J. E. Savage

Three Measures of Decoder Complexity*

Abstract: Three measures of the complexity of error correcting decoders are considered, namely, logic complexity, computation time and computational work (the number of logical operations). Bounds on the complexity required with each measure to decode with probability of error P_e at code rate R are given and the complexity of a number of ad hoc decoding procedures is examined.

1. Introduction

Experience has taught that decoders for error correcting codes are generally complex machines, both conceptually and physically. While experience also provides intuitive notions of complexity, concrete results wait for carefully defined complexity measures. Three such measures that are intuitively appealing yet amenable to analysis are logic complexity (the number of circuit elements), computation time (measured in units of the switching time of gates) and computational work (the number of logical operations). We shall present new results on the computational work done by several ad hoc decoding procedures and present a spectrum of earlier results on each of the complexity measures.[1,2]

Of the three complexity measures, computational work is perhaps the most interesting. It is defined as the product of the number of logic elements in a decoder model and the number of decoding cycles required to decode a received word. Consequently, it is useful in those high-speed data applications in which the decoder cost and decoder speed are important. When decoder cost alone is important and the number of cycles per block is unimportant, logic complexity is the applicable complexity measure. And when the decoding time per block is the most important quantity, such as in very-high-speed data transmission situations, the computation-time measure plays the most important role.

We begin with a discussion of codes, decoding rules, machine models and the three complexity measures. This is followed by a description of ensemble results obtained with logic complexity as the measure. Using counting arguments we show that most decoders in each of several classes are very complex and we relate their complexity to characteristics of the classes. The significance of these results is that decoders with small logic complexity are unlikely to be found by random selection.

We then turn to computational work and derive lower bounds that must be satisfied if an error probability P_e is to be attained at code rate R on noisy channels. Ad hoc decoding schemes are examined and concatenated coding is shown to have a moderately small rate of growth of work with decoding reliability. Two feedback coding strategies introduced by Schalkwijk and Kailath[3,4] are studied from the standpoint of computational work and shown to be inferior to or to offer no improvement over concatenated coding. We also demonstrate that sequential decoding is far from optimal at very small P_e as shown before. We show that concatenated codes include some binary parity-check codes, which demonstrates the existence of well-structured codes with good decoders.

A lower bound on the computation time of decoders is determined and is shown to grow as the logarithm of reliability E, which is the negative logarithm of P_e. This rate of growth can nearly be reached by a decoding procedure for codes that are similar to concatenated codes.

2. Codes, decoders and complexity measures

We shall deal exclusively with block codes, but the knowledgeable reader will note that truncated convolutional codes also form block codes.

Definition: A *block code* of length n and rate R (in bits) is a collection of $M = 2^{nR}$ code words $\{w_m\}$, $1 \leq m \leq M$, each of which is an n-tuple over some finite alphabet

$$\Sigma_A = \{\sigma_0, \sigma_1, \cdots, \sigma_{A-1}\}.$$

* This work has been supported by the National Aeronautics and Space Administration under grant NGR 40-002-082 and by the National Science Foundation under grants GK-3302 and GK-13162.

J. E. Savage is with the Division of Engineering and the Center for Computer and Information Sciences at Brown University.

A special class of block codes is the class of *linear codes*, which are vectors in the row space of a $k \times n$ matrix **G** or, equivalently, in the null space of an $n \times (n - k)$ matrix **H**. We also recognize concatenated codes in which two encoders are concatenated and symbols in the outer code are used to select code words from the inner code. Many other restricted code types exist.

Throughout the paper we assume that codes will be used on discrete memoryless channels (DMC's) with input alphabet Σ_A and output alphabet Σ_B. These channels are described by an $A \times B$ matrix of transition probabilities $\{p_{j/i}\}$, where $p_{j/i}$ is the probability that the jth output is received given that the ith input is transmitted.

The n channel outputs that are received upon transmission of a code word are submitted to a decoder either serially, in parallel, or in a serial-parallel format. (Observe that as many as n parallel channels may be available.) The decoder will then interpret the received n-tuple and reach a decision that will be represented by the decoder outputs. We emphasize that *any representation of decoder decisions will be acceptable.* We shall restrict our attention to decoders that reach at most $2^{nR} + 1$ decisions (corresponding to the 2^{nR} code words and "no decision"). And any representation of these decisions by decoder output will be acceptable.

Definition: A *block decoding rule* for a code of length n and rate R is a mapping of n-tuples over Σ_B to integers in the set $\{1, 2, 3, \cdots, 2^{nR} + 1\}$.

Note that any such mapping constitutes a block decoding rule and that a rule need not require the specification of a code. However, important rules such as the maximum likelihood rule and the standard array rule for linear codes are incomplete without code specification.

