A Heuristic Discussion of Probabilistic Decoding*

ROBERT M. FANO†, FELLOW, IEEE

This is another in a series of invited tutorial, status and survey papers that are being regularly solicited by the PTGIT Committee on Special Papers. We invited Professor Fano to commit to paper his elegant but unelaborate explanation of the principles of sequential decoding, a scheme which is currently contending for a position as the most practical implementation to date of Shannon's theory of noisy communication channels.
—*Special Papers Committee.*

THE PURPOSE of this paper is to present a heuristic discussion of the probabilistic decoding of digital messages after transmission through a randomly disturbed channel. The adjective "probabilistic" is used to distinguish the decoding procedures discussed here from algebraic procedures[1] based on special structural properties of the set of code words employed for transmission.

In order to discuss probabilistic decoding in its proper frame of reference, we must first outline the more general problem of transmitting digital information through randomly disturbed channels, and review briefly some of the key concepts and results pertaining to it.[2] These key concepts and results were first presented by C. E. Shannon, in 1948,[3] and later sharpened and extended by Shannon and others. The first probabilistic decoding procedure of practical interest was presented by J. M. Wozencraft,[4] in 1957, and extended soon thereafter by B. Reiffen.[5] Equipment implementing this procedure has been built at Lincoln Laboratory[6] and is at present being tested in conjunction with telephone lines.

I. The Encoding Operation

We shall assume, for the sake of simplicity, that the information to be transmitted consists of a sequence of equiprobable and statistically independent binary digits. We shall refer to these digits as information digits, and to their rate, R, measured in digits per second, as the information transmission rate.

The complex of available communication facilities will be referred to as the transmission channel. We shall assume that the channel can accept as input any time function whose spectrum lies within some specified frequency band, and whose rms value and/or peak value are within some specified limits.

The information digits are to be transformed into an appropriate channel input, and must be recovered from the channel output with as small a probability of error as possible. We shall refer to the device that transforms the information digits into the channel input as the encoder, and to the device that recovers them from the channel output as the decoder.

The encoder may be regarded, without any loss of generality, as a finite-state device whose state depends, at any given time, on the last ν information digits input to it. This does not imply that the state of the device is uniquely specified by the last ν digits. It may depend on time as well, provided that such a time dependence is established beforehand and built into the decoder, as well as into the encoder. The encoder output is uniquely specified by the current state, and therefore is a function of the last ν information digits. We shall see that the integer ν, representing the number of digits on which the encoder output depends at any given time, is a critical parameter of the transmission process.

The encoder may operate in a variety of ways that depend on how often new digits are fed to it. The digits may be fed one at a time every $1/R$ seconds, or two at a time every $2/R$ seconds, and so forth. The limiting case in which the information digits are fed to the encoder in blocks of ν every ν/R seconds is of special interest and corresponds to the mode of operation known as block encoding. In fact, if each successive block of ν digits is fed to the encoder in a time that is short compared with $1/R$, the decoder output depends only on the digits of the last block, and is totally independent of the digits of the preceding blocks. Thus, the encoder output during each

* Received November 24, 1962. The work of this laboratory is supported in part by the U. S. Army, the Air Force Office of Scientific Research, and the Office of Naval Research.

†Department of Electrical Engineering and Research Laboratory of Electronics, M.I.T., Cambridge, Mass.

[1] W. W. Peterson, "Error-Correcting Codes," The M.I.T. Press, Cambridge, Mass., and John Wiley and Sons, Inc., New York, N. Y.; 1961.

[2] R. M. Fano, "Transmission of Information, The M.I.T. Press, Cambridge, Mass., and John Wiley and Sons, Inc., New York, N. Y.; 1961.

[3] C. E. Shannon, "A mathematical theory of communication," *Bell Sys. Tech. J.*, vol. 27, pp. 379–623; July, October, 1948.

[4] J. M. Wozencraft, "Sequential Decoding for Reliable Communications," Res. Lab. of Electronics, M.I.T., Cambridge, Mass., Technical Rept. 325; 1957. See also J. M. Wozencraft and B. Reiffen, "Sequential Decoding," The M.I.T. Press, Cambridge, Mass., and John Wiley and Sons, Inc., New York, N. Y.; 1961.

[5] B. Reiffen, "Sequential Encoding and Decoding for the Discrete Memoryless Channel," Res. Lab. of Electronics, M.I.T., Cambridge, Mass., Technical Rept. 374; 1960.

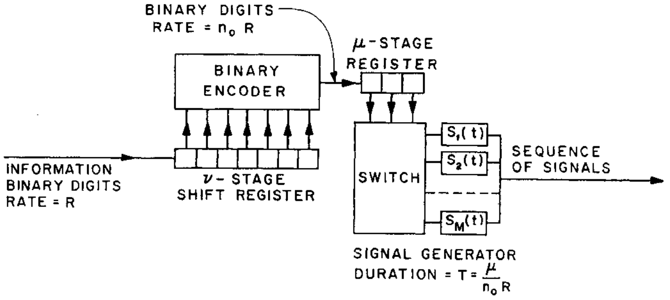
[6] K. M. Perry and J. M. Wozencraft, "SECO: A self-regulating error correcting coder-decoder," IRE Trans. on Information Theory, vol. IT-8, pp. 128–125; September, 1962.

time interval of length ν/R, corresponding to the transmission of one particular block of digits, is completely independent of the output during the time intervals corresponding to preceding blocks of digits. In other words, each block of ν digits is transmitted independently of all preceding blocks.

The situation is quite different when the information digits are fed to the encoder in blocks of size $\nu_0 < \nu$. Then the encoder output depends not only on the digits of the last block fed to the encoder, but also on $\nu - \nu_0$ digits of preceding blocks. Therefore, it is not independent of the output during the time interval corresponding to preceding blocks. As a matter of fact a little thought will indicate that the dependence of the decoder output on his own past extends to infinity in spite of the fact that its dependence on the input digits is limited to the last ν. For this reason, the mode of operation corresponding to $\nu_0 < \nu$ is known as sequential encoding. The distinction between block encoding and sequential encoding is basic to our discussion of probabilistic decoding.

The encoding operation, whether of the block or sequential type, is best performed in two steps, as illustrated in Fig. 1. The first step is performed by a binary encoder that generates n_0 binary digits per input information digit, where the integer n_0 is a design parameter to be selected in view of the rest of the encoding operation, and of the channel characteristics. The binary encoder is a finite-state device whose state depends on the last ν information digits fed to it, and possibly on time as discussed above. The dependence of the state on the information digits is illustrated in Fig. 1, by showing the ν information digits as stored in a shift register with serial input and parallel output. It can be shown that the operation of the finite-state encoder need not be more complex than a modulo-2 convolution of the input digits with a periodic sequence of binary digits of period equal to $n_0\nu$. A suitable periodic sequence can be constructed simply by selecting the $n_0\nu$ digits equiprobably and independently at random. Thus, the complexity of the binary encoder grows linearly with ν, and its design depends on the transmission channel only through the selection of the integers n_0 and ν.

