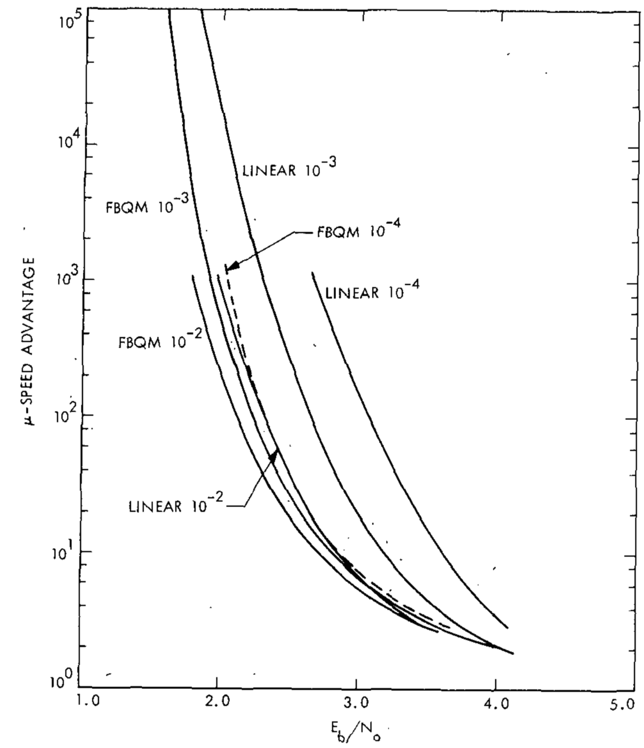

Fig. 6. SNR required to achieve desired erasure probability for varying decoder speed.

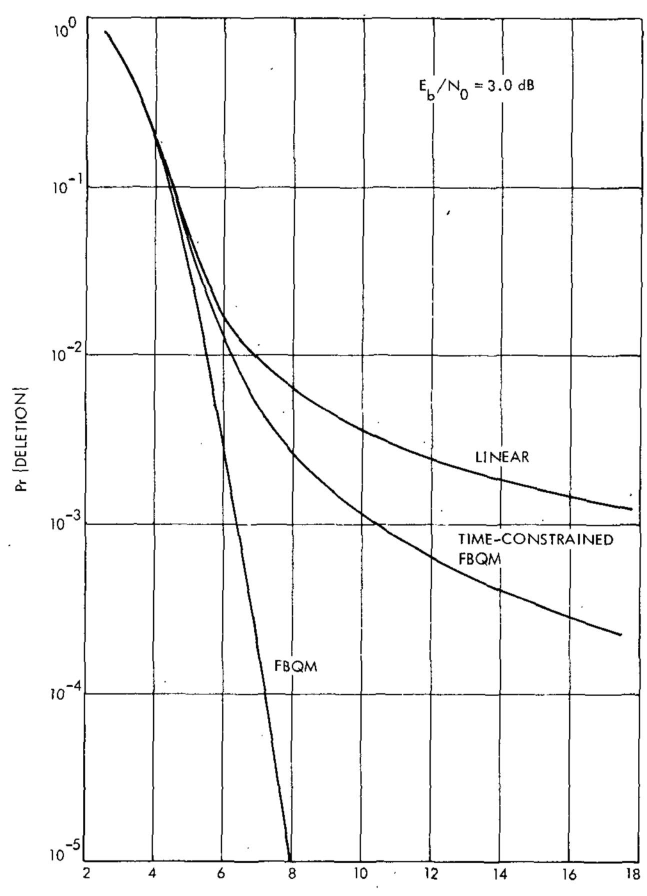

Fig. 7. Performance of FBQM as modified to observe a time-ordering constraint on emitted data.

ceptually possible, a different algorithm could perhaps be found for decoder scheduling which would show improvement over the FBQM–unscrembler combination and still satisfy the time-sequence constraint.

VIII. SUMMARY

This concise paper has developed both a lower bound to the erasure probability of a sequential decoder with an infinite buffer, and a memory-management strategy (FBQM) for decoders with a finite buffer which performs close to the bound. Both the erasure probability of FBQM and its lower bound exhibit an exponential decrease with increasing decoder speed at the sequential-decoding computational cutoff point, where the erasure probability of a conventional sequential decoder exhibits only inverse proportionality.