Let Y_m, $1 \leq m \leq 2^{nR}$, be the disjoint sets of channel output n-tuples associated with the code words w_m, $1 \leq m \leq 2^{nR}$, in a code of length n, rate R. Let U be the collection of n-tuples not contained in any Y_m. Then, a *decoding error* occurs when w_m is sent if a y is received that is not contained in Y_m. Therefore, the *probability of error* P_e is defined by

$$P_e = \sum_{m=1}^{2^{nR}} \frac{1}{M} \sum_{y \in Y_m} P(y|w_m), \tag{1}$$

where $P(y|w_m)$ is the probability that n-tuple y is received given that code word w_m has been transmitted (which is a product of channel transition probabilities). We assume in (1), as we shall throughout this paper, that the code words are a priori equally probable of being selected for transmission. For convenience we also define *reliability* E by

$$E = -\log_2 P_e, \tag{2}$$

where the base 2 indicates that E is measured in bits.

If fair comparisons are to be made between the complexities of two decoders, one must insist that they be constructed with similar circuit elements. For this reason, we choose to model all decoders by sequential machines or combinational machines constructed with binary logic elements with a fan-in of two and individual binary memory cells.

Definition: A *sequential machine* S is described by a 5-tuple $S = \langle \Sigma_I, \Sigma_J, Q, \delta, \gamma \rangle$, where Σ_I is the input alphabet of S, Σ_J is its output alphabet, Q is the state set, δ is the next state function and γ is the output function, where

$$\delta: \Sigma_I \times Q \rightarrow Q$$

$$\gamma: \Sigma_I \times Q \rightarrow \Sigma_J.$$

A combinational machine uses only logic elements and is simulated by a sequential machine that is given one input symbol only. We require that the memory cells of a sequential machine be accessed individually by the logic elements and not through tape heads or by a matrix of connections as in core memory. However, equivalent circuits for such storage units can be created using logic elements and individually accessed cells and the number of additional logic elements used should be included in the complexity measures.

Definitions of the three measures of complexity are as follows: *Logic complexity* χ_1 is the number of binary logic elements and memory cells in a decoder model. *Computational work* χ_2 is the product XT of the number of logic elements X and cycles T required by a sequential machine model of a decoder to decode a received word. *Computation time* τ is the product T and τ_0, the maximum number of logic levels between all inputs of the sequential machine model and all outputs, where the inputs and outputs include connections to the memory cells. Thus,

$$\chi_2 = XT, \text{ and } \tau = T\tau_0. \tag{3}$$

We are now ready to begin our examination of each of the three complexity measures.

3. Logic complexity

The results that have been obtained with the measure logic complexity are ensemble results.[1] A class of decoders is given and a counting argument is applied to derive a lower bound to the logic complexity of a very large fraction of the decoders. Upper bounds are also given to the logic complexity of the most complex decoder in a class. These results, then, are most useful when decoders of small logic complexity are being sought in a large collection.

Consider some complete set of logic elements, such as the set AND, OR and NOT. Consider also sequential or

combinational machines that have p external inputs and q external outputs. Then, the number of such machines that can be constructed with logic complexity χ_1 or less does not exceed[1] $N(\chi_1) = (2\chi_1 + 2p + 4)^{(2\chi_1+q+1)}$. If there are N different decoders to build and χ_1, p and q are such that

$$N = N(\chi_1)^{1/(1-\epsilon)} \tag{4}$$

for some fixed ϵ, $0 < \epsilon < 1$, then, the fraction F of the decoders that can be built with complexity χ_1 or less does not exceed

$$F \leq \frac{N^{1-\epsilon}}{N} = \frac{1}{N^{\epsilon}}. \tag{5}$$

If ϵ is fixed and N is large, F will be near zero and most of the N decoders will require a logic complexity greater than χ_1. These arguments are the basis for the following theorem.

Theorem[1]*:* Consider a class of N different decoding rules realized by machines with p external binary inputs and q external binary outputs. Fix ϵ, $0 < \epsilon < 1$, and let p^* be the larger of p and q. Then, for large N, almost all of the decoding rules require a logic complexity χ_1 that satisfies

$$\chi_1 + p^* + 2 \geq \frac{1}{2}\frac{(1-\epsilon)\log_2 N}{\log_2(\log_2 N)}. \tag{6}$$

If p^* is much smaller than the right-hand side, the inequality applies principally to χ_1.

