#4

CODING FOR TWO-WAY CHANNELS

JOHN M. WOZENCRAFT
MICHAEL HORSTEIN

TECHNICAL REPORT 383

JANUARY 3, 1961

LOAN COPY ONLY

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
RESEARCH LABORATORY OF ELECTRONICS
CAMBRIDGE, MASSACHUSETTS

CODING FOR TWO-WAY CHANNELS

John M. Wozencraft and Michael Horstein

Abstract

When time-variant noisy two-way channels are protected by coding, they may be
used to provide essentially noiseless feedback, with delay. Service messages can be
automatically exchanged between terminals, and transmission altered in such a way
that the average communication rate is increased, given fixed receiver computers.
The system is somewhat similar to human communication, in that typical errors
are corrected, while grievous ones initiate a request for retransmission. One-way
experimental data are presented to complement the approximate theoretical analysis.

I DEFINITION OF THE PROBLEM

The problem with which we are concerned is that of providing substantially error-free two-way digitalized communication over noisy channels. We assume that at each terminal there exists a decoding computer of fixed computational capacity; our objective is to maximize the number of decoded digits per second available at the computer outputs. Although generalization is possible, in order to fix ideas we also assume initially that the two communication links are Binary Symmetric Channels, abbreviated BSC.

II. ONE-WAY ENCODING AND DECODING

For transmission rates[*] R_t less than the channel capacity C, we know that the attainable decoder probability of error, P_e, for each one-way link approaches zero exponentially as the encoder constraint length n increases; otherwise, P_e is bounded away from zero.[1,2] In a very real sense, achieving accurate communication over noisy one-way channels becomes impossible as R_t approaches C.

It is instructive here to consider an encoding-decoding scheme for which the decoding problem exhibits in fine structure much the same type of behavior: when the channel perturbation of some particular transmission is typical, decoding is easy; as the perturbation increases, decoding becomes difficult. The Peterson procedure[3] for decoding Bose-Chaudhuri codes[4] is of this class, as is the Gallager low-density parity check code.[11] In particular, so also is the sequential search procedure for decoding convolutional codes,[5] to which we now direct our attention. With convolutional codes, each successive information digit i in an infinite sequence influences the next n transmitted digits, where n is the code constraint length. Thus the set of allowable transmitter sequences divides at each information position into two parts, consistent respectively with i = 0 and i = 1.

The sequential decoding procedure operates briefly as follows. The decoder is able to construct each allowable transmitter message at will, digit by digit. Given a segment of received message of length n, it searches through the set of all allowable messages, trying to construct one that is a "close" replica of this segment. The receiver uses, in fact, a set of progressively relaxed criteria to define "close." If it is successful when using the most stringent criterion, it decodes the first information digit of the received sequence as the corresponding first information digit used in constructing the trial transmitter sequence. Otherwise, the decoder proceeds to the next most stringent criterion, and so on. The procedure terminates with the first success and the first information digit in the received segment is thereby decoded. The decoder

---

[*]Except when it is explicitly stated to the contrary, all quantities are per symbol rather than per second.

successively repeats this process for each information digit embedded in (the infinite) received sequence.

For a BSC, the probability of receiving np_1 transmission errors out of n decreases rapidly for p_1 greater than the channel transition probability p_0. It is possible for large n to exploit this fact by choosing the decoding criteria in such a way that the (unconditional) probability of terminating on the jth criterion is approximately