The second part of the encoding operation is a straightforward transformation of the sequence of binary digits generated by the binary encoder into a time function that is acceptable by the channel. Because of the finite-state character of the encoding operation, the resulting time function must necessarily be a sequence of elementary time functions selected from a finite set. The elementary time functions are indicated in Fig. 1 as $S_1(t)$, $S_2(t)$, $\cdots$, $S_M(t)$, where M is the number of distinct elementary time functions, and T is their common duration. The generation of these elementary time functions may be thought of as being controlled by a switch, whose position is in turn set by the digits stored in a μ-stage binary register. The digits generated by the binary encoder are fed to this register μ at a time, so that each successive group of μ digits is transformed into one of the elementary signals. The number of distinct elementary signals M cannot exceed 2^μ, but it may be smaller. A value of M substantially smaller than 2^μ is used when some of the elementary signals are to be employed more often than others. For instance, with $M = 2$ and $\mu = 2$ we could make one of the 2 elementary signals occur 3 times as often as the other, by connecting 3 of the switch positions to one signal and the remaining one to the other.



R = TRANSMISSION RATE IN BITS/SEC

ν, μ, n_0 = POSITIVE INTEGERS

S(t) = TIME FUNCTION OF DURATION T

M = NUMBER OF DISTINCT $S(t) \leq 2^{\mu}$

Fig. 1—The encoding operation.

While the character of the transformation of binary digits into signals envisioned in Fig. 1 is quite general, the range of the parameters involved is limited by practical considerations. The number of distinct elementary signals M must be relatively small, and so must be the integer n_0. The values of M and n_0, as well as the forms of the elementary signals, must be selected with great care in view of the characteristics of the transmission channel. In fact, their selection results in the definition of the class of time functions that may be fed to the channel, and therefore, in effect, to a redefinition of the channel.[7] Thus, one faces a compromise between equipment complexity and degradation of channel characteristics.

Fig. 2 illustrates two choices of parameters and of elementary signals, which would be generally appropriate when no bandwidth restriction is placed on the signal and thermal agitation noise is the only disturbance present in the channel. In case a) each digit generated by the binary encoder is transformed into a binary pulse, while in case b) each successive block of 4 digits is transformed into a sinusoidal pulse 4 times as long, and of frequency proportional to the binary number spelled by the group of 4 digits. The example illustrated in Fig. 3 pertains instead to the case in which the signal bandwidth is so limited that the shortest pulse duration permitted is equal to the time interval corresponding to the transmission of 2 information digits. In this case the elementary signals are pulses of the shortest permissible duration, with 16 different amplitudes.

[7] J. Ziv, "Coding and decoding for time-discrete amplitude continuous memoryless channels," IRE Trans. on Information Theory, vol. IT-8, pp. 199–205; September, 1962.

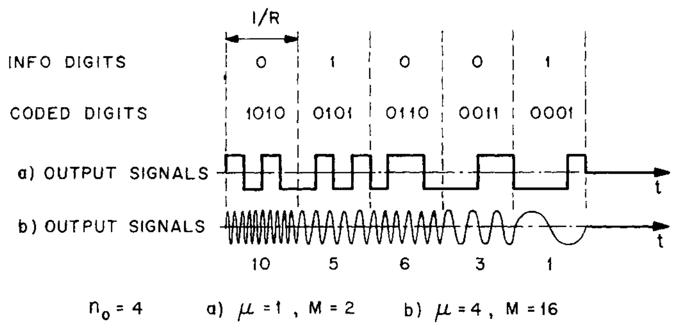


Fig. 2—Examples of encoding for a channel with unlimited band.

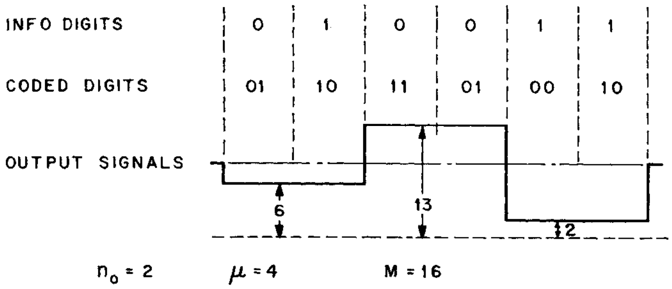


Fig. 3—Example of encoding for a band-limited channel.

These examples should make it clear that the encoding process illustrated in Fig. 1 includes, as special cases, the traditional forms of modulation employed in digital communication. What distinguishes the forms of encoding envisioned here from the traditional forms of modulation is the order of magnitude of the integer ν. In the traditional forms of modulation the value of ν is very small, often equal to 1 and very seldom greater than 5. Here instead we envision values of ν of the order of 50 or more. The reason for using large values of ν will become evident later on.

II. Channel Quantization

Let us suppose that the encoding operation has been fixed to the extent of having selected the duration and identities of the elementary signals. We must consider next how to represent the effect of the channel disturbances on these signals. Since most of our present detailed theoretical knowledge is limited to channels without memory, we shall limit our discussion to such channels. A channel without memory can be defined for our purpose as one whose output during each time interval of length T, corresponding to the transmission of an elementary signal, is independent of the channel input and output during preceding time intervals. This implies that the operation of the channel can be described within any such time interval without reference to the past or the transmission. We shall also assume that the channel is stationary in the sense that its properties do not change with time.

Let us suppose that the elementary signals are transmitted with probabilities $P(S_1)$, $P(S_2)$, $\cdots$, $P(S_M)$, and indicate by $S'(t)$ the channel output during the time interval corresponding to the transmission of a particular signal. The observation of the output $S'(t)$ changes the probability distribution over the ensemble of elementary signals, from the *a priori* distribution $P(S)$ to the *a posteriori* conditional distribution $P(S \mid S')$. The latter distribution can be computed, at least in principle, from the *a priori* distribution and the statistical characteristics of the channel disturbances. More precisely, we may regard the output $S'(t)$ as a point S' in a continuous space of suitable dimensionality. Then, if we indicate by $p(S' \mid S_k)$ the conditional probability density (assumed to exist) of the output S' for a particular input S_k, and have

$$p(S') = \sum_{k=1}^{M} P(S_k)p(S' \mid S_k) \tag{1}$$

the probability density of S' over all input signals, we obtain

$$P(S \mid S') = \frac{P(S)p(S' \mid S)}{p(S')}. \tag{2}$$

Knowing the *a posteriori* probability distribution $P(S \mid S')$ is equivalent, for our purposes, to knowing the output signal S'. In turn, this probability distribution depends on S' only through the ratios of the M probability densities $p(S' \mid S)$. Furthermore, these probability densities cannot be determined, in practice, with infinite precision. Thus, we must decide, either implicitly or explicitly, the tolerance within which the ratios of these probability densities are to be determined.