The number of block decoding rules for codes of block length n and rate R is the number of mappings of n-tuples over Σ_B to the integers $\{1, 2, \cdots, 2^{nR} + 1\}$. A simple counting argument shows this number to be $(2^{nR} + 1)^{B^n}$. The number of generator matrices $\mathbf{G}$ for systematic binary linear codes is $2^{n^2R(1-R)}$. Two such codes are equivalent if one is obtained from the other by the application of some permutation to each n-tuple. The counting argument that leads to $N(\chi_1)$ for combinational machines involves counting the number of permutations of inputs. With these observations we have from (6) the following.

Theorem[1]*:* Map a set of n received channel letters into a number of binary digits proportional to n. These binary digits are to be supplied to a decoding machine. Fix ϵ, $0 < \epsilon < 1$. Then, for large n, almost all block decoders (sequential or combinational machines) require a logic complexity that satisfies

$$\chi_1 \geq \tfrac{1}{2}(1-\epsilon)\frac{R}{\log_2 B} B^n, \tag{7}$$

and almost all combinational bounded-distance decoders† of systematic binary linear codes require

$$\chi_1 \geq \tfrac{1}{2}(1-\epsilon)\frac{n^2R(1-R)}{\log_2 n^2R(1-R)}. \tag{8}$$

We conjecture that (8) holds also for sequential machine decoders.

Similar results hold for other classes of decoding rules. All that is required is the number N of *distinct* rules in the class.

The bounds of (7) and (8) can nearly be reached. Using a result of Lupanov[5] for combinational machines one can prove the following theorem.

Theorem[1]*:* Given $\epsilon > 0$, every block decoder for codes of rate R and length n can be realized with logic complexity χ_1, bounded by

$$\chi_1 \leq 4(1+\epsilon)\frac{R}{\log_2 B} B^n \tag{9}$$

for large n.

One can also show that every linear code can be decoded by a machine with logic complexity proportional to n^2. This machine generates each of the 2^{nR} code words in succession until it finds a word that is within some specified distance of the received word. Thus, it exhibits good growth of logic complexity but requires an unreasonably large number of decoding cycles. Logic complexity, therefore, suffers because it fails to include a sufficient number of important decoder parameters.

Theorem[1]*:* Given $0 < \epsilon_1 < 1$ and $0 < \epsilon_2$, almost all decoders for block codes require a logic complexity bounded by

$$\chi_1 \geq \tfrac{1}{2}(1-\epsilon_1)\frac{R}{\log_2 B} 2^{(\log_2 B)E/E_L(R)} \tag{10}$$

for large reliability E on DMC's with lower bound exponent $E_L(R)$ for $0 < R < C$, channel capacity. Also, every block decoder can be constructed with logic complexity

$$\chi_1 \leq 4(1+\epsilon_2)\frac{R}{\log_2 B} 2^{(\log_2 B)E/E_r(R)}, \tag{11}$$

where $E_r(R)$ is the random-code exponent.

These two results follow from lower and upper bounds on the probability of error in DMC's.[6,7] We note that the results hold for block decoders with or without feedback from decoder to encoder since the class of all decoders with and without feedback is the same.

4. Computational work

As stated earlier, computational work χ_2 is the product of X, the number of logic elements in a sequential machine decoder, and T, the number of machine cycles needed to decode a received word:

† They correct errors out to one half of the minimum distance or less.

$$\chi_2 = XT. \tag{12}$$

Computational work can be interpreted as the number of logic uses by a decoder. Since computation is done with logic elements, it is intuitively plausible that a minimum number of logic uses is required to achieve a reliability E at rate R on a noisy DMC. We give a lower bound to χ_2 that depends explicitly on E, R and channel parameters. The computational work required by several ad hoc decoding procedures has been determined[2] and will be reported here. We also bound the computational work required by the Schalkwijk and Kailath feedback coding strategies, examine a buffered feedback strategy for the binary erasure channel and demonstrate the existence of decoders for systematic, binary linear codes whose computational work grows as the square of block length and reliability.

Let $f(\cdot)$ be the decoding function realized by a sequential machine decoder with $\chi_2 = XT$. By replicating the logic box of this machine T times we create a combinational machine with XT logic elements that computes $f(\cdot)$. If $C(f)$ is the smallest logic complexity required to compute f with a combinational machine, then[2]

$$\chi_2 = XT \geq C(f). \tag{13}$$

Using the inequality (13), one can show that for equiprobable code words the smallest achievable probability of error $P_e^M(X, T, R)$ that can be obtained on a given DMC by a sequential machine decoder with X logic elements in T cycles obeys[2]

$$P_e^M(X, T, R) \geq P_e^M(XT, 1, R), \tag{14}$$

where $P_e^M(\chi_2, 1, R)$ is the probability of error associated with a combinational machine.