$$P_j = (1-x) x^{j-1} \qquad (0 < x < 1) \tag{1}$$

where x can be freely chosen to minimize the task of decoding. Furthermore, if any particular information digit is decoded on the jth criterion, and the constraint length n is large, we expect for a judicious choice of x that the next information digit will be decoded on either the $(j-1)^{th}$, j^{th}, or $(j+1)^{th}$ criterion. This follows from the fact that the perturbing noise pattern segment when decoding the $(i+1)^{th}$ information digit is obtained by lopping off the first few digits of the ith pattern, and adding a new tail. Thus we are led to representing the decoding procedure approximately as an infinite Markov process, as shown in Fig. 1. The probability parameter p reflects the stability of the decoding problem from digit to digit, and should increase asymptotically to one as n is made larger.
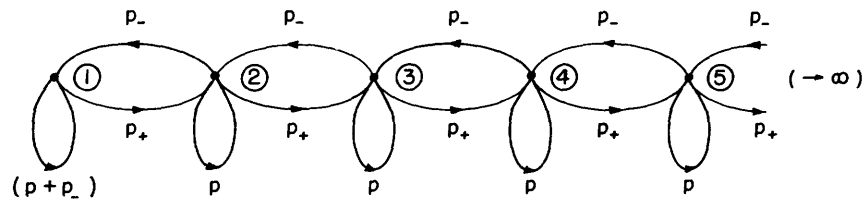

Fig. 1. Infinite Markov representation. The assignments $\left[0 < p < 1;\ p_+ = \frac{x}{1+x} (1-p); \right.$ $\left. p_- = \frac{1}{1+x} (1-p) \right]$ yield the state probabilities $P_j = x^{j-1} (1-x)$. Each operation of the Markov process corresponds to the decoding of an information digit. When the process is in state j, a digit has just been decoded on the jth criterion; the decoding complexity measure for decoding the next digit is N_j.

It is convenient to estimate the amount of computation required to decode an information digit in terms of the average number of trial digits, belonging to allowable transmitter segments whose first information digit is incorrect, that the decoder will construct before finding a "close" sequence. If the previous digit has been decoded on criterion j in our Markov representation, it can be shown by a slight modification of previous work[5] that a rough bound on computational labor so estimated is given by

$$N_j = An^B x^{-jB} \tag{2}$$

where A is a coefficient independent of n and x, n is the constraint length of the convolutional code, and B is the ratio of R_t and the quantity $[1 - \log_2(1 + \sqrt{4p_0(1-p_0)})]$.

The average computation estimate is given by

$$\bar{N} = \sum_{j=1}^{\infty} P_j N_j \tag{3}$$

which converges for $B < 1$ to

$$\bar{N} = An^B(1-x) \frac{x^{-B}}{1 - x^{1-B}} \tag{4}$$

Since x is a free design parameter, we may choose it to minimize the last factor in Eq. 4. Then

$$x = B^{1/1-B} \tag{5}$$

$$\bar{N} = An^B(1-B^{1/1-B}) \cdot \frac{B^{-B/1-B}}{1 - B} \tag{6}$$

$$\lim_{B \to \epsilon} \bar{N} = A; \quad \lim_{B \to 1-\epsilon} \bar{N} = An(e-1)\frac{1}{\epsilon} \tag{7}$$

where ϵ is a positive infinitesimal. The convergence limit, $B = 1$, corresponds to a transmission rate R_t that is always greater than 1/2 of the channel capacity. For $B < 1$, $\bar{N}$ grows less rapidly than linearly with n.

Finally, it is important to point out that the results stated above obtain only when no decoding errors have been made. When an error does occur, the number of computations thereafter is always very large, and successive digits are also very likely to be decoded in error. However, since the probability of making a first error goes exponentially to zero with increasing constraint length n,[5] we can choose n to make it as small as we like.

Let us now tacitly assume that no decoding errors have been made, and that the decoder has always available undecoded received message upon which to work. Let R_r denote the average receiver decoded output rate in digits per second; and M, the fixed computational capability of a decoder in trial-sequence digits constructible per second. Then, on the average, the data rate at the channel input must be $\leq R_r$ for our computer-limited communication system, and the ratio M/R_r measures the computational cost of the communication. For the one-way coding considered thus far, the estimate of this ratio is simply

$$\frac{M}{R_r} = \bar{N} \tag{8}$$

In the sequel we consider strategies for exploiting two-way channels to decrease this ratio. Before doing so, however, it is advantageous to look more closely at the actual operating characteristics of the one-way system.

III. EXPERIMENTAL VALIDATION OF ONE-WAY MODEL

The decoding procedure described above seems eminently plausible, but, on account of statistical dependencies, it cannot be fully analyzed in a rigorous manner. In order to validate the effectiveness of sequential decoding experimentally, Horstein[5,6] has simulated a one-way Binary Symmetric Channel and the encoding-decoding system on the IBM 704 computer.