The performance of this decoder has been approximately evaluated using simulation. A model for the number of computations required to decode a fixed-length block of data was also developed to aid in the evaluation. Performance curves for the FBQM decoder have been presented which show that significant performance improvements are possible, i.e., reduction by 0.5 dB in the value of E_b/N_0 required to achieve an erasure rate of 10^{-3} or a reduction in erasure rate from 10^{-2} to 10^{-4} by the addition of the FBQM strategy, with all other decoding parameters held constant.

A brief discussion was also presented of the problems associated with the reordering into strict time sequence of the FBQM decoder output data. It was shown that even if this unscrambler must share memory with the FBQM decoder itself, a performance improvement equivalent to a better than threefold increase in buffer size is possible, relative to a strictly linear-management decoder.

REFERENCES

[1] J. A. Heller, "Decoding and synchronization research: Description and operation of a sequential decoder simulation program," in *Supporting Research and Advanced Development*, Jet Propulsion Laboratory, Pasadena, Calif., Space Programs Summary 37-58, Aug. 31, 1969, vol. III, p. 42.

[2] J. E. Savage, "The distribution of the sequential decoding computation time," *IEEE Trans. Inform. Theory (International Symposium on Information Theory)*, vol. IT-11, pp. 143–147, Apr. 1966.

[3] I. M. Jacobs and E. R. Berlekamp, "A lower bound to the distribution of computation for sequential decoding," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 167–174, Apr. 1967.

[4] L. Kleinrock, "A continuum of time-sharing scheduling algorithms," in *1970 Spring Joint Computer Conf., AFIPS Conf. Proc.* (Atlantic City, N. J.: 1970), vol. 36, pp. 453–458.

On Using Natural Redundancy for Error Detection

MARTIN E. HELLMAN, MEMBER, IEEE

***Abstract*—In this paper we develop a simple encoder/decoder pair which utilizes the natural redundancy of a source for error detection. It does so with no loss in rate of transmission and with only minimal hardware cost.**

I. INTRODUCTION

Most natural sources of data possess redundancy. That is, not all sequences of characters are meaningful or "typical" source outputs. Thus, if an error occurs in the transmission of such data, the error can frequently be detected or even corrected by use of the natural redundancy. For example if the English language message "I AM

Paper approved by the Associate Editor for Communication Theory of the IEEE Communications Society for publication without oral presentation. Manuscript received June 15, 1973; revised April 4, 1974. This work was supported in part by NSF Grant GK-33250 and in part by the Joint Services Electronics Program under Contract N00014-67-A-0044.

The author is with the Department of Electrical Engineering, Stanford University, Stanford, Calif. 94305.

NOT ABLE TO PROVIDE SUPPORT" is transmitted and received as "I AM NOT AGLE TO PROVSDE SUPPORT" then the two errors (indicated by underbars) are detectable (since there are no English words AGLE and PROVSDE) and are even correctable with a slight effort.

There are two problems with relying on natural redundancy in this fashion. First, there are frequent error patterns which are not correctable, and there are even a large number which are not detectable. For example, if the above message is received as "I AM NOW ABLE TO PROVIDE SUPPORT," not only is the error undetectable, but the meaning of the message is completely reversed. Since this undetectable single-character error is much more likely to occur than the previously cited correctable double-character error, the natural redundancy is not well suited for error correction and detection. A second problem relates to automated detection and correction of errors. A human can easily perform the above detection and correction tasks if he is familiar with English. However, a machine would have greater difficulty in performing this task. Either it must be given a large number of rules governing the spelling and grammar, or if it relies on spotting unallowable digrams (such as VS in PROVSDE) it will miss many errors (such as in AGLE where all digrams are possible when considered by themselves). Similar remarks apply to the use of trigrams, etc. For sources other than natural languages it may even be difficult for a human to spot meaningless outputs.

For these reasons redundant bits are often added to a message for the purpose of error detection or error correction. These redundant bits are added in a manner which makes errors easy to detect and/or correct by automatic means. The theory of error-correcting codes is well documented [1]–[3] and will not be reviewed here. It will be sufficient to note that the addition of such redundant bits results in a lowered rate of transmission of information. Sometimes to offset this loss an effort is made to remove redundancy from the source output. Such source coding, also known as data or bandwidth compression, is usually difficult (i.e., expensive) to implement. In this paper we develop a simply implemented COder/DECoder pair (a CODEC) which transforms the natural redundancy into a form which is ideally suited for error control. This paper treats the problem of error detection and thereby lays the groundwork for a later paper on error correction [6].