Equation (14) implies that a lower bound to probability of error with sequential machine decoders doing computational work $\chi_2 = XT$ can be obtained by examining combinational decoders of the same logic complexity. Consider a combinational machine in which some output is connected directly to an input. In such a machine, a change in that one input changes the pattern of machine outputs or the decoding decision. Thus, a single error in that input will result in a decoding error.

Lemma[2]*:* On a completely connected DMC with smallest transition probability $P_{\min}$, $P_e^M(XT, 1, R) \geq P_{\min}$ (with or without channel feedback) if any decoder output is equal to an input.

If the probability of error is to be very small, each input can only be connected to outputs through logic elements. Let n be the smallest block length consistent with $P_e = 2^{-nE}$ and rate R on a given completely connected DMC. Then,

$$\chi_2 = XT \geq \tfrac{1}{2}n \tag{15}$$

since the logic elements have at most two inputs and the n received letters are converted into at least n binary inputs to the decoder. From this argument we have the following theorem.

Theorem[2]*:* On completely connected DMC's for which there exists a lower bound to probability of error having exponent $E_L(R)$ (the coefficient of block length n for n large) the smallest computational work required of any decoder that achieves reliability E at rate R satisfies

$$\chi_2 \geq \frac{1}{2}\frac{E}{E_L(R)} \tag{16}$$

for large E when $0 < R < C$, channel capacity. The bound holds with and without channel feedback with $E_L(R)$ suitably defined.

Corollary: Under the same conditions as above, on DMC's

$$\chi \geq \frac{1}{4}\left[\frac{E}{\log_2 (e^2/P_{\min})}\right]^2 \tag{17}$$

at $R = C$; and on the binary symmetric channel (BSC) with crossover probability p

$$\chi \geq \frac{1}{16}\left[\frac{p2^E}{1 - p}\right]^2 \tag{18}$$

at $R = C$.

We conclude from the above that high coding reliability, especially at rates near channel capacity, requires many logic uses or a decoder that has many logic elements, uses many cycles to decode, or both.

The dependence of the bound of (16) on reliability cannot be substantially improved at code rates near zero because we can demonstrate a decoder for codes of small rate whose computational work grows with E as $[E/E(p)]$ $\log_2$ $[E/E(p)]$ on the BSC with crossover probability p, where $E(p) > 0$ for $p < 1/4$. The decoder corrects up to t errors in a code of M code words of length n from a maximal-length sequence code. We fix M and let t be one half of the minimum distance or $t \approx n/4$. The decoder is a bounded-distance decoder and generates each of the M code words. Each word is generated by a shift register containing a number of logic elements proportional to $\log_2 n$. The number of logic elements in other decoder circuits grows no faster than this, while the number of decoder cycles $T = n$. The overall decoder computational work grows as $Mn \log_2 n$ and n is bounded for large E by $n \leq E/E(p)$, from which the desired result follows.

The binary erasure channel (BEC) has two inputs that are either received correctly or erased. (The probability of no erasure is q.) Although it is not a completely connected channel, one might expect the above theorem to

apply. If so, the result of this theorem is good at nonzero rates also, since we can show existence of a feedback coding strategy for the BEC,[2] the decoder of which has χ_2 proportional to $[E/E_r(R)]$ log $[E/E_r(R)]$ (and χ_1 proportional to log $[E/E_r(R)]$) for large E, where $E_r(R)$ is the random code exponent for the BEC.

This feedback coding strategy requires that each of $k = nR$ binary digits be repeated until received correctly, with feedback used to determine if a repeat is necessary.* Digits are released to a customer when they first arrive unerased so that the spacing between decoded digits is not uniform. Y. Kim has recently shown[8] that if a buffer is employed to space the decoded digits, then the decoder computational work will grow as E^2 for large E. The essential steps in his argument are given below.

Let source symbols have duration t_s, channel symbols have duration t_c, assume that decoded digits are fed to a buffer which stores B binary digits and that the first decoded digit is supplied at time t_0 with later digits following with a spacing of t_s. Let n_j be the number of transmissions of the jth information digit required for it to be received unerased for the first time. Then, a decoding error will occur if

$$\sum_{j=1}^{k} n_j > n. \tag{19}$$

Suppose that the lth information digit has not been decoded when it is supposed to be available to the customer. Then,

$$\left(\sum_{j=1}^{l} n_j\right) t_c > t_0 + lt_s, \tag{20}$$

and if (20) holds for any $1 \leq l \leq nR$ the event "buffer underflow" occurs. If more than $B + l$ digits have been decoded at time $t_0 + lt_s$, the buffer will have "overflowed." Buffer overflow occurs if

$$\left(\sum_{j=1}^{r} n_j\right) t_c < t_0 + (r - B - 1)t_s \tag{21}$$

for some $B + 1 \leq r \leq nR$.