In these experiments, the transmission rate R_t and code constraint length n were held constant at 1/3 and 72. The actual channel transition probability p_0 was varied between .01 and .06, while the transition probability p_0', which the decoder was programmed to expect, was taken (in an attempt to determine the sensitivity of the procedure to a priori channel estimates) to be .01 or .02. The main experimental results are summarized in Table I, in which the units of $\bar{N}$ are taken to be the construction of a single trial transmitter digit and its comparison against a received digit. A total of 24,000 information digits was decoded, all without error. This is not too surprising: even for $p_0 = .06$, the block-code probability of error at n = 72 is well overbounded by .000413, and only 4000 digits were decoded at this transition probability. For completeness, the rate parameter B is also tabulated.

Several interesting points about these data deserve mention. First of all, the results are somewhat (but not disastrously) sensitive to differences between the expected and actual channel transition probabilities. Furthermore, the approximate minimization of Eq. 5 does not, of course, yield a true minimum, although it does provide a good starting point for experimental minimization. (The fact that $\bar{N}$ is less in row 2 of Table I than in row 1 is an interesting aberration, but is not germane to the present discussion. A full explanation[5,6] requires detailed consideration of the decoding algorithm.) Finally, the concept of sequential decoding, and also certain procedural short-cuts that cannot easily be introduced into the mathematical formulation, are demonstrated to be computationally reasonable. All in all, the picture with regard to average computation is encouraging.

In spite of this pleasant long-term average behavior, however, the one-way sequential decoding procedure suffers from considerable variability. In Fig. 2 a typical sample of the actual short-term (10-digit) average decoding computation requirements for a run of 1000 successive information digits is plotted. For this sample, $p_0 = .04$, $p_0' = .02$, and $\bar{N} = 22$. Peak demands (up to 290 computations per decoded digit) occurred in clusters having a base duration of approximately $nR_t = 24$ digits, as expected.

In order to investigate the validity of the Markov representation of Fig. 1, it is interesting to consider the criteria-level occupancy of this same sample of 1000 successive decoding operations. In Table II we tabulate the number of times the decoding decision was made at each level, and the number of up, return, and down level transitions. In addition to the transitions incorporated in Table II, there were three strangers:

4

Table I.

p_o	p_o'	R_t/C	$\overline{N}$	B
.01	.01	.352	11.6	.451
.02	.01	.388	8.8	.517
.02	.02	.388	11.7	.517
.04	.01	.440	43.8	.637
.04	.02	.440	20.7	.637
.06	.02	.496	73.2	.757

Table II.

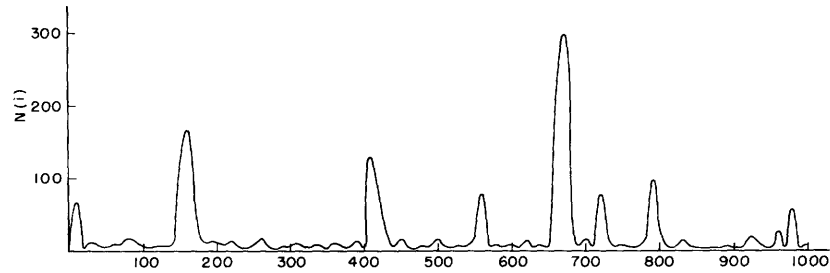| Criterion Level	Level Occupancy	Up Transitions	Return Transitions	Down Transitions
1	909	11	898	0
2	60	9	41	10
3	21	4	10	6
4	9	1	5	2
5	1	0	0	0

Fig. 2. Short-term (10-digit) average computation load. Experimental results ($p_o = .04$, $p_o' = .02$) for 1000 successive decoded digits (for these digits, $\overline{N} = 22$).
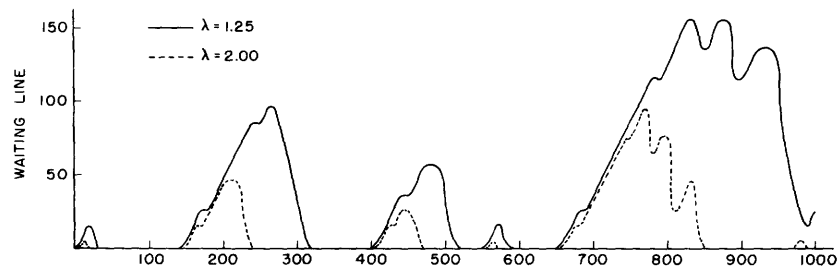