II. ERROR PROPAGATION

Error propagation is a problem with many error-correcting codes [4], [5], particularly those which have feedback in the decoder. In the extreme, a code which suffers from catastrophic error propagation can make an infinite number of decoding errors as a result of only a finite number of transmission errors. Significant effort has been directed toward alleviating this problem. It may therefore seem counterproductive to now devote some effort toward designing codes with extreme error propagation. However, the solution to that task is the crux of this paper.

Suppose we represent the English language message "I AM NOT ABLE TO PROVIDE SUPPORT" in binary form using a 5-bit code with A = 00001, B = 00010, C = 00011, $\cdots$, Z = 11010, blank = 11011, period = 11100, comma = 11101, quote = 11110, question mark = 11111, and star = 00000. We then encode the bit string, low-order bits first, with the rate-one convolutional encoder shown in Fig. 1(a). The inverse operation or decoder is shown in Fig. 1(b). Thus, if the channel introduces no errors, the message is decoded correctly. Now suppose a single-character transmission error occurs and it causes the T of NOT to be decoded as a W. The error propagates around the feedback path in the decoder and causes many decoding errors. To be exact the decoded message is "I AM NOWJ.NXAAVWM, EWTY,ROVBGZ, RI." The error propagation is quite severe and is essentially perfect in the following sense.

Definition

A rate-one binary CODEC is perfectly error propagating if following the first transmission error the decoder output is a string of independent random bits. It is said to be k-error propagating if the k decoded bits following the first transmission error are independent and random. In either case we require the system to be causal in that decoded bits preceding the first transmission error must be error free.

The following theorem establishes the ease with which perfectly error-propagating codes can be designed.

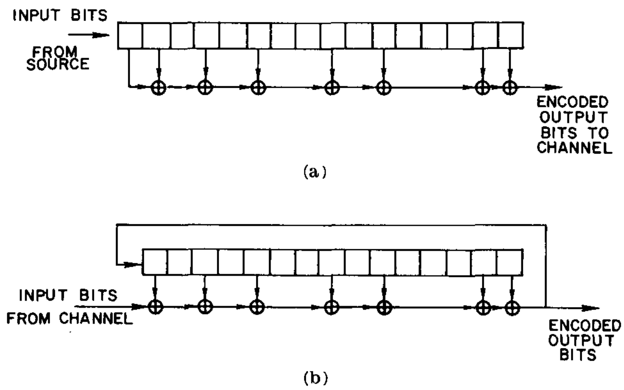

Fig. 1. Rate-one binary error-propagating encoder and decoder.

Theorem

A rate-one binary-convolutional CODEC with constraint length (number of stages in encoder shift register) ν is $(\nu - 1)$-error propagating if the first stage of the encoder is included in the output sum and all other stages are included in the output sum with probability one-half independently of one another. Equivalently, as shown in Fig. 2 the encoding operation is given by

$$x_k = u_k \oplus \sum_{i=1}^{\nu-1} a_i u_{k-i} \tag{1}$$

with $\{a_i\}_{i=1}^{\nu-1}$ being independent random variables, each taking the values 0 and 1 with equal probability. The decoding operation, also shown in Fig. 2 is given by

$$\hat{u}_k = y_k \oplus \sum_{i=1}^{\nu-1} a_i \hat{u}_{k-i}. \tag{2}$$

By convention, for $j \leq 0$, u_j and $\hat{u}_j$ are taken to be 0.

Proof: It is easily seen that preceding the first transmission error $y_k = x_k$ and therefore $\hat{u}_k = u_k$. Thus, if we let j_0 denote the time of the first transmission error, $u_k = \hat{u}_k$ for $k < j_0$. Now

$$\begin{aligned}\hat{u}_{j0} &= y_{j0} \oplus \sum_{i=1}^{\nu-1} a_i \hat{u}_{j0-i}\\ &= e_{j0} \oplus x_{j0} \oplus \sum_{i=1}^{\nu-1} a_i \hat{u}_{j0-i}\\ &= 1 \oplus \left(u_{j0} \oplus \sum_{i=1}^{\nu-1} a_i u_{j0-i}\right) \oplus \sum_{i=1}^{\nu-1} a_i u_{j0-i}\\ &= 1 \oplus u_{j0}.\end{aligned} \tag{3}$$

Therefore, a decoding error occurs at time j_0.