The average of n_j is $1/q$, where q is channel capacity as well as the probability of no erasure. If the probability of error, underflow or overflow is to be small, the averages of the random variables on the left-hand sides of (19), (20) and (21) must satisfy, respectively,

$$\begin{aligned} &nR/q < n, \\ &lt_c/q < t_0 + lt_s, \\ &rt_c/q > t_0 + (r - B - 1)t_s. \end{aligned} \tag{22}$$

Also, the total time to produce k source digits must equal the time to transmit n channel digits or $R = t_c/t_s$. But R must also be less than channel capacity q. From the last inequality of (22) we then have, setting $r = nR$,

$$B + 1 > (t_0/t_s) + nR(1 - R/q). \tag{23}$$

We also find that the buffer size must be proportional to n for large n. The number of logic elements required to create such a buffer will be also proportional to n and the number of decoder cycles T will equal n. Consequently, χ_2 will be proportional to n^2 and in the best case, n cannot grow faster than linearly with E. Therefore, the *buffered feedback coding strategy* described above *requires a computational work proportional to* E^2 for large E. This result suggests that no improvement on the E^2 rate of growth is possible when a uniform spacing is desired between decoded digits.

We have shown previously[2] that decoders for concatenated Reed-Solomon codes[9] require a computational work

$$\chi_2 \leq AC^2\left[\frac{E}{E_c(R)}\right]^2 \tag{24}$$

on DMC's, where A is a constant of the decoders, C is channel capacity and $E_c(R)$ is the concatenated coding exponent. These decoders achieve this good dependence on E without the aid of channel feedback. We have also demonstrated[2] that the Ziv iterative decoding procedure does a computational work χ_2 bounded by

$$\chi_2 \leq B\left[\frac{E}{1 - (R/C)^{1/3}}\right]^5, \tag{25}$$

where B is a constant of the procedure.

Sequential decoding has been examined[2] and the arguments used to bound the computational work required are worth repeating here. The decoder requires a buffer that stores B branches or a number X of logic elements that exceeds B. We put a lower bound n on the number T of cycles needed to decode a block of n information digits and note that the probability of buffer overflow in n transmissions, which results in an uncertainty in the transmitted message, is bounded by

$$P_{BF} \geq n\left[\frac{1}{(SF)B}\right]^{\alpha(R)} \exp\left[-0(\log B)^{1/2}\right], \tag{26}$$

where SF is the speed factor, $\alpha(R)$ is the Pareto exponent, and the function $0(n)$ is such that $0(n)/n \to 0$ as $n \to \infty$. Since $2^{-E} = P_e \geq P_{BF}$, we solve for $\chi_2 = XT$ and find that

$$\chi_2 \geq \frac{n^{1-1/\alpha(R)}2^{E/\alpha(R)}}{(SF)} \tag{27}$$

for large E and fixed SF (which is limited by the switching speed of elements). Thus, the computational work of sequential decoders is exponential in E for large E and they are far from optimal in this case.

* An error occurs if more than n repeats are required.

The theorem at the end of Section 3 states that almost all decoders require a computational work that grows exponentially with reliability [note the application of (13)]. So sequential decoders are not much better than the worst decoders in terms of computational work for large reliability. Nevertheless, they are still very useful in space and other applications where the channel is noisy but the desired probability of error is not too small.

We turn next to an examination of two feedback coding strategies[3,4] invented by Schalkwijk and Kailath for the additive gaussian noise channel with noiseless delayless feedback. Details of the two schemes can be found in the literature.[3,4,10] It suffices for our purposes to observe that the strategy for the infinite bandwidth channel case requires the calculation of the quantity*

$$U = My_1 + M \sum_{k=2}^{N} y_k/k \tag{28}$$

to make a decision, where y_j is the jth received channel output, M is the number of code words and N is the number of transmissions. Here

$$M = 2^{Rt}, N = 2^{Ct}, \tag{29}$$

where t is the duration of the messages and R and C are code rate and channel capacity measured in bits. We wish to compare the computational work of this scheme with bounds derived earlier, so we assume that a peak power limit is applied to the channel and that sufficiently fine quantization of channel input and output will result in a channel approximating the additive gaussian noise channel. Under these conditions the probability of error is[10]

$$P_e = 2^{-\epsilon(a)t[1+0(t)]}, \tag{30}$$

where $\epsilon(a)$ is a function of the ratio a of peak to average power.