Fig. 3. Waiting line for 1000 successive decoded digits. (Solid line: $\lambda = 1.25$; dashed line, $\lambda = 2.0$.) The ordinate scale is in units of information digits.

one from 3 to 1, one from 4 to 2, and one from 5 to 2. The one-level transition restriction on our Markov representation appears to be well justified.

Assignment of experimental values to the criterion parameter x and the stability parameter p is somewhat difficult. The values x = 1/3 (cf. the value, from Eq. 5, of x = .288) and p = 2/3 appear to be reasonable approximations for purposes of obtaining a rough feeling for the problem. The root of the difficulty lies in the purposeful exaltation of criterion 1 in the programming of the decoder, a mathematical elaboration on the decoding strategy considered here that results in a reduced value for $\bar{N}$.

Variations in decoding demand complicate efficient utilization of the decoding computer. Unless the computational capability of a decoder exceeds the peak computational requirement, a waiting line of undecoded traffic will, with probability one, eventually build up and overflow any fixed computer storage. Let $\lambda > 1$ be the ratio of average computational requirement $\bar{N}$ to computational capability M. Then the frequency of storage overflow decreases as λ is increased, although the efficiency of computer utilization decreases also. As an example of waiting-line behavior, Fig. 3 plots waiting lines ($\lambda = 1.25$ and $\lambda = 2.0$) for the same representative example of 1000 successive decoded digits that we have considered before.

IV. NOISELESS FEEDBACK WITH DELAY

One straightforward way to solve this overflow problem and constrain the magnitude of the variations in decoding computation would be to use a noiseless feedback channel to request retransmission of those received message passages that impose inordinately high computation demands. In some respects, such a procedure is suggestive of the ARQ system.[7,8] Bearing in mind that the output of a decoder can be made substantially noiseless, let us for the moment assume that such an error-free reverse channel is available. It is convenient first to consider a non-Markovian decoder in which successive criteria-level occupancies are statistically independent, but still occur with the probabilities P_j given by Eq. 1.

Let us adopt the following strategy. Whenever the decoder is unsuccessful in decoding a digit with the kth criterion, it sends back a repeat-request over the feedback channel. In general, there will be some delay of, say, T_s seconds (proportional to the constraint span n of the code) before the retransmission is available at the receiver. We then have the following situation. With probability P_j, the decoder will spend approximately N_j/M seconds decoding an information digit. With probability

$$P_s = 1 - \sum_{j=1}^{k} P_j = x^k \tag{9}$$

the decoder will be turned off for T_s seconds. The (estimated) average number of

output digits per second (if we assume that undecoded traffic is otherwise always available) is, therefore,

$$R_r = \cfrac{\sum_{j=1}^{k} P_j}{\frac{1}{M} \sum_{j=1}^{k} P_j N_j + P_s T_s} \tag{10}$$

If we let

$$\overline{N}_k = \sum_{j=1}^{k} P_j N_j \tag{11}$$

and (for the moment) arbitrarily set $T_s = 7nR_t \dfrac{\overline{N}_k}{M}$, we have, finally, for the estimated computational cost per decoded digit

$$\frac{M}{R_r} = \overline{N}_k \frac{1 + 7nR_t P_s}{1 - P_s} \tag{12}$$

It is now possible to choose the repeat level k so as to minimize this cost measure. A little thought will make it clear that not only is the variability of the computational load curtailed by this strategy, but also that our average cost is decreased. In particular, for high transmission rates R_t, such that $B \geqslant 1$ and $\overline{N}$ goes to infinity, $\overline{N}_k$ is still finite. For low rates R_t, where $B \rightarrow \epsilon$, there is infinitesimal advantage, since the probability P_1 of terminating on the first criterion goes to one.