The effect of introducing such a tolerance is to lump together the output signals S' for which the ratios of the probability densities remain within the prescribed tolerance. Thus, we might as well divide the S' space into regions in which the ratios of the densities remain within the prescribed tolerance, and record only the identity of the particular region to which the output signal S' belongs.

Such a quantization of the output space S' is governed by considerations similar to those governing the choice of the input elementary signals, namely, equipment complexity and channel degradation. We shall not discuss this matter further, except for stressing again that such quantizations are unavoidable in practice; their net result is to substitute for the original transmission channel a new channel with discrete sets of possible inputs and outputs, and a correspondingly reduced transmission capability.[7]

III. Channel Capacity

It is convenient at this point to change our terminology to that commonly employed in connection with discrete channels. We shall refer to the set of elementary input signals as the input alphabet, and to the individual signals as input symbols. Similarly, we shall refer to the set of regions in which the channel output space has been divided as the output alphabet, and to the individual regions as output symbols. The input and output alphabets will be indicated by X and Y, respectively, and particular symbols belonging to them will be indicated by x and y. Thus, the transmission channel is completely described by the alphabets X and Y, and by the set of conditional probability distributions $P(y \mid x)$.

We have seen that the net effect of the reception of a symbol y is to change the *a priori* probability distribution $P(x)$ into the *a posteriori* probability distribution

$$P(x \mid y) = \frac{P(x)P(y \mid x)}{P(y)} = \frac{P(x, y)}{P(y)}, \tag{3}$$

where $P(x, y)$ is the joint probability distribution of input and output symbols. Thus, the information provided by a particular output symbol y about a particular input symbol x is defined as

$$I(x; y) = \log \frac{P(x \mid y)}{P(x)} = \log \frac{P(y \mid x)}{P(y)} = \log \frac{P(x, y)}{P(x)P(y)}. \tag{4}$$

We shall see that this measure of information and its average value over the input and/or output alphabets play a central role in the problem under discussion.

It is interesting to note that $I(x; y)$ is a symmetrical function of x and y, so that the information provided by a particular y about a particular x is the same as the information provided by x about y. In order to stress this symmetry property, $I(x; y) = I(y; x)$ is often referred to as the mutual information between x and y. In contrast,

$$I(x) = \log \frac{1}{P(x)} \tag{5}$$

is referred to as the self-information of x. This name follows from the fact that, for a particular symbol pair $x = x_k$, $y = y_i$, $I(x_k; y_i)$ becomes equal to $I(x_k)$ when $P(x_k \mid y_i) = 1$, that is, when the output symbol y_i uniquely identifies x_k as the input symbol. Thus, $I(x_k)$ is the amount of information that must be provided about x_k in order to uniquely identify it, and as such is an upper bound to the value of $I(x_k; y)$.

In the particular case of an alphabet with L equiprobable symbols, the self-information of each symbol is equal to $\log L$. The information is measured in bits when base-2 logarithms are used in the expressions above. Thus the self-information of the symbols of a binary equiprobable alphabet is equal to 1 bit.

Let us suppose that the input symbol is selected from the alphabet X with probability $P(x)$. The average, or expected value, of the mutual information between input and output symbols is, then,

$$I(X; Y) = \sum_{XY} P(x, y)I(x; y). \tag{6}$$

This quantity depends on the input probability distribution $P(x)$ and on the characteristics of the channel represented by the conditional probability distributions $P(y \mid x)$. Thus, its value for a given channel depends on the probability distribution $P(x)$ alone.

The channel capacity is defined as the maximum value of $I(X; Y)$ with respect to $P(x)$, that is,

$$C = \operatorname*{Max}_{P(x)} I(X; Y). \tag{7}$$

It can be shown[8] that if a source that generates sequences of x symbols is connected to the channel input, the average amount of information per symbol provided by the channel output about the channel input cannot exceed C, regardless of the statistical characteristics of the source.

IV. Error Probability for Block Encoding

Let us consider now the special case of block encoding, and suppose that a block of ν information digits is transformed by the encoder into a sequence of N elementary signals, that is, into a sequence of N input symbols. Since the information digits are, by assumption, equiprobable and independent of one another, it takes an amount of information equal to log 2 (1 bit) to identify each of them. Thus, the information transmission rate per channel symbol is given by

$$R = \frac{\nu}{N} \log 2. \tag{8}$$

(Note that the same symbol is used to indicate the information transmission rate, whether per channel symbol or per unit time.)

The maximum amount of information per symbol which the channel output can provide about the channel input is equal to C, the channel capacity. It follows that we cannot expect to be able to transmit the information digits with any reasonable degree of accuracy at any rate $R > C$. Shannon's fundamental theorem asserts, furthermore, that for any $R < C$ the probability of erroneous decoding of a block of ν digits can be made as small as desired by employing a sufficiently large value of ν and a correspondingly large value of N. More precisely, it is possible[9] to achieve a probability of error per block bounded by

$$P_e < 2^{-\nu(\alpha/R)+1}, \tag{9}$$

where α is independent of ν and varies with R as illustrated schematically in Fig. 4. Thus, for any $R < C$, the probability of error decreases exponentially with increasing ν.

It is clear from (9) that the probability of error is controlled primarily by the product of ν and α/R, the latter quantity being a function of R alone for a given channel. Thus, the same probability of error can be obtained with a small value of ν and relatively small value of R, or with a value of R close to C and a correspondingly larger value of ν. In the first situation, which corresponds to the traditional forms of modulation, the encoding and decoding equipment is relatively simple because of the small value of ν, but the channel is not utilized efficiently. In the second situation, on the contrary, the channel is efficiently utilized, but the relatively large value of ν implies that the terminal equipment must be substantially more complex. Thus, we are faced with a compromise between efficiency of channel utilization and complexity of terminal equipment.

[8] Fano, *op. cit.*, see Sec. 5.2.

[9] Fano, *op. cit.*, see ch. 9.