When the channel is bandlimited, the new quantities are

$$U = My_1 + M(\alpha^2 - 1)^{1/2} \sum_{k=2}^{N} y_k/\alpha^k; \quad \alpha^N = 2^{Ct}, \tag{31}$$

where α is a simple function of signal to noise ratio. M and P_e are given by (29) and (30) except that a new $\epsilon(a)$ applies.

The two procedures require $T = N$ cycles and the bandlimited channel scheme requires the storage of numbers ranging from α^2 to α^N or at least $N - 1$ bits of storage. Therefore, the computational work of the first scheme is bounded by (use $X \geq 1$)

$$\chi_2 \geq T = 2^{Ct}, \tag{32}$$

while that of the second scheme is bounded by

$$\chi_2 \geq (N - 1)N \tag{33}$$

since the storage cells must be accessed by logic elements and since $X \geq N - 1$. Note that these inequalities apply regardless of the number of bits of quantization taken at the channel output. Solving for t in (32) and N in (33) and equating E with $\epsilon(a)t[1 + 0(t)]$ we have for the infinite bandwidth case

$$\chi_2 \gtrsim 2^{CE/\epsilon(a)} \tag{34}$$

and in the bandlimited case

$$\chi_2 \geq \left[\frac{CE}{\epsilon(a) \log_2 \alpha}\right]^2. \tag{35}$$

Consequently, the *Schalkwijk feedback coding scheme* for bandlimited gaussian channels *does not improve on concatenated coding*. The first scheme for channels with infinite bandwidth is very far from optimal for large E.

We next show that the class of concatenated codes[9] includes binary parity-check (BPC) codes or we exhibit decoders for BPC codes whose complexity grows as the square of block length n. This compares very favorably with the "almost all" lower bound of (8) for the class of BPC codes which grows as $n^2/\log_2 n$. Additionally, one can show that the probability of error obtainable with these codes on the BSC decreases exponentially with block length.

Consider a Reed-Solomon (RS) outer code with code word symbols from GF(2^m), a Galois field of characteristic 2. Let $(i_1, i_2, \cdots, i_n)$ be a code word such that $n = 2^m - 1$; $(i_1, i_2, \cdots, i_k)$ are the information symbols of the word; $k = n + 1 - d$; d is the code minimum distance; and the symbols $(i_{k+1}, \cdots, i_n)$ are check symbols. Since the RS codes are linear, the term-by-term sum of two code words over GF(2^m) is another code word. But the symbols can each be represented by binary m-tuples and field addition is then term-by-term addition, modulo 2, of m-tuples. Now use the m-tuples $i_1, i_2, \cdots, i_n$ as information digits for code words in a BPC code of length n^*, rate r^* and minimum distance d^*. If $w = (w_{i_1}, w_{i_2}, \cdots, w_{i_n})$ is a complete code word formed by this process, then the term-by-term sum, modulo 2, of any two such code words is another code word and this establishes that some concatenated codes are BPC codes.

We next show that the error-correcting capability of such a code is greater than or equal to $\lfloor d/2 \rfloor \times \lfloor d^*/2 \rfloor$.† Let each inner code word be decoded and the decoder output represented as m-tuples. A decoding error at this stage will not occur if no more than $\lfloor d^*/2 \rfloor$ channel errors occur. If no more than $\lfloor d/2 \rfloor$ inner decoding errors

* Here we follow Wyner.[10]

† $[X]$ represents the greatest integer less than or equal to X.

occur, the decoder for the RS outer code can correct such symbol errors. Thus, if no more than $\lfloor d/2 \rfloor \times \lfloor d^*/2 \rfloor$ channel errors occur in a word of length nn^*, no distribution of these errors can force more than $\lfloor d/2 \rfloor$ symbol errors in the outer code and a decoding error.

In the inner code the number of information digits per word is $n^*r^* = m = \log_2 (n + 1)$. Thus, the concatenated code words have $km = rn \log_2 (n + 1)$ information digits (r is the rate of the RS code), rate rr^* and have normalized error-correction capability $\rho = (\lfloor d/2 \rfloor/n) \times (\lfloor d^*/2 \rfloor/n^*)$. From the Varshamov-Gilbert-Sacks existence theorem, we know that there exist inner codes of rate $r^* < 1 - H(2\lambda)$ for large n^*. Therefore, if r and r^* are fixed, we can fix ρ at approximately $(1 - r)\lambda/2$ for large n and n^* and if $\rho > p$, the crossover probability of a BSC, the probability of more than $nn^*\rho$ errors, or a decoding error, will decrease exponentially in the concatenated code block length nn^*.