When we consider using noiseless feedback in connection with the Markov decoding
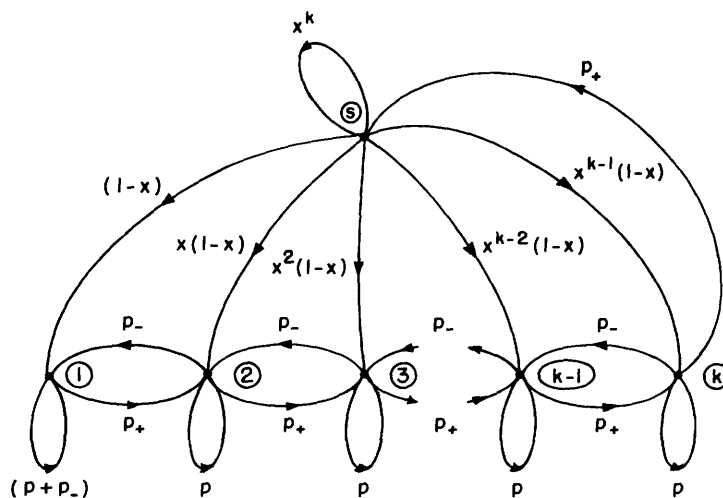

Fig. 4. Finite Markov representation.

representation, the statistical dependency that accentuates the waiting-line problem in one-way channels now works to our advantage by further improving the average computational cost. We again move into state s (repeat-request) when we are unsuccessful in decoding at criterion level k, and move back down from s into any state j with probability P_j (since the channel disturbances are without memory). The resulting finite Markov decoding representation is therefore as shown in Fig. 4.

The state-occupancy probabilities Q_j for this finite Markov process are no longer the old probabilities P_j. Instead, we now have

$$Q_j = P_j \frac{\dfrac{1 - x^{k-j+1}}{1 - x} - (k-j+1) \, x^k}{D} \tag{13}$$

and

$$Q_s = P_s \frac{q \, \dfrac{1 - x}{1 + x}}{D} \tag{14}$$

where

$$q = 1 - p \qquad \text{(the stability parameter)} \tag{15}$$

and

$$D = \frac{1 - x^{2k+1}}{1 - x} - (1+2k) \, x^k + qx^k \frac{1 - x}{1 + x} \tag{16}$$

Although these expressions are somewhat complicated, it is possible to show that, for all k and x,

$$Q_s \leqslant P_s; \quad Q_1 \geqslant P_1; \quad \frac{Q_{j+1}}{Q_j} \leqslant \frac{P_{j+1}}{P_j} \tag{17}$$

where the equality signs hold in the limit $x \to 0$ or $k \to \infty$. These are exactly the characteristics that one would seek in attempting to reduce both the mean and variability of the computation load. When Eqs. 13-16 are substituted in Eq. 11, we have

$$\bar{N}_k = \frac{An^B}{D} \, x^{-B} \left\{ \frac{1 - x^{k(1-B)}}{1 - x^{1-B}} \left[1 - x^k(1-x)(k+1)\right] - x^k \left[\frac{x^{-kB} - 1}{x^{-B} - 1} \right] \right. $$

$$\left. + \left[\frac{1 - x^{k(1-B)}}{(1-x^{1-B})^2} - \frac{kx^{k(1-B)}}{1 - x^{1-B}} \right](1-x) \, x^k \right\} \tag{18}$$

This expression can now be introduced into Eq. 12, with Q_s substituted for P_s, and the cost measure M/R_r minimized with respect to k and (granted sufficient energy) x.

Although our Markov representation seems to incorporate most of the dominant qualitative features of the actual decoding process, it is sufficiently inaccurate quantitatively that an overabundance of numerical work is scarcely justified. The limited

Table III.

