as the time information bits when present. For high P_e the error ϵ is mainly determined by the channel noise, while the influence of the compression distortion is negligible. By reducing P_e, the importance of the compression errors becomes higher and higher. It is important to note that using the second method for time information (CU_2), the improvement is small with respect to the CU_1 case. At the same time the C_a value is reduced to 2.6. In the CC_2 case,

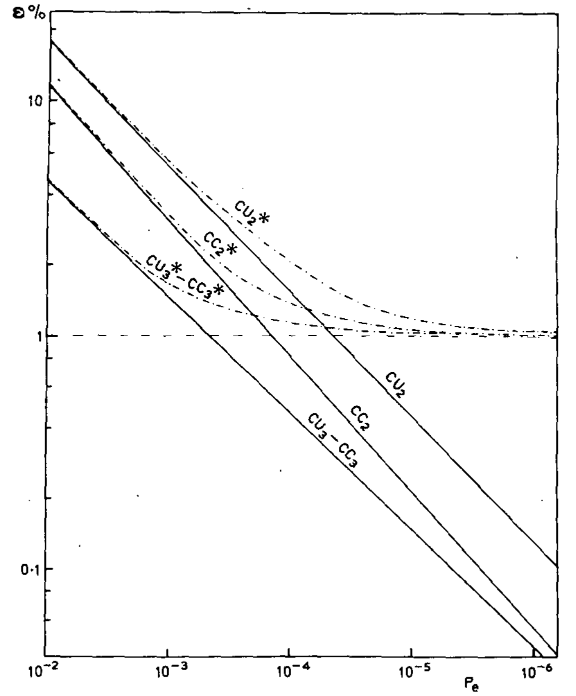

Fig. 10. rms error ϵ versus P_e for a channel with memory.

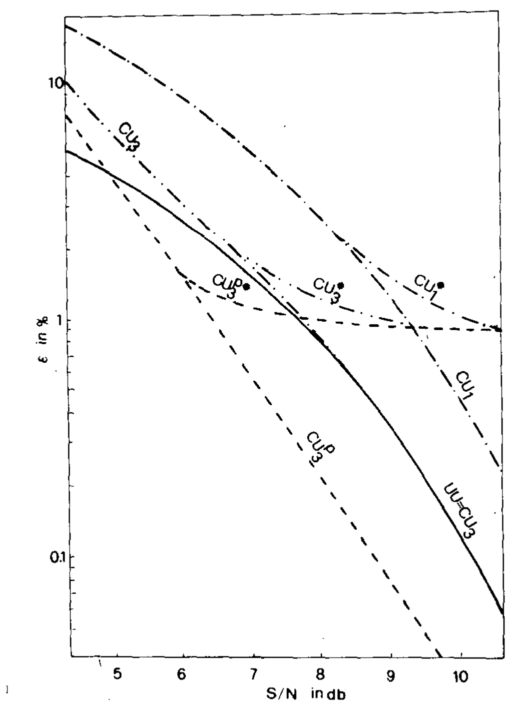

Fig. 11. rms error versus signal-to-noise ratio for a memoryless channel. Index p indicates systems with simple parity-check coding (see Section III).

particularly for low P_e, we have an improvement for ϵ with respect to CC_1, but the compression ratio is reduced to 2.2. The third system (CU_3) shows a very high efficiency; in fact, the rms error due only to the channel error is lower with respect to the UU and UC cases and to the other cases. This follows from the ability to identify in many cases the errors in the time synchronization and, from this, also to reduce the influence of the errors in the most important bits of the samples. The CC_3 case generally has an efficiency similar to CU_3. In fact, when an uncorrectable burst happens, the code can correct some errors in the time information symbols and some information necessary for the identification of the error positions is lost. In this case a higher ϵ can result.

TABLE I
COMPRESSION RATIO FOR SOME SIMULATED DATA COMPRESSION SYSTEMS

	System							
	UU	UC	CU_1	CU_2	CU_3	CC_1	CC_2	CC_3
C_a	1	0.89	3.07	2.6	2.6	2.73	2.16	2.16

The second scheme is a communication system for the simultaneous transmission of voice and data using the same carrier. The voice signal is transmitted by modulating the carrier amplitude, while the data signal modulates the phase of the same carrier [20], [21]. This scheme was studied for the introduction of a data link between aircraft and ground stations through a simple and economical implementation. The general block diagram of this communication system is shown in Fig. 12. It is an interesting and complete example, as it includes two types of modulation, amplitude (AM) and digital (phase or frequency), transmit and receive filters, communication channel, channel coding, and so on.

The digital modulations utilized for data transmission are binary PSK (BPSK), quaternary PSK (QPSK), FSK, and MSK [21].