Letting

$$d_k = u_k \oplus \hat{u}_k \tag{4}$$

denote the sequence of decoding errors, we have seen that $d_k = 0$ for $k < j_0$ and $d_{j_0} = 1$. The decoding operation is linear over the binary field, and therefore the response of the decoder to the sequence $y_k = x_k \oplus e_k$ is the mod two sum of its response to x_k and its response to e_k. It follows that d_k, the sequence of decoding errors, is the response of the decoder to e_k, the sequence of transmission errors, and therefore

$$\begin{aligned}d_{j0+1} &= e_{j0+1} \oplus \sum_{i=1}^{\nu-1} a_i d_{j0+1-i}\\ &= e_{j0+1} \oplus a_1.\end{aligned} \tag{5}$$

Since a_1 is as likely to be 0 as it is to be 1, and since it is independent of e_{j0+1}, it follows that d_{j0+1} also takes on the values 0 and 1 with equal probability. Similarly, since

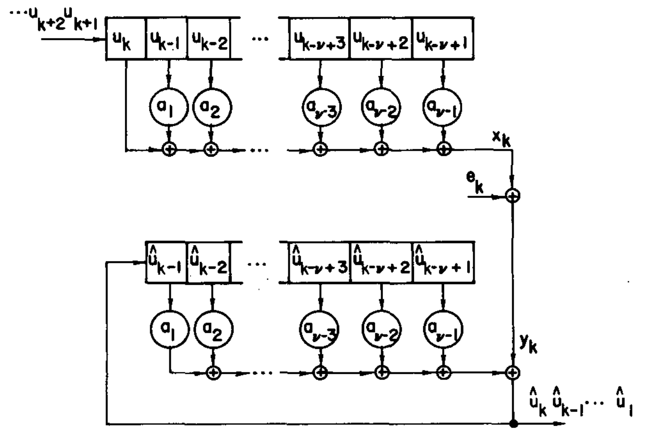

Fig. 2. General, rate-one binary error-propagating CODEC.

$$d_{j0+2} = e_{j0+2} \oplus a_1 d_{j0+1} \oplus a_2 \quad (6)$$

and since a_2 is independent of a_1, d_{j0+1} and e_{j0+2}, it follows that d_{j0+2} is random and independent of d_{j0+1}. In general, for $1 \leqq k \leqq \nu - 1$,

$$d_{j0+k} = e_{j0+k} \oplus \sum_{i=1}^{\nu-1} a_i d_{j0+k-i} = e_{j0+k} \oplus \sum_{i=1}^{k-1} a_i d_{j0+k-i} \oplus a_k \quad (7)$$

and since a_k has not been included in previous sums, d_{j0+k} is random and independent of all previous decoding errors.

Corollary: If N is the length of the message to be encoded and $\nu = N$ then the CODEC is perfectly error propagating.

Although the previous arguments rely on the $\{a_i\}_{i=1}^{\nu-1}$ being chosen at random it is clear that certain choices are to be avoided (e.g., all $a_i = 0$), and other choices to be favored (e.g., $a_{\nu-1} = 1$, since if $a_{\nu-1} = 0$ we could drop off the last stage of the encoder and decoder shift registers). When $N \gg \nu$ there are probably other heuristics which should be developed. For example, should the taps be chosen to maximize the number of transmission errors required to stop the error propagation? The following argument shows that this is probably not a good criterion.