		No Feedback	Statistically Independent Feedback	Markov Feedback
$nR_t = 33$ (q=.333)	B = 0	A ( )	A ( )	A ( )
	$B = \frac{1}{2}$	$3A\sqrt{n}$ ( )	$2.99\,A\sqrt{n}$ (9)	$2.97\,A\sqrt{n}$ (5)
	$B = 1 - \epsilon$	$1.72\,An\frac{1}{\epsilon}$ ( )	$14.6\,An$ (7)	$9.10\,An$ (4)
$nR_t = 67$ (q=.150)	B = 0	A ( )	A ( )	A ( )
	$B = \frac{1}{2}$	$3A\sqrt{n}$ ( )	$2.99\,A\sqrt{n}$ (10)	$2.96\,A\sqrt{n}$ (5)
	$B = 1 - \epsilon$	$1.72\,An\frac{1}{\epsilon}$ ( )	$15.7\,An$ (8)	$8.76\,An$ (4)

10

data, partially minimized, that are presented in Table III suffice to point out the major results. In this table are entered the estimates of $\left.\frac{M}{R_r}\right|_{min}$, and (where appropriate), in parentheses, the minimizing value of k. The data are computed for various conditions of feedback, transmission-rate parameter B, and information-digit constraint span nR_t.

The first noteworthy point in connection with these data is that the substantive gain in average cost with feedback is restricted to values of R_t such that $B > 1/2$; for $B \to 1$ the gain is effectively infinite. Furthermore, this gain is quite insensitive to any increase in the null duration T_s implied by a reduction in the probability of error (increased n). In fact, the Markov feedback gain may even improve slightly if the stability parameter p increases fast enough with n. For the data presented, p was taken as .667 (q=.333) for nR_t = 33, and as .850 (q=.150) for nR_t = 67.

Second, the variability of the computational demand, measured in terms of the optimum repeat-request level k, is increasingly curtailed as B increases. Again, for $B < 1/2$, this does not appear to be a major effect because the probability of ever needing a criterion greater than k is small — even in the absence of feedback. On the other hand, it is indeed fortunate that the remedies of feedback are most effective in exactly those circumstances (B>1/2) in which they are most needed.

The final point, which can be of considerable import, is that the probability of error is reduced when feedback is employed. Only those high-error channel-transition patterns that carry the correct message within the kth criterion of an incorrect message can lead to a decoder mistake; the limbo beyond k leads only to a repeat-request. Thus for a given channel and tolerable P_e the information digit constraint span nR_t of the code, and hence the basic complexity of the computers, can be reduced.

V. DERIVED NOISELESS FEEDBACK

Next, let us consider in more detail the problem of deriving noiseless feedback from a noisy channel. Given a sufficiently large code constraint length n, the encoded data can eventually be recovered with negligible error, for $R_t < C$. It is clear that our procedure should therefore include the insertion of the retransmission request as a service bit into the backwardgoing information stream, before encoding; this service bit is then itself protected from the vagaries of the channel.

It is most convenient to envision a fully synchronous system, and to identify these service bits by their location within the information stream. Insofar as possible, we desire to minimize the dead time between repeat-request and retransmission. Let us therefore usurp the first bit in each span of nR_t information digits for service purposes, and adopt the convention that "0" means continue, and "1" means retransmit. The effective information rate of the system is thereby reduced by the generally negligible factor $1 - \frac{1}{nR_t}$. We note in passing that closer spacing of service bits would be of little

benefit in reducing dead time, since a delay of at least n transmitted digits (nR_t information digits) is already implied by the use of redundant coding.

The probability of decoding the service bit incorrectly goes to zero exponentially with n, and can safely be neglected for significantly large n. The cogent problem to be considered is not whether a repeat-request will be interpreted in error, but how long it will take the distant receiver to recognize it; in order to decode the retransmission, the near receiver must know at what time it starts. Since the feedback channel may itself give rise to decoder delays and the initiation of repeat-requests, we are led to seek a strategy whereby each terminal reacts identically to either the receipt or the transmission of a service "1". We must also provide a span of time, beyond that necessary to receive all of the n digits of encoded transmission that are constrained by a service digit, for the decoder to process this segment of received message; and this time span must be sufficiently in excess of the duration of decoding peaks to allow a certain amount of averaging over residual computational variations. Since our model is fully synchronous, it is both possible and more convenient to measure time in "number of information digits" than in seconds; a decoding time allowance of $2nR_t$ digits, after complete segment receipt, appears to be as small as is reasonable.[*] In this regard, the effect of statistical dependence is to degrade rather than improve the efficiency of feedback.