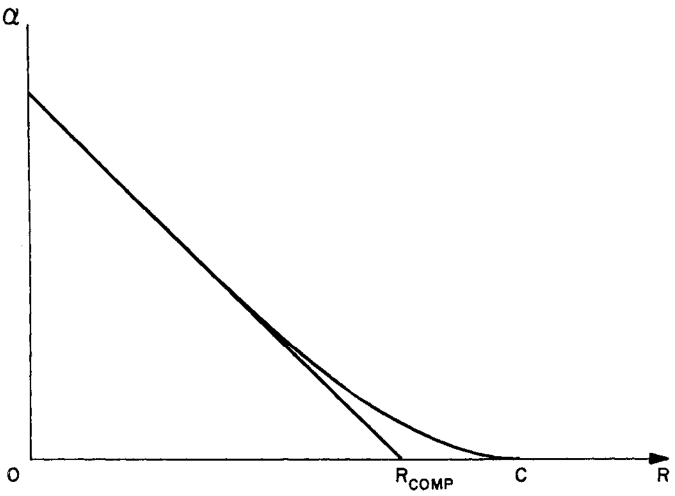


Fig. 4—Relation between the exponential coefficient α and the information transmission rate R in (9).

It was pointed out in Section I that the operation to be performed by the binary encoder is relatively simple, namely, the convolution of the input information digits with a periodic sequence of binary digits of period equal to $n_0\nu$. Thus, roughly speaking, the complexity of the encoding equipment grows linearly with ν. On the other hand, the decoding operation is substantially more complex, both conceptually and in terms of the equipment required to perform it. The rest of this paper is devoted to it.

V. Probabilistic Block Decoding

We have seen that in the process of block encoding each particular sequence of ν information digits is transformed by the encoder into a particular sequence of N channel-input symbols. We shall refer to any such sequence of input symbols as a code word, and we shall indicate by u_k the code word corresponding to the sequence of information digits which spells k in the binary number system. The sequence of N output symbols resulting from an input code word will be indicated by v.

The probability that a particular code word u will result in a particular output sequence v is given by

$$P(v \mid u) = \prod_{j=1}^{N} [P(y \mid x)]_j, \tag{10}$$

where the subscript j indicates that the value of the conditional probability is evaluated for the input and output symbols that occupy the jth positions in u and v. On the other hand, since all sequences of information digits are transmitted with the same probability, the *a posteriori* probability of any particular code word u after the reception of a particular output sequence v is given by

$$P(u \mid v) = \frac{P(v \mid u)P(u)}{\sum_{U} P(v \mid u)P(u)} = 2^{-\nu} \frac{P(v \mid u)}{P(v)}. \tag{11}$$

Thus, the code word that *a posteriori* is most probable for a particular output v is the one that maximizes the conditional probability $P(v \mid u)$ given by (10). We can conclude that in order to minimize the probability of error the decoder should select the code word with the largest probability $P(v \mid u)$ of generating the sequences v output from the channel.

While the specification of the optimum decoding procedure is straightforward, its implementation presents very serious difficulties for any sizable value of ν. In fact, there is no general procedure for determining the code word corresponding to the largest value of $P(v \mid u)$ without having to evaluate this probability for most of the 2^ν possible code words. Clearly, the necessary amount of computation grows exponentially with ν and very quickly becomes prohibitively large. However, if we do not insist on minimizing the probability of error, we may take advantage of the fact that, if the probability of error is to be very small, the *a posteriori* most probable code word must be almost always substantially more probable than all other code words. Thus, it may be sufficient to search for a code word with a value of $P(v \mid u)$ larger than some appropriate threshold, and take a chance on the possibility that there be other code words with even larger values, or that the value for the correct code word be smaller than the threshold.

Let us consider, then, what might be an appropriate threshold. Let us suppose that for a given received sequence v there exists a code word u_k for which

$$P(u_k \mid v) \geq \sum_{i \neq k} P(u_i \mid v), \tag{12}$$

where the summation extends over all the other $2^\nu - 1$ code words. Then u_k must be the *a posteriori* most probable code word. The condition expressed by (12) can be rewritten, with the help of (11), as

$$P(v \mid u_k) \geq \sum_{i \neq k} P(v \mid u_i). \tag{13}$$

The value of $P(v \mid u_k)$ can be readily computed with the help of (10). However, we are still faced with the problem of evaluating the same conditional probability for all of the other code words. This difficulty can be circumvented by using an approximation related to the random-coding procedure employed in deriving (9).

In the process of random coding each code word is constructed by selecting its symbols independently at random according to some appropriate probability distribution $P_0(x)$. The right-hand side of (9) is actually the average value of the probability of error over the ensemble of code-word sets so constructed. This implies, incidentally, that satisfactory code words can be obtained in practice by following such a random construction procedure.

Let us assume that the code words under consideration have been constructed by selecting the symbols independently at random according to some appropriate probability distribution $P_0(x)$. It would seem reasonable then to substitute for the right-hand side of (13) its average value over the ensemble of code-word sets constructed in the same random manner. In such an ensemble of code-word sets, the probability $P_0(u)$ that any particular input sequence u be chosen as a code word is

$$P_0(u) = \prod_{j=1}^{N} [P_0(x)]_j, \tag{14}$$

where the subscript j indicates that $P_0(x)$ is evaluated for the jth symbol of the sequence u. Thus, the average value of the right-hand side of (12), with the help of (10), is

$$(2^{\nu} - 1) \sum_{U} P_0(u)P(v \mid u) = (2^{\nu} - 1) \prod_{j=1}^{N} [P_0(y)]_j, \tag{15}$$

where U is the set of all possible input sequences, and

$$P_0(y) = \sum_{X} P_0(x)P(y \mid x) \tag{16}$$

is the probability distribution of the output symbols when the input symbols are transmitted independently with probability $P_0(x)$. Then substituting the right-hand side of (15) for the right-hand side of (13) and expressing $P(v \mid u_k)$ as in (10) yields

$$\prod_{j=1}^{N} \left[\frac{P(y \mid x)}{P_0(y)}\right]_j \geq 2^{\nu} - 1. \tag{17}$$

Finally, approximating $2^{\nu} - 1$ by 2^{ν} and taking the logarithm of both sides yields

$$\sum_{j=1}^{N} \left[\log \frac{P(y \mid x)}{P_0(y)}\right]_j \geq NR, \tag{18}$$

where R is the transmission rate per channel symbol defined by (8).

The threshold condition expressed by (18) can be given a very interesting interpretation. The jth term of the summation is the mutual information between the jth output symbol and the jth input symbol, with the input symbols assumed to occur with probability $P_0(x)$. If the input symbols were statistically independent of one another, the sum of these mutual informations would be equal to the mutual information between the output sequence and the input sequence. Thus, (18) states that the channel output can be safely decoded into a particular code word if the mutual information that it provides about the code word, evaluated as if the N input symbols were selected independently with probability $P_0(x)$, exceeds the amount of information transmitted per code word.