First let us note that a single transmission error propagates indefinitely since a feedback shift register (the decoder) cannot reach the all-zero state from a nonzero state without external inputs. Let us also note that there exist choices of the taps which allow only one additional channel error to stop the error propagation. In particular, if the taps on the decoder correspond to a maximal-length shift-register sequence (MLSRS) [1]–[3], then two transmission errors separated by $2^{\nu-1} - 1$ error-free transmissions will cause decoding errors to occur only during the $2^{\nu-1}$ time units preceding the second transmission error. This is because the $\nu - 1$ error bits in the decoder shift register start in state 100···000 after the first transmission error. During the $2^{\nu-1}$ following time units the decoder is free running, and therefore generates an MLSRS. Following the last of these $2^{\nu-1} - 1$ error-free transmissions the decoder is in state 000···001 so that if no further transmission errors occurred the next state of the decoder would be 100···000 which would be the start of the second cycle of the MLSRS. However, the second transmission error causes the decoder to go from state 000···001 to state 000···000 instead. Error propagation has ceased.

If $\nu = 20$ the finite error propagation caused by the two errors is of length greater than 10^6. This seems quite sufficient to ensure detection of the errors. Furthermore, the MLSRS generated during this time is a moderately good pseudorandom sequence and therefore gives the effect of a much longer constraint length. Remember that to generate a truly random sequence of length 10^6 after the first transmission error would require $\nu = 10^6 + 1$. We will therefore call this an essentially perfect error-propagating code, without precisely defining this term.

III. UNDETECTED ERROR PROBABILITY

From the English language example of Sections I and II it is clear that error-propagating codes allow significant error detection. In this section we quantify these capabilities and develop a CODEC which has even better error-detection capabilities than the one developed in the last section. We also discuss automatic detection of errors.

The undetected error probability is clearly a function of the source statistics. A very redundant, highly structured source such as English makes it easy to spot errors soon after they occur. A less redundant source has outputs which more closely resemble a totally random bit string and it takes longer to recognize that an error has occurred. With a fixed-length message, or a fixed degree of error propagation this translates into a higher undetected error probability.

If the source has fractional rate R then there are 2^{RN} meaningful source outputs of length N. We define $D = 1 - R$ as the fractional redundancy. A source with no redundancy ($D = 0$, $R = 1$) has 2^N, or all possible binary N-tuples, as allowable outputs. English [8] and other natural languages are between 50- and 75-percent redundant ($R = 0.5$ to 0.25, $D = 0.5$ to 0.75).

Now suppose we have a perfectly error-propagating, or an essentially perfect error-propagating code and use it to transmit a source output N bits long. If the first transmitted bit is received in error then the decoder puts out an essentially random bit stream. The probability of the error going undetected $P(e)$ is the probability that this random bit stream is a meaningful source output. Therefore,

$$P(e) \doteq 2^{RN}/2^N = 2^{-ND}. \quad (8)$$

Similarly, if the first error occurs k bits from the end of the message then

$$P(e) \doteq 2^{-kD}. \quad (9)$$

It is seen that the early bits are well protected against undetected errors while later bits are less well protected. It is possible to correct this by use of a two-stage encoding procedure. First encode the information bits $\boldsymbol{u}$ to obtain $\boldsymbol{x}$ as in (1). Then encode $\boldsymbol{x}$, last bit first to obtain the sequence $\boldsymbol{z}$. The taps for the two encoders are chosen independently and at random. The $\boldsymbol{z}$ sequence is transmitted as it comes out of the second encoder. The decoder also has two stages. The first stage takes $\boldsymbol{y} = \boldsymbol{z} \oplus \boldsymbol{e}$ and outputs $\hat{\boldsymbol{x}}$ in reverse order. The second stage takes $\hat{\boldsymbol{x}}$ in proper order, and decodes it to obtain $\hat{\boldsymbol{u}}$. Now any transmission errors propagate in both directions and if $N \doteq \nu$ we find $P(e) \doteq 2^{-ND}$. To be precise, if $\boldsymbol{d} = \boldsymbol{u} \oplus \hat{\boldsymbol{u}}$ and j_0 denotes the position of the first transmission error, one can show that except for $j = N + 1 - j_0$, the d_j are independent random bits. Further,

$$\Pr(d_{N+1-j_0} = 1) = 1/2 + 2^{-(N+1-j_0)} \quad (10)$$

and there is weak dependence of d_{N+1-j_0} on the preceding error bits. Thus $P(e) \doteq 2^{-ND}$ with $2^{-(N-1)D}$ being an upper bound.