The bandpass filters in the communication chain were modeled as Butterworth filters with the following characteristics:

- transmit filter: fourth order, with -3 dB single-side bandwidth 7.5 kHz;
- receive filter: eighth order, with -3 dB single-side bandwidth 5 kHz.

They were implemented in the frequency domain as shown in Fig. 7.

The signal was processed in blocks of 2048 samples; the sampling frequency was chosen as 19 200 Hz.

We give first some tests of importance sampling techniques to evaluate the bit error probability in the data signal for this communication system. These tests were carried out in the absence of amplitude modulation and using the BPSK modulation for data transmission. In Fig. 13 we present the bit error probability versus the bit number N_{bit} utilized for the simulation for two different signal-to-noise ratios S/N in the communication channel. Curves (a) refer to a signal-to-noise ratio equal to 6 dB, while curves (b) refer to a signal-to-noise ratio equal to 4.16 dB. These high bit error probabilities were also obtained, through a standard Monte Carlo simulation, performed with $N_{\text{bit}} = 3500$. They are shown by the horizontal dotted lines in Fig. 13 for comparison. It is important to note that the simulated communication chain is quite general and includes many nonlinearities. Nevertheless, also in these cases the accuracy of the importance sampling is high, as it was also verified from other results obtained in the simulation of a radar receiver [17]. The gain in the

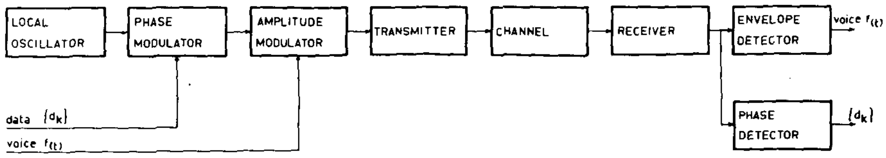

Fig. 12. Block diagram of the communication system transmitting both voice and data.

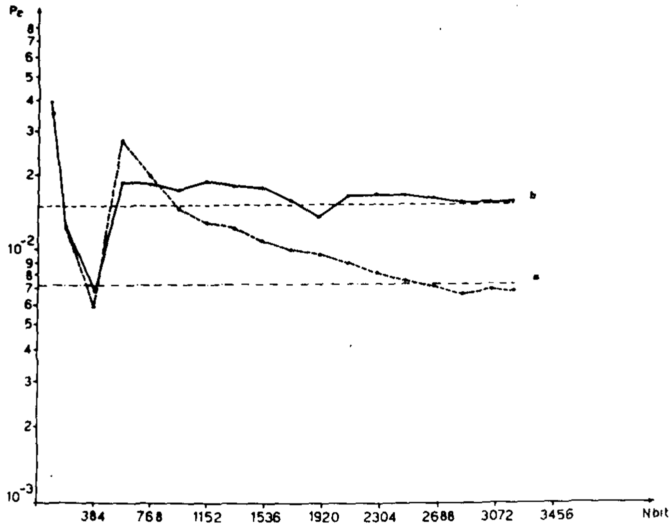

Fig. 13. Bit error probability versus the simulation bit number obtained by an importance sampling technique. (a) $S/N = 6$ dB. (b) $S/N = 4.16$ dB.

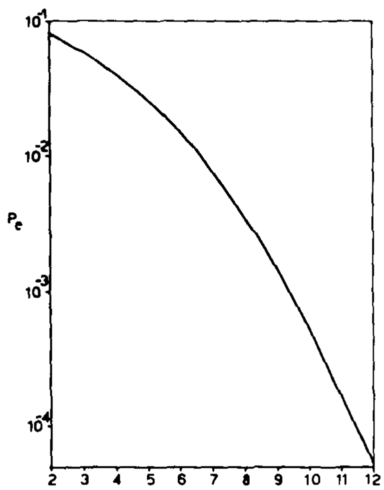

Fig. 14. Probability of error P_e versus signal-to-noise ratio S/N (dB) for a hybrid modulation system (AM-PSK) at a bit rate $v_s = 600$ bits/s.

computation time using importance sampling techniques is often high.

In the following, some results are presented for the system utilizing a hybrid amplitude–phase modulation for the simultaneous transmission of voice and data signals. An application of the importance sampling technique is shown in Fig. 14, where the probability of error P_e versus the signal-to-noise ratio S/N is reported for the communication system of Fig. 12 with AM-PSK modulation and a bit rate v_s equal to 600 bits/s.