It turns out that the threshold value on the right-hand side of (18) is not only a reasonable one, as indicated by our heuristic arguments, but the one that minimizes the average probability of error for threshold decoding over the ensemble of randomly constructed code-word sets. This has been shown by C. E. Shannon in an unpublished memorandum. The bound on the probability of error obtained by Shannon is of the form of (9); however, the value of α is somewhat smaller than that obtained for optimum decoding. Shannon assumes in his derivation that an error occurs whenever (17) either is satisfied for any code word other than the correct one or it is not satisfied for the correct code word.

The probability of error for threshold decoding, although larger than for optimum decoding, is still bounded as in (9). This fact encourages us to look for a search procedure that will quickly reject any code word for which (17) is not satisfied, and thus converge relatively quickly on the code word actually transmitted. We observe, on the other hand, that, even if we could reject an incorrect code word after evaluating (17) over some small but finite fraction of the N symbols, we would still be faced with an amount of computation that would grow exponentially with ν. In order to avoid this exponential growth, we must arrange matters in such a way as to be able to eliminate large subsets of code words, by evaluating the left-hand side of (17) over some fraction of a single code word. This implies that the code words must possess the kind of tree structure which results from sequential encoding, as discussed in the next section.

It is just the realization of this fact that led J. M. Wozencraft to the development of his sequential decoding procedure in 1957. Other decoding procedures, both algebraic[1] and probabilistic,[10] have been developed since, which are of practical value in certain special cases. However, sequential decoding remains the only known procedure that is applicable to all channels without memory. As a matter of fact, there is reason[11] to believe that some modified form of sequential decoding may yield satisfactory results in conjunction with a much broader class of channels.

VI. Sequential Decoding

The rest of this paper is devoted to a heuristic discussion of a sequential decoding procedure recently developed by the author. This procedure is similar in many respects to that of Wozencraft,[4-6] but it is conceptually simpler and therefore it can be more readily explained and evaluated. An experimental comparison of the two procedures is in progress at Lincoln Laboratory, M. I. T., Lexington, Mass. A detailed analysis of the newer procedure will be presented in a forthcoming paper.

Let us reconsider in greater detail the structure of the encoder output in the case of sequential encoding, that is, when the information digits are fed to the encoder in blocks of size ν_0 (in practice ν_0 is seldom larger than 3 or 4). The encoder output, during the time interval corresponding to a particular block, is selected by the digits of the block from a set of 2^{ν_0} distinct sequences of channel input symbols. The particular set of sequences from which the output is selected is specified, in turn, by the $\nu - \nu_0$

[10] R. G. Gallager, "Low density parity-check codes," IRE Trans. on Information Theory, vol. IT-8, pp. 21-28; January, 1962.

[11] R. G. Gallager, "Sequential Decoding for Binary Channels with Noise and Synchronization Errors," Lincoln Lab., M.I.T., Lexington, Mass., Rept. No. 25G-2; 1961.

information digits preceding the block in question. Thus, the set of possible outputs from the encoder can be represented by means of a tree with 2^{ν_0} branches stemming from each node. Each successive block of ν_0 information digits causes the encoder to move from one node to the next one, along the branch specified by the digits of the block.

The two trees shown in Fig. 5 correspond to the two examples illustrated in Figs. 2(b) and 3. The first example yields a binary tree ($\nu_0 = 1$), while the second example yields a quaternary tree ($\nu_0 = 4$).

In summary, the encoding operation can be represented in terms of a tree in which the information digits select at each node the branch to be followed. The path in the tree resulting from the successive selections constitutes the encoder output. This is equivalent to saying that each block of ν_0 digits fed to the encoder is represented for transmission by a sequence of symbols selected from a set of 2^{ν_0} distinct sequences, but the particular set from which the sequence is selected depends on the preceding $\nu - \nu_0$ information digits. Thus, the channel output, during the time interval corresponding to a block of ν_0 information digits, provides information not only about these digits but also about the preceding $\nu - \nu_0$ digits.

The decoding operation may be regarded as the process of determining, from the channel output, the path in the tree followed by the encoder. Suppose, to start with, that the decoder selects at each node the branch which *a posteriori* is most probable, on the basis of the channel output during the time interval corresponding to the transmission of the branch. If the channel disturbance is such that the branch actually transmitted does not turn out to be the most probable one, the decoder will make an error, thereby reaching a node that does not lie on the path followed by the encoder. Thus, none of the branches stemming from it will appear as a likely channel input. If by accident one branch does appear as a likely input, the same situation will arise with respect to the branches stemming from the node in which it terminates, and so forth and so on. This rough notion can be made more precise.

Let us suppose that the branches of the tree are constructed, as in the case of block encoding, by selecting symbols independently at random according to some appropriate probability distribution $P_0(x)$. This is accomplished in practice by selecting equiprobably at random the $n_0\nu$ binary digits specifying the periodic sequence with which the sequence of information digits is convolved, and by properly arranging the connections of switch positions to the elementary signals in Fig. 1. Then, as in the case of threshold block decoding, the decoder, as it moves along a path in the tree, evaluates the quantity

$$I_N = \sum_{j=1}^{N} \left[\log \frac{P(y \mid x)}{P_0(y)} \right]_j, \tag{19}$$

where y in the jth term of the summation is the jth symbol output from the channel, and x in the same term is the jth symbol along the path followed by the decoder.

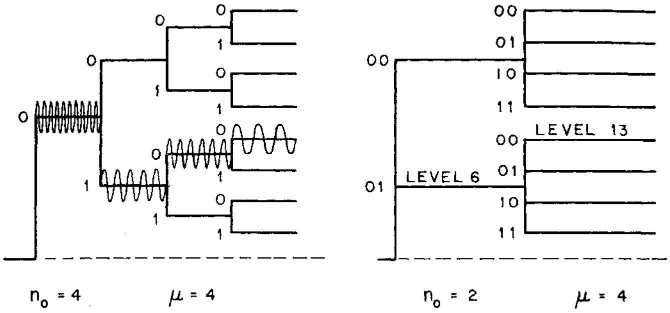


Fig. 5—Encoding trees corresponding to the examples of Figs. 2(b) and 3.

As long as the path followed by the decoder coincides with that followed by the encoder I_N can be expected to remain greater than NR (R, the information transmission rate per channel symbol, is still equal to the number of channel symbols divided by the number of corresponding information digits, but it is no longer given by (8)). However, once the decoder has made a mistake and has thereby arrived at a node that does not lie on the path followed by the encoder, the terms of I_N corresponding to branches beyond that node are very likely to be smaller than R. Thus I_N must eventually become smaller than NR, thereby indicating that an error must have occurred at some preceding node. It is clear that in such a situation the decoder should try to find the place where the mistake has occurred in order to get back on the correct path. It would be desirable, therefore, to evaluate for each node the relative probability that a mistake has occurred there.