As an example of the efficacy of this technique, we encoded the message "I AM NOT ABLE TO PROVIDE SUPPORT" using the encoder of Fig. 1(a) for the first stage and a different constraint length sixteen encoder for the second stage (taps = 1011010011101001). When the same single-character error (taking the T of NOT to a W with the *single*-stage CODEC) was simulated the output of the two-stage CODEC was "XRJNZFXHCRQX'R?MEWCA?OSQVHA*K BOH." Note that just as for the single-stage CODEC, even when $N > \nu$ error propagation is essentially perfect for moderate values of ν. In this example, $N = 165$ bits $> 10\nu$.

IV. DISCUSSION

Although this paper is concerned with using the natural redundancy for error detection, it is clear that the unidirectional error-propagating CODEC can be used for at least limited error correction. For example, the decoded message "I AM NOWJ.NXAAVWM,EWTY, ROVBGZ, RI" indicates that the first transmission error occurred at or prior to the J of NOWJ, since such a sequence of letters is not allowed in English. Similarly, the sequence I AM is probably error free since it is followed by several plausible characters. If we try changing the J of NOWJ to a space we obtain the output "I AM NOW HU.CVKIWXRORBHUWTZHUIGK*" which indicates that either this was not the proper correction or that more than one character is in error. Trying other single-character corrections on the J and W of NOWJ results in meaningful text only when the proper correction (W to T) is made. If a second error occurs later and it is far enough removed from the first error, then correction of both errors can proceed independently. Use of a more sophisticated decision algorithm (basically, a sequential decoder) allows correction of many more error patterns. Surprisingly, such an algorithm will decode reliably (i.e., with a vanishingly small error probability as $\nu \to \infty$) so long as the rate of the source is less than the capacity of the channel. This technique is thus optimal. The proof of this result is left to a later paper [6] which also generalizes this technique to coders with rates other than 1. In particular if the channel is noiseless then rates larger than 1 can be used to produce data compression. To date, optimal source coders or data compressors have been complex pieces of equipment, while the optimal source coders derived in [6] are extremely simple to implement. All of the complexity is transferred to the decoder, a situation well matched to remote telemetry or other applications where complex operations are easier to perform at the receiver than at the transmitter.

A word is in order concerning automatic implementation of the recognition of meaningful source outputs. When the source output is a natural language such as English, a human is usually the ultimate destination and he can easily perform this task. In those instances where machine recognition is required, use of first- or second-order frequency statistics is quite feasible. Using only individual character frequencies with a 5-bit code results in $D \doteq 0.18$ so that transmission errors 23 or more characters from the end of the message have $P(e) \leq 10^{-6}$. Use of a more standard code, such as the 8-bit ASCII code results in $D \doteq 0.48$, and transmission errors nine or more characters from the end of the message have $P(e) \leq 10^{-6}$. Limited error detection is particularly simple with codes such as ASCII where some characters are reserved for future use and are therefore currently not allowed. The detector merely waits for these unallowed characters to announce the presence of an error.

For data sources such as digitized TV or facsimile, other detection schemes are needed. When an error-propagating CODEC is used to transmit a digitized picture, a transmission error will be easily detectable by the snowy-looking random pattern it causes. Automatic recognition of this condition could rely on frequency of transition between gray levels or more complicated statistics of the source.

This discussion could continue with many other special cases, but we will stop here since our point has been made: complete knowledge of the rules of a language is not needed. This remark applies equally well to the error-correction problem treated in [6].

Although it was not mentioned in the body of the paper, it is obvious that there are other circuits for producing error-propagating codes. The convolutional encoder, feedback decoder pair was used for purposes of example because of its ease of implementation. There is one other CODEC which deserves mention, this being the feedback encoder, convolutional decoder pair which results from interchanging the encoder and decoder in Figs. 1 and 2. A CODEC of this type with constraint length ν is also $\nu - 1$ error propagating, but now the error propagation ceases $\nu - 1$ time units after the transmission error. This can be useful if automatic recovery is desired. For example, with $\nu = 31$ and the 5-bit/character code previously used "I AM NOT ABLE TO PROVIDE SUPPORT" is decoded as "I AM NOYRZXFSZTO PROVIDE SUPPORT" when the error pattern is the one which caused the T to be decoded as a W with the CODEC of Fig. 1. Savage [7] proposed the use of such feedback encoders/convolutional decoders for eliminating undesired periodicities in the information sequence. He noted the error-propagation problem and in fact ruled out convolutional encoders with feedback decoders because of the catastrophic error propagation. It is hoped that this paper will help focus attention on the brighter side of what has been regarded as a serious problem.