5. Computation time

The third measure of decoder complexity examined here is the minimum decoding time τ to decode a code of rate R with reliability E on a noisy DMC. Given any decoder that executes T cycles and has a delay of τ_0 per cycle, the time to compute outputs is

$$T\tau_0 \geq \tau, \tag{36}$$

since a combinational decoder with computation time $T\tau_0$ can be realized by the cascade combination of T copies of the logic unit in the sequential machine decoder. Equation (36) justifies our use of the computation time of combinational decoders to generate a lower bound to computation time.

To derive a lower bound to τ on completely connected DMC's, we assume that some nonconstant output of a combinational decoder is a function of K received channel letters. Channel letters will be individually coded into binary-tuples so this output depends on at least K nonconstant inputs to the decoder. Thus, at least $\log_2 K$ levels of logic or a computation time of at least $\log_2 K$ will be required. However, if $P_{\min}$ is the minimum channel transition probability, the probability of a decoding error P_e must satisfy

$$P_e \geq P_{\min}^K \tag{37}$$

since every K-tuple has probability of at least $P_{\min}^K$ of being received and some pattern of K transmissions must result in a change in the output or a decoding error. Thus, we have the following theorem.

Theorem[2]*:* On a completely connected DMC with minimum transition probability of $P_{\min}$, a decoding time of τ

$$\tau \geq \log_2 \frac{E}{-\log_2 P_{\min}} \tag{38}$$

is required to decode any code of rate $R > 0$ with reliability E.

To show that the lower bound can be approached in its dependence on E, we consider a form of iterated or concatenated coding[2,9] in which the outer code consists of a number of binary BCH codes. To form each outer code word we form k_0 code words of length n from a binary BCH code of rate r. The k_0 digits in the ith position of each code word are collected together and used to select one word from an inner code of length $N = \log_2 n$ and rate R_0. Therefore,

$$k_0 = R_0 \log_2 n \tag{39}$$

and the overall block length of a complete code word is N_0:

$$N_0 = n \log_2 n. \tag{40}$$

It can be shown[12] that an inner code for this iteration scheme exists such that the overall probability of error decreases exponentially in n. Hence, for large E,

$$n \leq \frac{E}{E_c(R)}, \tag{41}$$

where $E_c(R)$ is the concatenated coding exponent[9] and $R = rR_0$.

The inner decoder can be realized in disjunctive normal form with a number of levels of logic proportional to $\log_2 n$. The decoder for the BCH code can be realized by a combinational machine with a number of levels of logic proportional to $(\log_2 n)$ $[\log_2 (\log_2 n)]$. To see this, we observe that 1) a decoder calculates syndromes with a delay proportional to $\log_2 n$ since the modulo-2 addition of at most n binary digits is required to compute each digit in the syndrome vector. 2) With the Peterson procedure, the number of transmission errors is determined by measurement of the rank of a $t \times t$ matrix which requires calculation of determinants of $t \times t$ or smaller matrices or the sum of t products, each product containing t Galois field elements. Thus, a total delay proportional to $(\log_2 n)$ $[\log_2 (\log_2 n)]$ will be required in this step.[2] 3) When the rank of the $t \times t$ matrix has been determined, a matrix inversion is required that also has delay proportional to $(\log_2 n)$ $[\log_2 (\log_2 n)]$. 4) The last step is the calculation of error locations from the elementary symmetric functions determined from the matrix inversion above. In a combinational machine this step can be completed with a delay proportional to $\log_2 n$. The error locations are used to change received digits.

The above decoder can be realized with a delay bounded by $F(\log_2 n)$ $[\log_2 (\log_2 n)]$ for some constant F, n large. Therefore, we have the following theorem.

Theorem: There exist decoding procedures for arbitrary DMC's for decoding codes of rate R with reliability E

which have a computation time τ bounded by

$$\tau \leq F\left[\log_2 \frac{E}{E_c(R)}\right] \log_2 \left[\log_2 \frac{E}{E_c(R)}\right] \tag{42}$$

for large E, where F is a constant.

While the lower bound can be approached, it should be clear that the decoding time of any decoder that accepts received channel digits sequentially must use a time proportional to the code length n, which is at best proportional to E. Thus, the lower bound can be reached by decoders that process all digits in a received word simultaneously. For real-time decoding, this implies that n parallel channels feed the decoder. The lower bound might also be reached when received data are stored in the memory of a computer and when these data potentially could be processed simultaneously.

6. Conclusions

The complexity of decoders for error-correcting codes has been studied using three measures: logic complexity, computational work and computation time. Standard decoder models consisting of sequential machines and combinational machines constructed of binary logic elements and memory cells have been used so that fair comparisons between the relative complexities of decoders can be made.

