Retransmission Error Control with Memory

PRADEEP S. SINDHU, STUDENT MEMBER, IEEE

Abstract—Automatic-repeat-request (ARQ) is one of the most commonly used error control techniques today. In this paper, an error control technique that is a basic improvement over ARQ is presented. The technique uses the simple idea of utilizing erroneously received blocks in an ARQ system for error control, retaining most of the other aspects of ARQ. The technique is termed ARQ-with-memory (MRQ).

The general MRQ system is described, and simple upper and lower bounds are derived on the throughput achievable by MRQ. The performance of MRQ with respect to throughput, message delay and probability of error is compared to that of ARQ by simulating both systems using error data from a VHF satellite channel being operated in the ALOHA packet broadcasting mode [9].

I. INTRODUCTION

MANY data communication systems today use a class of error control techniques known as automatic-repeat-request (ARQ) for dealing with transmission errors. ARQ is often preferred over forward-error-correction (FEC) because it is simple, reliable and relatively insensitive to the kinds of errors that occur on real channels. Studies comparing FEC and ARQ [2-6], using both theoretical models and experimental data, have shown that for dependent error channels, ARQ is superior to FEC except for high values of channel error probability. Burton and Sullivan [1] have concluded that while existing ARQ systems may be inadequate for future applications, the solution lies not in FEC systems, but rather in a better ARQ technique.

Although ARQ systems work well for low error rates, their performance as measured by the throughput[1] T deteriorates steadily with increasing transmission error probability. In addition, the probability of long message delay is often unacceptable when dependencies in errors extend over several messages. FEC systems on the other hand, may involve unacceptable decoder complexity in order to achieve reliability comparable to ARQ for noisy channels.

This paper proposes an error control technique which is an extension of the ARQ idea, and is capable of giving the reliability of ARQ without a corresponding drop in throughput for noisy channels. The technique is applicable to conventional point-to-point channels but has particular value in packet broadcasting [7] because it fits in naturally with error control techniques currently used on those channels.

The basic idea behind the technique is that a received block which is determined to contain errors should not be discarded, because it contains useful information about the transmitted block. If the retransmission of this block is also in error, the collective information present in the first block and the retransmission can be used to correct certain errors in the two blocks. In case correction is not possible using the two blocks, additional retransmissions can be used until enough redundancy is present to allow correction.

For a given ARQ system, whether simple or hybrid[2], it is clear that if such correction is incorporated, the average number of retransmissions will tend to decrease. The throughput T for such a system will, therefore, be higher than the throughput for the corresponding ARQ system without correction, but the reliability may be poorer because of the possibility of making errors during correction. However, the reliability can usually be restored to that of the original ARQ system by including enough additional parity check bits in each transmitted block, and we will show that the loss in throughput due to these additional redundant bits can be made negligible if system parameters are chosen appropriately.

An algorithm has been developed for determining the transmitted message from two or more corrupted copies when the channel over which data are transmitted is affected by dependent errors. For this algorithm, the probability of uncorrectable error decreases rapidly with the number of copies available for correction, and the probability of undetected error can be made negligible.

Section II describes an ARQ-With-Memory (MRQ) system based on this idea. The repetition-redundancy error correction algorithm for MRQ is developed in Section III. The performance of such a system is examined in Section IV, and the results of a simulation study on a noisy packet broadcasting channel are presented. These results are used to compare the throughput, message delay distribution and probability of error of MRQ with those of ARQ. Section V concludes the paper.

II. ARQ WITH MEMORY

A) Definition

An ARQ-With-Memory (MRQ) system (Fig. 1) is an ARQ system that is provided additional logic and memory at the receiver for the purpose of correcting errors using retransmitted blocks. Following the convention for ARQ, the term hybrid-MRQ denotes an MRQ system that has embedded FEC applied to individual blocks, independent of and prior to repe-

Paper approved by the Editor for Communication Systems Disciplines of the IEEE Communications Society for publication without oral presentation. Manuscript received August 31, 1976; revised January 3, 1977. This work was supported by The ALOHA System, a research project at the University of Hawaii, which is supported by the Advanced Research Projects Agency of the Department of Defense and monitored by NASA Ames Research Center under Contract NAS2-8590.