VII. Probability of Error along a Path

Let us indicate by N the order number of the symbol preceding some particular node, and by N_0 the order number of the last output symbol. Since all paths in the tree are *a priori* equiprobable, their *a posteriori* probabilities are proportional to the conditional probabilities $P(v \mid u)$, where u is the sequence of symbols corresponding to a particular path, and v is the resulting sequence of output symbols. This conditional probability can be written in the form

$$P(v \mid u) = \prod_{j=1}^{N} [P(y \mid x)]_j \prod_{j=N+1}^{N_0} [P(y \mid x)]_j. \tag{20}$$

The first factor on the right-hand side of (20) has the same value for all the paths that coincide over the first N symbols. The number of such paths, which differ in some of the remaining $N_0 - N$ symbols, is

$$m = 2^{(N_0 - N)R/\log 2}. \tag{21}$$

As in the case of block decoding, it is impractical to compute the second factor on the right-hand side of (20) for each of these paths. We shall again circumvent this difficulty by averaging over the ensemble of randomly constructed trees. By analogy with the case of threshold block decoding, we obtain

$$P_0(v \mid u) = \prod_{j=1}^{N} [P(y \mid x)]_j \prod_{j=N+1}^{N_0} [P_0(y)]_j, \qquad (22)$$

where $P_0(y)$ is given by (16).

Let P_N be the probability that the path followed by the encoder is one of the $m - 1$ paths that coincide with the one followed by the decoder over the first N symbols, but differ from it thereafter. By approximating $m - 1$ with m, we have

$$\begin{aligned} P_N &= K_1 2^{(N_0 - N)R/\log 2} \prod_{j=1}^{N} [P(y \mid x)]_j \prod_{j=N+1}^{N_0} [P_0(y)]_j \qquad (23) \\ &= K_2 2^{-NR/\log 2} \prod_{j=1}^{N} \left[\frac{P(y \mid x)}{P_0(y)} \right]_j, \end{aligned}$$

where K_1 and K_2 are proportionality constants. Finally, taking the logarithm of both sides of (23) yields

$$\log P_N = \log K_2 + \sum_{j=1}^{N} \left[\log \frac{P(y \mid x)}{P_0(y)} - R\right]_j. \qquad (24)$$

The significance of (24) is best discussed after rewriting it in terms of the order number of the nodes along the path followed by the decoder. Let us indicate by N_b the number of channel symbols per branch (assumed for the sake of simplicity to be the same for all branches) and by n the order number of the node following the Nth symbol. Then (24) can be rewritten in the form

$$\log P_n = \log K_2 + \sum_{k=1}^{n} \lambda_k, \qquad (25)$$

where

$$\lambda_k = \sum_{j=(k-1)N_b+1}^{kN_b} \left[\log \frac{P(y \mid x)}{P_0(y)} - R\right]_j \qquad (26)$$

is the contribution to the summation in (24) of the kth branch examined by the decoder. Finally, we can drop the constant from (25) and focus our attention on the sum

$$L_n = \sum_{k=1}^{n} \lambda_k, \qquad (27)$$

which increases monotonically with the probability P_n.

A typical behavior of L_n as a function of n is illustrated in Fig. 6. The value of λ_k is normally positive, in which case the probability that an error has been committed at some particular node is greater than the probability that an error has been committed at the preceding node. Let us suppose that the decoder has reached the nth node, and the value of λ_{n+1} corresponding to the *a posteriori* most probable branch stemming from it is positive. Then, the decoder should proceed to examine the branches stemming from the following node, on the assumption that the path is correct up to that point. On the other hand, if the value of λ_{n+1} is negative, the decoder should assume that an error has occurred and examine other branches stemming from preceding nodes in order of relative probability.

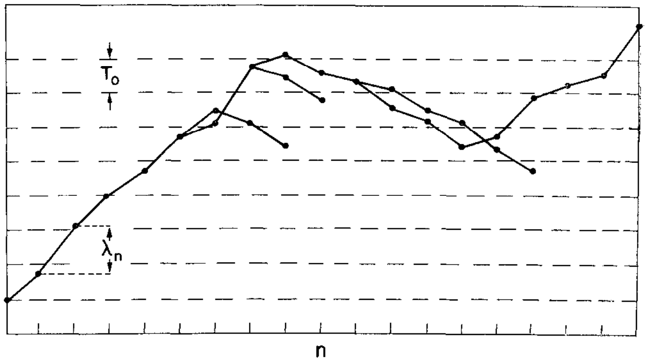


Fig. 6—Behavior of the likelihood function L_n along various tree paths. The continuous curve corresponds to the correct path.

VIII. A Specific Decoding Procedure

It turns out that the process of searching other branches can be considerably simplified if we do not insist on searching them in exact order of probability. A procedure is described below in which the decoder moves forward or backward from node to node depending on whether the value of L at the node in question is larger or smaller than a threshold T. The value of T is increased or decreased in steps of some appropriate magnitude T_0 as follows. Let us suppose that the decoder is at some node of order n, and that it attempts to move forward by selecting the most probable branch among those still untried. If the resulting value of L_{n+1} exceeds the threshold T, the branch is accepted and T is reset to the largest possible value not exceeding L_{n+1}. If, instead, L_{n+1} is smaller than T, the decoder rejects the branch and moves back to the node of order $n - 1$. If $L_{n-1} \geq T$, the decoder attempts again to move forward by selecting the most probable branch among those not yet tried, or, if all branches stemming from that node have already been tried, it moves back to the node of order $n - 2$. The decoder moves forward and backwards in this manner until it is forced back to a node for which the value L is smaller than the current threshold T.

When the decoder is forced back to a node for which L is smaller than the current threshold, all of the paths stemming from that node must contain at least a node for which L falls below the threshold. This situation may arise because of a mistake at that node or at some preceding node, as illustrated in Fig. 6 by the first curve branching off above the correct curve. It may also result from the fact that, because of unusually severe channel disturbances, the values of L along the correct path reach a maximum and then decrease to a minimum before rising again, as illustrated by the main curve in Fig. 6. In either case, the threshold must be reduced by T_0 in order to allow the decoder to proceed.

After the threshold has been reduced, the decoder attempts again to move forward by selecting the most probable branch, just as if it had never gone beyond the node at which the threshold had to be reduced. This leads the decoder to retrace all of the paths previously examined to see whether L remains above the new threshold along any one of them. Of course, T cannot be allowed to increase while the decoder is retracing any one of these paths, until it reaches a previously unexplored branch. Otherwise, the decoder would keep retracing the same path over and over again.