Several classes of decoding rules have been studied using the logic complexity measure and we have given lower bounds on the logic complexity of almost all decoders in each class. This bound grows exponentially with block length n for the class of block decoders and as $n^2/\log_2 n$ for the class of binary parity-check codes. While this type of result is useful when selecting a class of decoders for examination, it is insufficient since decoders with small logic complexity may require an enormous number of clock cycles to decode a block.

Computational work, which is defined as the product of the number of logic elements in a decoder and the number of cycles to decode a received word, is a more useful measure. A lower bound to this measure has been derived that is linear in decoding reliability E and inversely proportional to the sphere-packing exponent or any similar exponent on a lower bound to probability of error. This type of bound applies with or without feedback on completely connected DMC's. The existence of decoders for low-rate codes whose computational work grows as $E \log E$ demonstrates that the dependence of the lower bound on E cannot be substantially improved upon at low rates.

While the lower bound applies only to completely connected channels, with or without feedback, there is a feedback coding strategy for the BEC, which is not completely connected, whose computational work grows as $E \log E$ for all code rates less than channel capacity. This suggests that substantial improvement in the lower bound cannot be had at nonzero rates also. However, there is other information to suggest that the best lower bound at nonzero rates grows as E^2. This follows because buffering of the output of the feedback strategy for the BEC mentioned above in order to insure a uniform separation between decoded digits has a complexity with this rate of growth.

The computational work of a number of ad hoc decoding procedures has been determined and we find that concatenated coding is the best of these known and its E^2 growth is equal only to that of the Schalkwijk bandlimited feedback strategy for the peak-power-limited, additive, gaussian noise (AGN) channel. The Ziv iterative coding procedure has a complexity growing as E^5, while sequential decoding, the Schalkwijk-Kailath white AGN channel feedback strategy with a peak energy constraint, and almost all block decoders do a computational work that grows exponentially in E. The nonoptimality of these procedures as measured by computational work is clearly evident for large E.

A lower bound to the third measure, computation time, has been derived for completely connected channels which grows logarithmically in E. An iterative coding procedure, which is a variant of concatenated coding, has been shown to be decodable in time, growing only slightly faster than logarithmically in E. Thus, the behavior of the best decoding time on E is essentially known. However, this best time cannot be reached by decoders that receive one digit at a time; it can only be reached by sequential machine decoders that execute a small number of cycles, perhaps only one cycle.

Decoders are often complex machines and it is useful to have some idea of whether a given decoder is excessively complex. These results will help in this matter. Hopefully a further study of decoder complexity will show how to relate important parameters of important families of codes to decoder complexity, measured in some fashion. Then, a theory of decoder complexity will be more helpful in design.

References

1. J. E. Savage, "The Complexity of Decoders—Part I: Classes of Decoding Rules," *IEEE Trans. Info. Theory* **IT-15,** No. 6, 689 (1969).
2. J. E. Savage, "The Complexity of Decoders—Part II: Computational Work and Decoding Time," submitted to the *IEEE Trans. Info. Theory*.
3. J. P. M. Schalkwijk and T. Kailath, "A Coding Scheme for Additive Noise Channels with Feedback—Part I: No Bandwidth Constraint," *IEEE Trans. Info. Theory* **IT-12,** No. 2, 172 (1966).
4. J. P. M. Schalkwijk, "A Coding Scheme for Additive Noise Channels with Feedback—Part II: Band-Limited Signals," *IEEE Trans. Info. Theory* **IT-12,** No. 2, 183 (1966).

5. O. B. Lupanov, "A Method of Circuit Synthesis," Izv. Sysch. uchebn. zaved., *Radiofizika,* No. 1 (1958).
6. R. G. Gallager, "A Simple Derivation of the Coding Theorem and Some Applications," *IEEE Trans. Info. Theory* **IT-11,** No. 1, 3 (1965).
7. C. E. Shannon, R. G. Gallager and E. R. Berlekamp, "Lower Bounds to Error Probability for Coding on Discrete Memoryless Channels," *Info. and Cont.* **10,** 65–103, 522–552 (1967).
8. Y. Kim, "The Computational Complexity of Decoders with Feedback on the BEC," M.S. Thesis, Division of Engineering, Brown University (1969).
9. G. D. Forney, *Concatenated Codes,* M.I.T. Press, Cambridge, Mass., (1966).
10. A. D. Wyner, "On the Schalkwijk–Kailath Coding Scheme with a Peak Energy Constraint," *IEEE Trans. Info. Theory* **IT-14,** No. 1, 129 (1968).
11. W. W. Peterson, *Error Correcting Codes,* M.I.T. Press, Cambridge, Mass., 1961.
12. J. E. Savage, "A Note on the Performance of Concatenated Codes," to be published, *IEEE Trans. Info. Theory* (July 1970).

Received November 4, 1969