The author is with the Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15213.

[1] We define throughput as the average number of information bits accepted by the receiving end divided by the average number of nonidle bits transmitted in the same time.

[2] A hybrid-ARQ system is one that employs embedded FEC on individual received blocks, a retransmission being requested only when FEC yields uncorrectable errors.

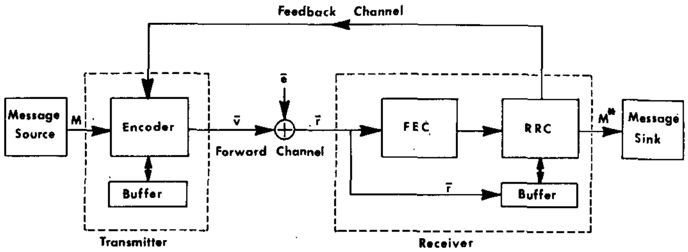

Fig. 1. ARQ-with-Memory (MRQ) sytem.

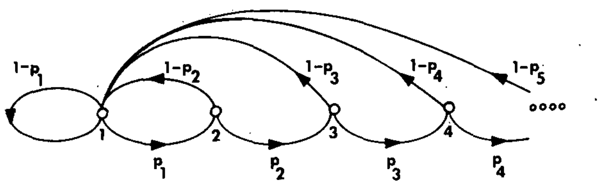

Fig. 2. Operation of MRQ.

tition redundancy correction. However, for the rest of this paper, we will drop the prefix 'hybrid,' and use the terms ARQ and MRQ to refer to both simple and hybrid systems.

The method of calling for retransmissions does not concern us here, and, for the rest of the paper, use of the feedback channel in Fig. 1 will remain implicit. In our description of MRQ, we consider only binary transmission although the ideas presented are applicable to the nonbinary case as well.

B) System Operation

At the transmitter, a binary stream of data is blocked into k-bit message blocks which are encoded into n-bit codewords from a linear block code $C(n, k)$. The codeword $\bar{v}$ corresponding to a message block M is sent over the forward channel (Fig. 1) and is also saved in a transmitter buffer. At the receiver, the corresponding received n-bit block $\bar{r}_1$ is checked to see if it is FEC-correctable into some codeword $\bar{v}^*$. If it is, $\bar{v}^*$ is assumed to be the transmitted codeword, and a positive acknowledgment is sent to the transmitter, inhibiting further retransmissions. If it is not, $\bar{r}_1$ is stored in a receiver buffer, and a retransmission of $\bar{v}$ is requested. If this retransmission $\bar{r}_2$ is FEC-correctable into some codeword, that codeword is assumed to be $\bar{v}$ and a positive acknowledgment is sent; otherwise $\bar{r}_1$ and $\bar{r}_2$ are input to a *repetition redundancy error correction* (RRC) procedure to be described. The errors in $\bar{r}_1$ and $\bar{r}_2$ may be such that this procedure is unable to correct them. In this event, another retransmission $\bar{r}_3$ is requested. If $\bar{r}_3$ is also not FEC-correctable, the RRC procedure is reapplied with $\bar{r}_1$, $\bar{r}_2$ and $\bar{r}_3$ as inputs. This iterative process is repeated till either the latest received word corresponding to $\bar{v}$ is FEC-correctable, or the cumulative store of received words corresponding to $\bar{v}$ is RRC-correctable.

The operation of an MRQ system can be conveniently represented by the state diagram of Fig. 2. Each state corresponds to a different number of received transmissions of $\bar{v}$, and is labeled by this number. From state i, the system goes to state 1 if the latest received word $\bar{r}_i$ is FEC-correctable or if $\bar{r}_1, \bar{r}_2 \cdots \bar{r}_i$ are RRC-correctable. Otherwise the system goes to state $i + 1$ after requesting another transmission. The labels on the edges are transition probabilities which we will characterize in a later section. For the moment, it suffices to say that the probabilities are defined once we make the assumption that errors from block to block are statistically independent and identically distributed.