If L remains above the new threshold along the correct path, the decoder will be able to continue beyond the point at which it was previously forced back, and the threshold will be permitted to rise again, as discussed above. If, instead, L still falls below the reduced threshold at some node of the correct path, or an error has occurred at some preceding node for which L is smaller than the reduced threshold, the threshold will have to be further reduced by T_0. This process is continued until the threshold becomes smaller than the smallest value of L along the correct path, or smaller than the value of L at the node at which the mistake has taken place.

The flow chart of Fig. 7 describes the procedure more precisely than can be done in words. Let us suppose that the decoder is at some node of order n. The box at the extreme left of the chart examines the branches stemming from that node and selects the one that ranks ith in order of decreasing *a posteriori* probability. (The value of λ for this branch is indicated in Fig. 7 by the subscript $i(n)$, and the integer $i(n)$ is assumed to be stored for future use for each value of n. The number of branches is $b = 2^{\nu_0}$. Thus $1 \leq i(n) \leq b$.) Next, the value L_{n+1} is computed by adding L_n and $\lambda_{i(n)}$. The value of L_n may be needed later, if the decoder is forced back to the nth node, and therefore it must be stored or recomputed when needed. For the sake of simplicity, the chart assumes that L_n is stored for each value of n.

The chart is self-explanatory beyond this point except for the function of the binary variable F. This variable is used to control a gate that allows or prevents the threshold from increasing, the choice depending on whether $F = 0$ or $F = 1$, respectively. Thus, F must be set equal to 0 when the decoder selects a branch for the first time, and equal to 1 when the branch is being retraced after a reduction of threshold. The value of F is set equal to 1 each time a branch is rejected; it is reset equal to 0 before a new branch is selected only if $T \leq L_n < T + T_0$ for the node to which the decoder is forced back. The value F is reset equal to 0 after a branch is accepted if $T \leq L_{n+1} < T + T_0$ for the node at which the branch terminates. It can be checked that, after a reduction of threshold, F remains equal to 1 while a path is being retraced, and it is reset equal to 0 at the node at which the value of L falls below the previous threshold.

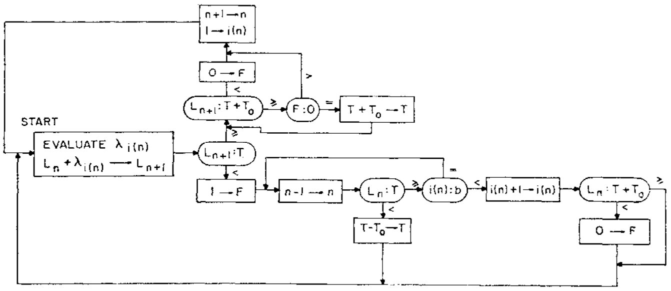


Fig. 7—Flow charge of the sequential-decoding procedure.
$1 \rightarrow F$ stands for: set F equal to 1
$L_n + \lambda_{i(n)} \rightarrow L_{n+1}$ stands for: set L_{n+1} equal to $L_n + \lambda_{i(n)}$
$n + 1 \rightarrow n$ stands for: substitute $n + 1$ for n(increase n by one)
$i(n) + 1 \rightarrow i(n)$ stands for: substitute $i(n) + 1$ for $i(n)$ (increase $i(n)$ by one)
$T + T_0 \rightarrow T$ stands for: substitute $T + T_0$ for T (increase T by T_0)
L_{n+1}: T stands for: compare L_{n+1} and T; follow path marked $\geq$, if $L_{n+1} \geq T$.

IX. Evaluation of the Procedure

The performance of the sequential decoding procedure outlined in the preceding section has been evaluated analytically for all discrete memoryless channels. The details of the evaluation and the results will be presented in a forthcoming paper. The general character of these results and their implications are discussed below. The most important characteristics of a decoding procedure are its complexity, the resulting probability of error per digit, and the probability of decoding failure. We shall define and describe these characteristics in order.

The notion of complexity actually consists of two related but separate notions: the amount of equipment required to carry out the decoding operation, and the speed at which the equipment must operate. Inspection of the flow chart shown in Fig. 7 indicates that the necessary equipment consists primarily of that required to generate the possible channel inputs, namely, a replica of the encoder, and that required to store the channel output and the information digits decoded. All other quantities required in the decoding operation can be either computed from the channel output and the information digits decoded, or stored in addition to them, if this turns out to be more practical. In Section I we found that the complexity of the encoding equipment increases linearly with the encoder memory ν, since the binary encoder must convolve two binary sequences of lengths proportional to ν. The storage requirements will be discussed in conjunction with the decoding failures.

The speed at which the decoding equipment must operate is not the same for all of its parts. However, it seems reasonable to measure the required speed in terms of the average number, $\bar{n}$, of branches which the decoder must examine per branch transmitted. A very conservative upper bound to $\bar{n}$ has been obtained which has the following properties. For any given discrete channel without memory, there exists a maximum information transmission rate for which the bound to $\bar{n}$ remains finite for messages of unlimited length. This maximum rate is given by

$$R_{\text{comp}} = \underset{P_0(x)}{\text{Max}} \{-\log \sum_Y [\sum_X P_0(x) \sqrt{P(y \mid x)}]^2\}. \qquad (28)$$

Then, for any transmission rate $R < R_{\text{comp}}$, the bound on $\bar{n}$ is not only finite but also independent of ν. This implies that the average speed at which the decoding equipment has to operate is independent of ν.

The maximum rate given by (28) bears an interesting relation to the exponential factor α in the bound, given by (9), to the error probability for optimum block decoding. As shown in Fig. 4, the curve of α vs R, for small values of R, coincides with a straight line of slope -1. This straight line intersects the R axis at the point $R = R_{\text{comp}}$. Clearly, $R_{\text{comp}} < C$. The author does not know of any channel for which R_{comp} is smaller than $\frac{1}{2}$ C, but no definite lower bound to R_{comp} has yet been found.

Next, let us turn our attention to the two ways in which the decoder may fail to reproduce the information digits transmitted. In the decoding procedure outlined above no limit is set on how far back the decoder may go in order to correct an error. In practice, however, a limit is set by the available storage capacity. Thus, decoding failures will occur whenever the decoder proceeds so far along an incorrect path that, by the time it gets back to the node where the error was committed, the necessary information has already been dropped from storage. Any such failure is immediately recognized by the decoder because it is unable to perform the next operation specified by the procedure.

The manner in which such failures are handled in practice depends on whether or not a return channel is available. If a return channel is available, the decoder can automatically ask for a repeat.[12] If no return channel is available, the stream of information digits must be broken into segments of appropriate length and a fixed sequence of $\nu - \nu_0$ digits must be inserted between segments. In this manner, if a decoding failure occurs during one segment, the rest of the segment will be lost but the decoder will start operating again at the beginning of the next segment.