One feedback strategy meeting the requirements discussed above is embodied in the following rules, which are applicable independently to each terminal:

1. If, when a service bit is due to be transmitted, the receiver either has not yet completely decoded the third preceding block of nR_t information digits or has decoded a service 1 therein, then the service bit is to be a "1". Otherwise, it is to be a "0".

2. Immediately upon initiating a service "1", a terminal is to retransmit, starting from 7 blocks of nR_t information digits back. Its receiver is to elide the offending block of received message, plus the next 4 blocks.

The effect of these rules is illustrated diagrammatically in Figs. 5 and 6. The abscissa has units of information digits, and each block, labeled a, b, etc., spans nR_t of them. The appropriate service bit is written at the beginning of each block. Underneath each block division mark, in parentheses, is the identification letter of the received message segment that is due to have been decoded by that time. The symbol X represents a receiver-recognized failure to successfully decode that block, either on account of waiting-line build-up or failure of the kth criterion. The curved arrows trace the action of the transmitters, and the straight arrows represent the elision by the receiver of received message. Figure 5 depicts the situation when a service "1" is successfully decoded; Fig. 6, when there are overlapping repeat-requests initiated

---

[*] It is this time allowance, in conjunction with the feedback rules, that lead to the choice of T_s following Eq. 11. If propagation time is significantly large, T_s must be increased accordingly.
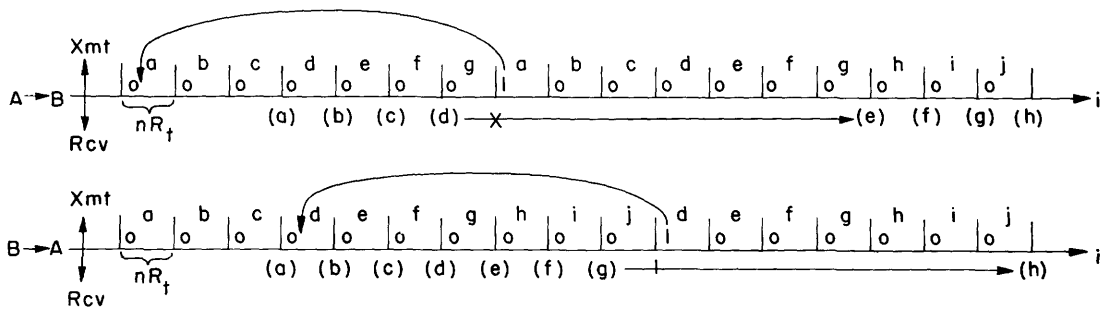
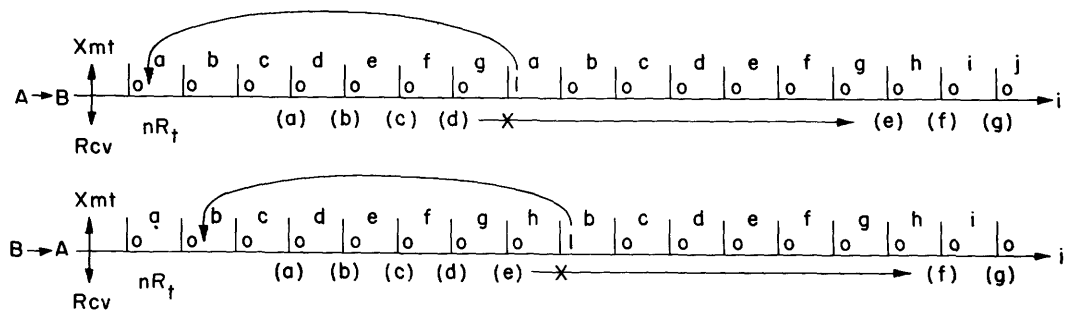Fig. 5. Implementation of two-way strategy: repeat-request decoded.

Fig. 6. Implementation of two-way strategy overlap.

at opposite ends. A little thought makes it clear that other conditions of overlap also lead to the re-establishment of information-digit synchronization and the continuation of error-free communication.