III. ERROR CORRECTION USING REPETITION REDUNDANCY

A) The Basic Idea

Consider $\bar{r}_1$ and $\bar{r}_2$, the first received word corresponding to $\bar{v}$ and its retransmission, respectively. Let $\bar{d}$ denote the modulo-2 sum of $\bar{r}_1$ and $\bar{r}_2$, and let $\bar{s}_1$ and $\bar{s}_2$ be their respective syndromes; $\bar{s}_i$ being defined as $\bar{r}_i H^T$ where H is the parity check matrix [10, 11] of $C(n, k)$.

The RRC algorithm is based on the following simple idea: if no bit-error overlaps occurred in $\bar{r}_1$ and $\bar{r}_2$, the vector $\bar{d}$ consists of 1's in just those positions where errors occurred in $\bar{r}_1$ or $\bar{r}_2$. In this case determination of $\bar{e}_1$, the pattern of errors in $\bar{r}_1$, corresponds to finding which of the nonzero bits of $\bar{d}$ are from $\bar{r}_1$.

Consider the set of 2^t "trial" error patterns that can be formed by picking different combinations of the t nonzero bits of $\bar{d}$. The vector $\bar{e}_1$ is in this set whenever $\bar{r}_1$ and $\bar{r}_2$ contain nonoverlapping bit-errors. Also, the sum of $\bar{s}_1$ and the syndrome of $\bar{e}_1$ is $\bar{0}$ by definition. This suggests using an exhaustive search, which assumes $\bar{e}_1$ to be the first trial pattern satisfying the above condition as the basic mechanism for correcting the errors in $\bar{r}_1$. Since there may be more than one trial pattern that satisfies the required condition, decoding errors will occur whenever the pattern picked is not equal to $\bar{e}_1$.

When such an exhaustive search is applied to an arbitrary (possibly overlapping) pattern of errors in $\bar{r}_1$ and $\bar{r}_2$, one of the following mutually exclusive events must occur: (i) the error pattern found is equal to $\bar{e}_1$ (the errors in $\bar{r}_1$ and $\bar{r}_2$ are *correctable* by RRC); (ii) the error pattern found is not equal to $\bar{e}_1$ (the errors are *undetectable* by RRC); and (iii) there is no pattern among the trial patterns that satisfies the required condition (the errors are *uncorrectable* by RRC). We note that the last event can occur only if $\bar{r}_1$ and $\bar{r}_2$ contain overlapping errors. In this case, additional retransmissions are needed to correct the errors.

Before we consider how the exhaustive search might make use of additional retransmissions, let us generalize the above procedure so that it becomes efficient for burst-error channels.

B) Modification for the Burst Channel

In a burst-error channel, a transmitted word is typically either not affected by errors at all, or when it is affected, the number of bit errors in the received word tends to be large. This seems to preclude the use of an exhaustive search described earlier because of the large number of patterns to be searched.

However, good use can be made of the fact that errors in the received word are not evenly scattered throughout, but occur in fairly well defined "bursts." Since the number of bursts in a word is, in general, much smaller than the number of bit-errors, an exhaustive search that works with bursts in-

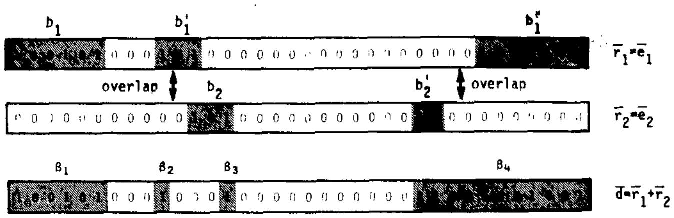

Fig. 3. A set of error patterns illustrating burst overlap; here $g = 3$ and $\bar{v} = \bar{0}$.