The other type of decoding failure consists of digits erroneously decoded which cannot be corrected, regardless of the amount of storage available to the decoder. These errors are inherently undetectable by the decoder, and therefore do not stop the decoding operation. They arise in the following way.

The decoder, in order to generate the branches that must be examined, feeds the information digits decoded to a replica of the encoder. As discussed in Section VI, the set of branches stemming from a particular node is specified by the last $\nu - \nu_0$ information digits. Then, let us suppose that the decoder is moving forward along an incorrect path and that it generates, after a few incorrect digits, a sequence of $\nu - \nu_0$ information digits that happen to coincide with those transmitted. This is a very improbable event because the decoder is usually forced back long before it can generate that many digits. However, it can indeed happen if the channel disturbance is sufficiently severe during the time interval involved. After such an event, the replica of the encoder (which generates the branches to be examined) becomes completely free of incorrect digits, and therefore the decoding operation proceeds just as if the correct path had been followed all the time. Thus, the intervening errors will not be corrected. As a matter of fact, if the decoder were forced back to the node where the first error was committed, it would eventually take again the same incorrect path.

The resulting probability of error per digit decoded is bounded by an expression similar to (9). However, the exponential factor α is larger than for block encoding, although, of course, it vanishes for $R = C$. This fact may be explained heuristically by noting that the dependence of the encoder output on its own past extends beyond the symbols corresponding to the last ν information digits. Thus, we might say that, for the same value of ν, the effective constraint length is larger for sequential encoding than for block encoding.

Finally, let us consider further the decoding failures mentioned above. Since these decoding failures result from insufficient storage capacity, we must specify more precisely the character of the storage device to be employed. Suppose that the storage device is capable of storing the channel output corresponding to the last n branches transmitted. Then a decoding failure occurs whenever the decoder is forced back n nodes behind the branch being currently transmitted. This is equivalent to saying that the decoder is forced to make a final decision on each information digit within a fixed time after its transmission. Any error in this final decision, other than errors of the type discussed above, will stop the entire decoding operation. No useful bound could be obtained to the probability of occurrence of the decoding failures resulting from this particular storage arrangement.

Next, let us suppose that the channel output is stored on a magnetic tape, or similar buffer device, from which the segments corresponding to successive branches can be individually transferred to the decoder upon request. Suppose also that the internal memory of the decoder is limited to n branches. Then, a decoding failure occurs whenever the decoder is forced back n branches from the farthest one ever examined, regardless of how far back this branch is from the one being currently transmitted.

Let us indicate by k the order number of the last branch dropped from the decoder's internal memory. There are two distinct situations in which the decoder may be

[12] J. M. Wozencraft and M. Horstein, "Coding for two-way channels," in "Information Theory, Fourth London Symposium," C. Cherry, Ed., Butterworths Scientific Publications, London, England, p. 11; 1961

forced back to this branch after having examined a branch of order $k + n$. The value of L along the correct path falls below L_k at some node of order equal to, or larger than, $k + n$; or it falls below some threshold $T \leq L_k$ at some earlier node, and there exists an incorrect path, stemming from the node of order k, over which the value L remains above T up to the node of order $k + n$.

An upper bound to the probability of occurrence of these events can be readily found. It is similar to (9), with $\nu = n\nu_0$, and with a value of α approximately equal to that obtained for threshold block decoding.

X. Conclusion

The main characteristic of sequential decoding that makes it particularly attractive in practice is that the complexity of the necessary equipment grows only linearly with ν, while the required speed of operation is independent of ν. Thus, it is economically feasible to use values of ν sufficiently large to yield a negligibly small probability of error for transmission rates relatively close to channel capacity.[6]

Another important feature of sequential decoding is that its mode of operation depends very little on the channel characteristics, and therefore most of the equipment can be used in conjunction with a large variety of channels.

Finally, it should be stressed that sequential decoding is in essence a search procedure of the hill-climbing type. It can be used, in principle, to search any set of alternatives represented by a tree in which the branches stemming from different nodes of the same order are substantially different from one another.

General Results in the Mathematical Theory of Random Signals and Noise in Nonlinear Devices*

H. B. SHUTTERLY†, MEMBER, IEEE

Summary—**An analysis is made of the output resulting from passing signals and noise through general zero memory nonlinear devices. New expressions are derived for the output time function and autocorrelation function in terms of weighted averages of the nonlinear characteristic and its derivatives. These expressions are not restricted to Gaussian noise and apply to any nonlinearity having no more than a finite number of discontinuities. The method of analysis used is heuristic.**

I. Introduction

RICE,[1] Bennett,[2] Middleton,[3,4] Campbell[5] and Price[6] have determined the output autocorrelation function and power spectrum for sinusoidal signals and Gaussian noise through a number of specific nonlinear characteristics, principally the half-wave ν-th law rectifier. Most of this work has been done using the transform method described by Rice. In this paper a real-plane method of analysis is used and expressions for the output time and autocorrelation functions are obtained which are applicable to general nonlinear devices with general inputs.

The organization of the paper is as follows: Section II states the basic results and presents one illustrative example.

In Section III the various results of the paper are derived. These consist of series expressions for the time and autocorrelation functions corresponding to general nonlinear transformations of random signal and noise processes.

Section IV discusses the relation between the expressions obtained in this paper and the expressions previously obtained by the transform method. It is shown that one of the expressions obtained in this paper for an input of Gaussian noise and a single sinusoidal signal is easily obtainable from the general transform solution. Readers familiar with the transform method may wish to read this section before reading the derivation in Section III, since a general solution for a Gaussian noise input is obtained here very simply and quickly.

* Received August 10, 1961; revised manuscript received, May 21, 1962.

† Westinghouse Research and Development Center, Pittsburgh, Pa. Formerly with Sperry Gyroscope Company, Great Neck, L. I., N. Y.

[1] S. O. Rice, "Mathematical analysis of random noise," *Bell Syst. Tech. J.*, vol. 24, pp. 46–156; January, 1946.

[2] W. R. Bennett, "Response of a linear rectifier to signal and noise," *J. Acoust. Soc. Am.*, vol. 15, pp. 164–172; January, 1944.

[3] D. Middleton, "Rectification of a sinusoidally modulated carrier in the presence of noise," Proc. IRE, vol. 36, pp. 1467–1477; December, 1948.

[4] D. Middleton, "Some general results in the theory of noise through nonlinear devices," *Quart. Appl. Math.*, pp. 445–498; January, 1948.

[5] L. Lorne Campbell, "Rectification of two signals in random noise," IRE Trans. on Information Theory, vol. IT-2, pp. 119–124; December, 1956.

[6] Robert Price, "A useful theorem for nonlinear devices having Gaussian inputs," IRE Trans. on Information Theory, vol. IT-4, pp. 69–72; June, 1958.