An important problem in this communication system is the interference between the two types of modulations:

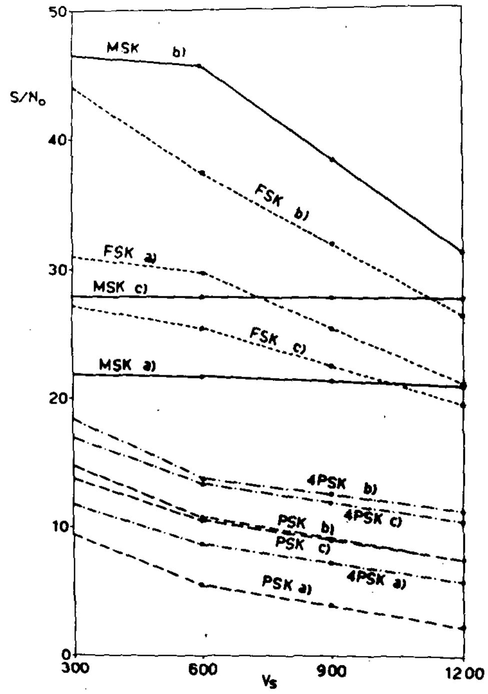

Fig. 15. Signal-to-noise ratio (dB) versus the bit rate for a voice and data communication system.

amplitude and phase. In order to estimate the distortion introduced by data and phase modulation on the voice, the signal at the output of the AM detector $f_1(t)$ is compared to the original signal $f(t)$, which modulates the carrier amplitude. An error signal $e(t) = f_1(t) - f(t)$ is therefore obtained. The simulation program gives some parameters relative to this error signal as the mean power and the signal-to-noise ratio S/N_0. Some different signals were considered in the simulation as analog transmitted signal [21]:

1) a tone at 937.5 Hz frequency
2) a tone at 1875 Hz frequency
3) a sum of five tones, given by

$$f(t) = \sum_{i=1}^{5} Q_i \cos[2\pi i(468.75)t] \tag{23}$$

and the coefficients Q_i are $Q_1 = Q_3 = 2/9$; $Q_2 = 1/3$; $Q_4 = Q_5 = 1/9$

4) a real voice signal having a time duration of 4 s.

The real voice signal is sampled and stored in the computer. After the simulation, the received samples are converted to an analog form and recorded on a tape for a subjective quality assessment from listeners. As an example, in Fig. 15 the signal-to-noise ratio S/N_0 versus the bit rate v_s is reported for the case in which the signal, modulating the carrier amplitude, is a tone at 937.5 Hz [curves (a)], at 1875 Hz [curves (b)], and the sum of five tones [curves (c)]. The modulation index is assumed equal to 0.8.

V. Conclusions

A very general computer simulation package for the analysis, design, and test of communication systems has been illustrated. Practically, it can deal with any kind of communication system of interest. Its modular structure allows one to modify any package module without affecting the other modules in order to fit any particular system requirement. Due to its relative complexity, it has been specifically developed for a large computer system. In particular, it has been implemented on a CII10070 computer. The programming language (ANSI Fortran) allows a sufficient degree of portability for use in other computer environments. Recent advances in the technology of minicomputers allow the package, or at least a somewhat simplified version, to be used in this class too.

The examples of applications of Section IV have shown the capabilities of the package to simulate communication systems of both theoretical and practical interest.

References

[1] V. Cappellini, A. G. Costantinides, and P. L. Emiliani, *Digital Filters and Their Applications.* London, England: Academic, 1978.

[2] C. M. Kortman, "Redundancy reduction: A practical method data compression," *Proc. IEEE*, vol. 55, pp. 253–263, Mar. 1967.

[3] G. Benelli, V. Cappellini, and F. Lotti, "Data compression techniques and applications," *Proc. IERE*, vol. 50, pp. 29–53, Jan.–Feb. 1980.

[4] N. S. Jayant, "Digital coding of speech waveforms: PCM, DPCM, and DM quantizers," *Proc. IEEE*, vol. 62, pp. 611–632, May 1974.

[5] G. Benelli, C. Bianciardi, V. Cappellini, and E. Del Re, "High efficiency digital communications using data compression and error correcting codes," in *Proc. EUROCON 77, Euro. Conf. Electrotech.*, Venice, Italy, May 1977.

[6] E. N. Gilbert, "Capacity of a burst-noisy channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253–1265, Sept. 1960.

[7] E. O. Elliot, "A model of the switched telephone network for data communications," *Bell Syst. Tech. J.*, vol. 44, pp. 89–109, Jan. 1965.

[8] W. W. Peterson and E. J. Weldon, Jr., *Error Correcting Codes.* Cambridge, MA: M.I.T. Press, 1972.

[9] S. I. Samoylenko, "Binoid error correcting codes," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 95–101, Jan. 1973.

[10] G. Benelli, C. Bianciardi, and V. Cappellini, "Some burst-error-correcting binoid codes," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 711–712, Nov. 1975.

[11] —, "Generalized Hamming codes for burst-error-correction," *Alta Freq.*, vol. 54, Nov. 1975.