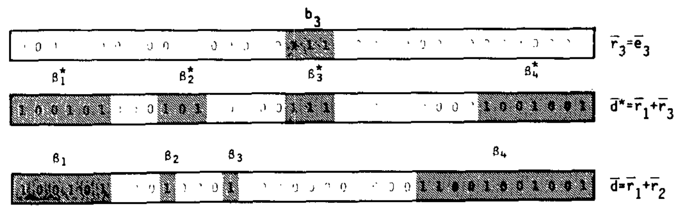

Fig. 4. Vectors formed in the updating process using $\bar{r}_3$; $\bar{r}_1$, $\bar{r}_2$, $\bar{v}$ and g are as in Fig. 3.

stead of bits will be practical. If, in addition, the likelihood of a burst from one received word overlapping one from the next received word is small, such a method will be effective as well. We now make the above intuitive ideas more precise, and describe the method in detail.

Let g be a positive integer called the guard-band used in defining a burst. This number is a parameter in the RRC algorithm, and we assume that it is set to some suitable value before the error correction process begins and remains fixed throughout. The 1's and 0's in the vector $\bar{d} = \bar{r}_1 + \bar{r}_2 = \bar{e}_1 + \bar{e}_2$ can in general be broken up into a sequence of the form $Z_1\beta_1 Z_2\beta_2 \cdots Z_l\beta_l Z_{l+1}$, where Z_1 and Z_{l+1} are sequences of zero or more 0's, each Z_i $(2 \leqslant i \leqslant l)$ is a sequence of g or more 0's, and each β_i $(1 \leqslant i \leqslant l)$ is a burst sequence that begins and ends with a 1 but does not contain g or more consecutive 0's. Corresponding to each burst sequence β_i, we define an n-bit burst vector $\bar{\beta}_i$ which consists of a sequence of 0's except for the sequence β_i appropriately positioned inside the n-bit word.

Clearly, the bursts of $\bar{d}$ will correspond exactly to bursts of either $\bar{e}_1$ or of $\bar{e}_2$ except where burst overlaps have occurred between $\bar{e}_1$ and $\bar{e}_2$. Let m_1 and n_1 be the beginning and ending positions of a burst sequence in $\bar{e}_1$ and m_2 and n_2 be the beginning and ending positions of a burst sequence in $\bar{e}_2$. The two bursts are said to overlap if and only if $(n_2 + g \geqslant m_1)$ and $(n_1 + g \geqslant m_2)$ where g is the guard-band as before. Assuming that the $\bar{0}$ codeword is transmitted, and g is set to 3, Fig. 3 shows a set of error patterns that results in two overlaps between $\bar{e}_1$ and $\bar{e}_2$. Notice that if g had been set to 1 or 2, there would be only one overlap.

Let B denote a list of burst vectors that initially consists of the burst vectors $\bar{\beta}_1, \bar{\beta}_2, \cdots, \bar{\beta}_l$ obtained from $\bar{d}$, and let S denote a list of syndromes that always correspond to the syndromes of the vectors in B. If no burst overlaps occurred in $\bar{e}_1$ and $\bar{e}_2$, B will contain the burst vectors of $\bar{e}_1$ as a subset and an exhaustive search that picks different subsets of B and checks to see if the sums of the corresponding syndromes from S plus $\bar{s}_1$ is $\bar{0}$, will result in the errors in $\bar{r}_1$ being corrected, or possibly, going undetected.

Suppose, however, that the errors in $\bar{r}_1$ and $\bar{r}_2$ are RRC-uncorrectable so that an additional retransmission $\bar{r}_3$ is required. It should be clear that this implies at least one burst-overlap occurred in $\bar{e}_1$ and $\bar{e}_2$. The idea is to try to use $\bar{r}_3$ to resolve the overlaps by updating B (and correspondingly S) so that the new B contains the burst vectors of $\bar{e}_1$ as a subset, thereby enabling the search to work. The updating is done as follows: let $\bar{d}^* = \bar{r}_1 + \bar{r}_3$; for each burst vector $\bar{\beta}$ in $\bar{d}$ if $\bar{\beta}$ overlaps with some $\bar{\beta}^*$ in $\bar{d}^*$, and $\bar{\beta} \neq \bar{\beta}^*$, then $\bar{\beta}^*$ is appended to B and the syndrome of $\bar{\beta}^*$ is appended to S. For example, if $\bar{r}_1$ and $\bar{r}_2$ are as in Fig. 3, and $\bar{r}_3$ is as shown in Fig. 4, the updating procedure will result in $\bar{\beta}_2{}^*$ and $\bar{\beta}_4{}^*$ being appended to B and their syndromes being appended to S.