ACKNOWLEDGMENT

The author wishes to thank C. Davis of Stanford University for his invaluable aid with the simulations.

REFERENCES

[1] E. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.
[2] S. Lin, *An Introduction to Error-Correcting Codes*. Englewood Cliffs, N. J.: Prentice-Hall, 1970.
[3] W. Peterson and E. Weldon, *Error-Correcting Codes*. Cambridge, Mass.: M.I.T. Press, 1972.
[4] J. Massey, *Threshold Decoding*. Cambridge, Mass.: M.I.T. Press, 1963.
[5] G. D. Forney, Jr., "Convolutional codes I: Algebraic structure," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, Nov. 1970.
[6] M. Hellman, "Joint source and channel encoding," *IEEE Trans. Inform. Theory*, to be published.
[7] J. Savage, "Some simple self-synchronizing digital data scramblers," *Bell Syst. Tech. J.*, vol. 46, pp. 449–487, 1967.
[8] C. Shannon, "Prediction and entropy of printed English," *Bell Syst. Tech. J.*, vol. 30, pp. 50–64, 1951.

Synchronization Using Pulse Edge Tracking in Optical Pulse-Position Modulated Communication Systems

R. M. GAGLIARDI, MEMBER, IEEE

***Abstract*—A pulse-position modulated (PPM) optical communication system using narrow pulses of light for data transmission requires accurate time synchronization between transmitter and receiver. The presence of signal energy in the form of optical pulses suggests the use of a pulse edge-tracking method of maintaining the necessary timing. In this report the edge-tracking operation in a binary PPM system is examined, taking into account the quantum nature of the optical transmissions. Consideration is given first to "pure" synchronization using a periodic pulsed intensity, then extended to the case where position modulation is present and auxiliary bit decisioning is needed to aid the tracking operation. Performance analysis is made in terms of timing error and its associated statistics. Timing error variances are shown as a function of system signal-to-noise ratio.**

I. INTRODUCTION

The successful operation of any digital communication system requires accurate time synchronization between the transmitter and receiver. In optical digital systems a common procedure is to use a noncoherent pulse-position modulation (PPM) mode of operation using narrow pulses of light intensity to carry the data [1]. The presence of signal energy in the form of optical pulses suggests the use of a pulse edge-tracking method of maintaining the necessary time synchronization. In pulse edge tracking the edges of the transmitted pulses are used as timing markers to adjust the synchronization of the receiver. When the optical pulses are transmitted as a periodic pulse train of known fixed frequency, the edge tracking corresponds to "pure" synchronization, in that the transmitted edges always occur at periodic points in time. When position modulation is present, however, the pulses of light are shifted according to the data, and the edge-tracking operation must be modified in order to maintain receiver timing. The latter type of synchronization is often called modulation-derived synchronization, or "impure" syncing, since the timing must be derived from, or accomplished in the presence of, the data modulation. In this paper we examine the pure and impure edge-tracking operation in an optical binary PPM system, taking into account the quantum nature of the light transmission. Performance comparisons are made in terms of the instantaneous timing error of the receiver and its associated statistics. The effect of imperfect timing on the overall data decoding operation has been studied elsewhere [2] and will not be considered here.

The time synchronization problem has of course received considerable attention in the past for the additive Gaussian noise channel, and the interested reader is referred to the presentations in recent books by Stiffler [3], Lindsey [4, ch. 3], and Lindsey and Simon [5]. Although the approach here parallels these earlier studies, the quantum nature of the optical channel produces equations significantly different than those of the purely Gaussian channel. Similar mathematical differences were previously observed with the optical

Paper approved by the Associate Editor for Communication Theory of the IEEE Communications Society for publication without oral presentation. This work was sponsored by the National Aeronautics and Space Administration, under NASA Contract NGR-05-108-104. This grant is part of the research program at NASA's Goddard Space Flight Center, Greenbelt, Md. Manuscript received November 27, 1972; revised June 5, 1974.
The author is with the Department of Electrical Engineering, University of Southern California, Los Angeles, Calif. 90007.