In order to maintain decoding synchronization, we have found it necessary to supplement the kth-state feedback criterion with a waiting-line criterion. This procedure, of course, serves also to solve the waiting-line overflow problem. The question of the optimum criterion level for initiating a repeat-request becomes less critical, and is substantively superseded by the question of determining the optimum lag to schedule between the receipt of a difficult message segment and the transmission of a repeat-request. This, it would appear, is a problem best suited for experimental evaluation.

VI. TIME-VARIANT SYSTEMS

Although the gain in computational efficiency with feedback is negligible for transmission rates R_t such that $B < 1/2$ — which, in fact, may well be a good design operating point — the gain at greater values of B is still important. Even if we transmit digitally at a constant information rate, when the channel is real we expect that its capacity (and hence B) will fluctuate slowly with time. Eventually, should the fluctuation be so severe that C drops below R_t, then decoding errors would encroach; but, long before this happens, a system that is computer-limited would fall irrevocably far behind in decoding the received message: B approaches the convergence limit of unity before R_t approaches C.

The classical engineering procedure with time-variant channels has been to design conservatively: when the channel is poor, communication is still good; when the channel is good, communication is perfect, but slower than it need be; and when the channel is bad, the error-rate becomes intolerable and communication is abandoned.

The exploitation of a code-protected feedback channel, however, seems capable of adding another dimension to system design. In the first place, over real channels, communication rates that are a significant fraction of channel capacity imply (in the absence of coding) error rates that are significantly large. Coding and decoding schemes that correct all probable error patterns, however, permit these higher rates to be used. It is primarily in this objective that the scheme considered here differs from other feedback work,[7-10] which is aimed primarily at error-detection rather than error-correction. Furthermore, with terminal and computer facilities of fixed clock rate, the redundancy ratio of codes can be adjusted by flexible automatic programming in such a way that operation is always near the limit of the equipment's computational capabilities. The difficulty of decoding is itself a most appropriate channel measurement. Thus it seems reasonable to augment the repeat-request strategy with other service bits, requesting an increase in transmitter rate should the decoder be underloaded, or a decrease should it be overloaded.

Terminal equipments that, say, double the effective communication provided by a

14

channel are economic whenever they cost less than does duplicating the channel. The intriguing prospect unfolds of designing communication systems that automatically operate, substantially without error, at whatever maximum rate the condition of the channel and the capital investment in terminal equipment permits.

References

1. C. E. Shannon, The Mathematical Theory of Communication (University of Illinois Press, Urbana, Illinois, 1949).

2. P. Elias, Coding for noisy channels, IRE Convention Record, Part IV, 1955, pp. 37-46.

3. W. W. Peterson, Encoding and error-correction procedures for the Bose-Chaudhuri codes (to appear in Trans. IRE, PGIT).

4. R. C. Bose and D. K. Ray-Chaudhuri, On a class of error-correcting binary group codes, Information and Control 3, 67 (1960).

5. J. M. Wozencraft and B. Reiffen, Sequential Decoding (The Technology Press of the Massachusetts Institute of Technology, Cambridge, Mass., and John Wiley and Sons, Inc., New York, 1961).

6. M. Horstein, An Experimental Study of Sequential Decoding for the Binary Symmetric Channel, Group Report 43-74, Lincoln Laboratory, M.I.T., November 20, 1960.

7. J. B. Moore, Constant-Ratio-Code and Automatic-RQ on transoceanic HF radio services, Trans. IRE, vol. CS-8, no. 1, pp. 72-75, March 1960.

8. H. C. A. van Duuren, Typendruktelegrafie over Radioverbindingen TOR, Tijdschrift van het Nederlands Radiogenootschop, March 1951. See also, Error probability and transmission speed on circuits using error detection and automatic repetition of signals, International Symposium on Data Transmission, Delft, September 1960.

9. J. J. Metzner and K. C. Morgan, Coded feedback communication systems, National Electronics Conference, Chicago, October 1960.

10. A. B. Fontaine and R. G. Gallager, Error statistics and coding for binary transmission over telephone circuits, Report No. 25-G-0023, Lincoln Laboratory, M.I.T., 1960.

11. R. G. Gallager, Low Density Parity Check Codes, Sc.D. Thesis, Department of Electrical Engineering, M.I.T., September 1960.