For the example shown, the updating procedure is able to resolve the overlaps, and barring undetected errors, correction will occur. If the errors in $\bar{r}_3$ are such that resolution is not possible, addition retransmissions are used exactly as $\bar{r}_3$, until correction occurs.

The RRC-Algorithm

Input: The received words $\bar{r}_1, \bar{r}_2, \bar{r}_3, \cdots$.

Output: $\bar{v}^*$, the algorithm's estimate of $\bar{v}$, the transmitted word corresponding to $\bar{r}_1, \bar{r}_2, \bar{r}_3, \cdots$.

Step 1: Initialize B to $\{\bar{\beta}_1, \bar{\beta}_2, \cdots, \bar{\beta}_l\}$, the bursts in $\bar{d} = \bar{r}_1 + \bar{r}_2$. Initialize S to the syndromes of $\bar{\beta}_1, \bar{\beta}_2, \cdots, \bar{\beta}_l$. Set $i = 2$.

Step 2: Let $\{\bar{\beta}_1{}', \bar{\beta}_2{}', \cdots, \bar{\beta}_j{}'\}$ be a subset of B such that the sum of the corresponding syndromes from S when added to $\bar{s}_1$ gives $\bar{0}$. Output $\bar{v}^* = \bar{r}_1 + \bar{\beta}_1{}' + \bar{\beta}_2{}' + \cdots + \bar{\beta}_j{}'$ as the algorithm's estimate of $\bar{v}$ and halt. If no such subset exists, go to Step 3.

Step 3: Set $i = i + 1$. Let $\bar{r}_i$ be the latest received word and let $\bar{d}^* = \bar{r}_1 + \bar{r}_i$. For each burst $\bar{\beta}$ in $\bar{d}$, if $\bar{\beta}$ overlaps with some $\bar{\beta}^*$ in $\bar{d}^*$, and $\bar{\beta} \neq \bar{\beta}^*$, then append $\bar{\beta}^*$ to B and its syndrome to S. Go to Step 2.

C) A Bound on the Probability of Uncorrectable Errors

In a system that involves retransmission, the probability of uncorrectable errors is intimately related to the throughput of the system. Here we show that this probability is bounded from above by the probability of a particular type of error pattern which we call a *degree-i burst overlap.* Inasmuch as this error pattern occurs with low probability, the bound provides a justification for the algorithm developed.

Definition: The received words $\bar{r}_1, \bar{r}_2, \cdots, \bar{r}_i$ are said to have a degree-i burst error overlap if and only if there is a burst in $\bar{e}_1$ that overlaps with at least one burst in each of $\bar{e}_2, \bar{e}_3, \cdots, \bar{e}_i$.

We now state and prove the theorem that forms the basis of our result.

Theorem: Let B and S denote the list of burst vectors and corresponding syndromes that have been iteratively updated using $\bar{r}_3, \bar{r}_4, \cdots, \bar{r}_i$. If the exhaustive search using B and S is unsuccessful then at least one of the following must be true:

M: There is a burst in $\bar{e}_1$ that exactly matches a burst in $\bar{e}_2$.

D_i: There is a degree-i overlap in $\bar{r}_1, \bar{r}_2, \cdots, \bar{r}_i$.

Proof: Assume that both M and D_i are false, and that the exhaustive search is unsuccessful. Since D_i is false, it follows that for every burst $\bar{b}$ in $\bar{e}_1$, there exists a received word $\bar{r}_m$ $(2 \leqslant m \leqslant i)$ such that $\bar{b}$ does not overlap any burst from $\bar{e}_m$.