[12] T. J. Lynch, "Performance measures for compressed and coded space telemetry systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-3, pp. 784–795, Sept. 1967.

[13] G. Benelli, V. Cappellini, and E. Del Re, "Integrated data communication systems with data compression and error correcting codes," in *Communication Systems and Random Process Theory* (NATO-ASI Series E, no. 25), J. K. Skwirzynski, Ed. Alphen aan den Rijn, The Netherlands: Sijthoff & Noordhoff, 1978.

[14] G. Benelli, "Influence of noise on some data compression algorithms," *Alta Freq.*, vol. 54, pp. 400–405, Nov. 1980.

[15] E. Del Re, "Bandpass signal filtering and reconstruction through minimum-sampling-rate digital processing," *Alta Freq.*, vol. 57, pp. 675–678, Sept. 1978.

[16] S. A. Gronemeyer and A. L. McBride, "MSK and offset QPSK modulation," *IEEE Trans. Commun.*, vol. COM-24, pp. 809–820, Aug. 1976.

[17] G. Benelli, V. Cappellini, E. Del Re, V. Ferrazzuolo, R. Pieroni, and I. Saltini, "Some importance sampling techniques with applications to radar systems," presented at the Int. Conf. Digital Signal Processing, Florence, Italy, Aug. 1978.

[18] K. S. Shanmugan and P. Balaban, "A modified Monte Carlo simulation technique for the evaluation of error rate in digital communication system," *IEEE Trans. Commun.*, vol. COM-28, pp. 1916–1924, Oct. 1980.

[19] M. Barducci, G. Benelli, V. Cappellini, and E. Del Re, "Digital simulation of integrated communication systems using data compression and error correcting codes," presented at the Int. Symp. Simulation, June 1977.

[20] G. Benelli, G. Borghi, V. Cappellini, E. Del Re, and S. Gueli, "Combined amplitude–phase modulation for a VHF communication link," presented at the IEEE Int. Conf. Commun., Denver, CO, June 1981.

[21] G. Benelli and R. Fantacci, "An integrated voice–data communication system for VHF links," *IEEE Trans. Commun.*, vol. COM-31, pp. 1304–1308, Dec. 1983.

Giuliano Benelli received the degree in physics from the University of Florence, Florence, Italy, in 1973.

In 1974 he taught channel encoding in the information theory course at the Scuola di Perfezionamento in Physics at the University of Florence. In 1975 he joined the Istituto de Elettronica of the University of Florence, first with a Research Fellowship and since 1981 as a Researcher. His main research interests are source and channel encoding, digital communications, and radar systems.

Vito Cappellini (M'68) was born in Pistoia, Italy, in 1938. He received the degree in electronic engineering from the Politecnico di Torino, Turin, Italy, in 1961.

In 1962 he was with the Fiat Company, Turin, and the Solvay Company, Rosignano Solvay, Italy, working on data processing and automatic control systems. In 1963 he began his research activity at the Centro Microonde, now the Istituto di Ricerca sulle Onde Elettromagnetiche (IROE) of the National Research Council (CNR), Florence, Italy, where he was mainly concerned, up to 1975, with ionospheric studies by means of satellites, data processing, digital telemetry, and communications. In 1975, he became a Full Professor of Electrical Communications at the University of Florence, Florence, being appointed Director of the Istituto di Elettronica of the Department of Engineering from 1977 to 1979. In 1981 he became Director of the IROE of CNR, Florence. He has published many papers in the fields of data processing, digital telemetry, and communications, and particularly in digital filtering, data compression, and error-correcting codes with applications in several areas such as communications, biomedicine, and remote sensing. He is also coauthor of the book *Digital Filters and Their Applications* (New York: Academic, 1978) and coeditor of *Digital Signal Processing* (Academic, 1980). His present research interests deal with digital image processing with applications to communications, biomedicine, remote sensing, and robotics.

Mr. Cappellini is a member of the European Association for Signal Processing (EURASIP), the Italian Electrotechnical and Electronic Association (AEI), ANIPLA, AICA, and AITA.

Enrico Del Re (M'78) was born in Florence, Italy, in 1947. He received the Dr.Ing. degree in electronic engineering from the University of Pisa, Pisa, Italy, in 1971.

After military service in the Italian Air Forces, involved in the maintenance and supervision of telecommunication and air traffic control equipment, from 1973 to 1975 he worked in the industry, engaged in the analysis of control circuits and systems for space applications. In 1975 he joined the Istituto di Elettronica of the University of Florence, Florence, first with a research fellowship and since 1978 as Associate Professor of Radio-Aids to Navigation. His main research interests are digital signal processing, digital communications, and radar signal processing.

Dr. Del Re is a member of the Italian Electrotechnical and Electronic Association (AEI) and the European Association for Signal Processing (EURASIP).