Since M is false, we are guaranteed that $\bar{b}$ is not completely erased from $\bar{d}$ by the appearance of an identical burst in $\bar{e}_2$, and that at least a portion of $\bar{b}$ appears in $\bar{d}$. Now this portion of $\bar{b}$ must overlap the complete $\bar{b}$ in $\bar{r}_1 + \bar{r}_m$, so $\bar{b}$ must have been appended to B at the time $\bar{r}_m$ was used for updating the lists. Thus every burst of $\bar{e}_1$ is in B and its syndrome in S after $\bar{r}_i$ has been used for updating. This implies that the search must succeed—a contradiction to our initial assumption. Therefore either (or both) of M and D_i must be true.

It is evident that the occurrence of M is a type of permanent error, one that cannot be corrected by the RRC algorithm. Accordingly, through the rest of our analysis we make the assumption that this type of error does not occur.

Let U_i denote the event that the errors in the first i received words are uncorrectable by MRQ, and let F_i denote the event that each of the first i received words has FEC uncorrectable errors. From the above theorem and the fact that FEC is applied before each iteration of the RRC algorithm, we have

$$U_i \Rightarrow D_i \cap F_i.$$

Or equivalently,

$$\Pr(U_i) \leqslant \Pr(D_i \cap F_i). \tag{1}$$

Using our earlier assumption of statistically independent and identically distributed errors, let

$$q = \Pr(D_i \cap F_i \mid D_{i-1}, F_{i-1}), \qquad i \geqslant 2,$$

where q is independent of i. Noting that

$$p_1 = \Pr(F_1) = \Pr(F_i \mid F_{i-1}),$$

and

$$p_i = \Pr(U_i \mid U_{i-1}), \qquad i \geqslant 2,$$

we get the desired result

$$\prod_{j=1}^{i} p_j = \Pr(U_i) \leqslant \Pr(D_i \cap F_i) = p_1 \cdot q^{i-1}. \tag{2}$$

In addition, since

$$\begin{aligned} q &= \Pr(D_i \mid F_i, F_{i-1}, D_{i-1}) \cdot \Pr(F_i \mid D_{i-1}, F_{i-1}) \\ &= \Pr(D_i \mid F_i, F_{i-1}, D_{i-1}) \cdot \Pr(F_i \mid F_{i-1}) \\ &\leqslant p_1, \end{aligned}$$

$$\Pr(U_i) \leqslant p_1{}^i = \Pr(\text{errors are uncorrectable by ARQ}).$$

The error correcting ability of MRQ is therefore superior to that of ARQ, as expected.

IV. PERFORMANCE OF MRQ

The two parameters that are of key interest in an ARQ system are the throughput T and the probability of incorrectly decoding a message P_e. By definition,

$$P_e = \Pr(\bar{v} \neq \bar{v}^*),$$

$$T = \lim_{t\to\infty} \frac{\text{Number of information bits accepted by receiver in } (0, t)}{\text{Number of non-idle bits transmitted in time } (0, t)} = (k/n) \cdot (1/\bar{N}),$$

where $\bar{N}$ is the average number of blocks transmitted per message. In systems where delay is important, it is also of interest to examine the distribution of N.

We now derive bounds on the throughput T and the probability P_e, and see how MRQ performs in practice. We use the results of a simulation study [8] that compares the performance of MRQ and ARQ on a VHF satellite channel which is operated in the ALOHA packet broadcasting mode [7, 9]. The block length used in this study was 640 bits, and a shortened (1651, 1631) Fire code provided single error and double adjacent error correction for the embedded FEC.

A) Throughput and Message Delay

From Fig. 2, the average number of blocks transmitted per message is easily seen to be

$$\bar{N} = (1 - p_1) + 2(1 - p_2)p_1 + 3(1 - p_3)p_1p_2 + \cdots = 1 + \sum_{j=1}^{\infty} \prod_{i=1}^{j} p_i. \tag{3}$$

A simple lower bound on $\bar{N}$ can be obtained by dropping the terms for $j \geqslant 2$ in (3):

$$\bar{N} \geqslant 1 + p_1. \tag{4}$$

This gives a corresponding upper bound on the throughput for an MRQ system:

$$T_{\mathrm{MRQ}} \leqslant \frac{k}{n} \cdot \frac{1}{1 + p_1}. \tag{5}$$

A lower bound on T_{MRQ} is given by the throughput achieved by the corresponding ARQ system:

$$T_{\mathrm{MRQ}} \geqslant \frac{k}{n} \cdot \frac{1}{1 + p_1 + p_1{}^2 + \cdots} = \frac{k}{n} \cdot (1 - p_1) = T_{\mathrm{ARQ}}. \tag{6}$$

Fig. 5 shows a plot of bounds (5) and (6) along with the values of throughput for ARQ and MRQ obtained by simula-

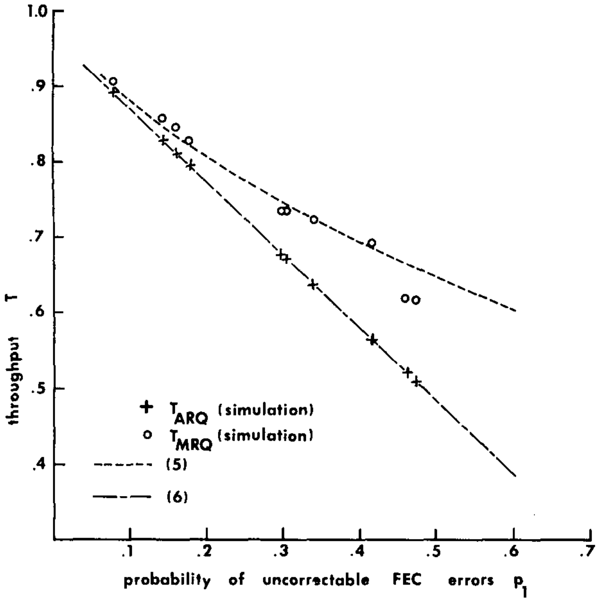

Fig. 5. Throughput for ARQ and MRQ and bounds on T_{MRQ}.

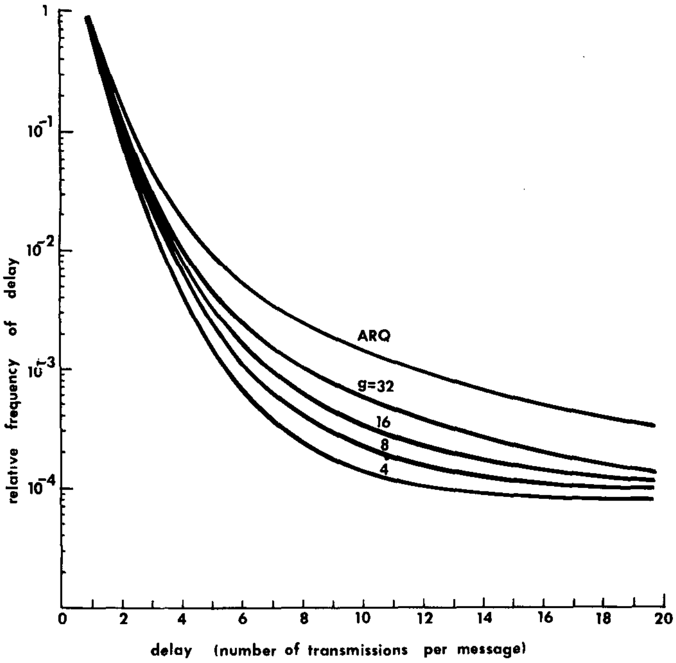

Fig. 6. Message delay distribution for ARQ and MRQ.

tion. For values of $p_1 < 0.4$, T_{MRQ} (simulation) is almost indistinguishable from the upper bound, indicating that the system performs near optimum on this type of channel throughout the range of error probabilities likely to be encountered in practice.

The probability that exactly i blocks will be transmitted for a message is given by:

$$\Pr(N=i)=(1-p_i)\prod_{j=1}^{i-1} p_j \leqslant (1-p_i)p_1 \cdot q^{(i-2)}, \qquad i \geqslant 1 \tag{7}$$

where the product evaluates to 1 for $i = 1$. The inequality comes about by using (2), and indicates that the distribution of N is at worst geometric with i. Fig. 6 shows the distribution of N for ARQ and for MRQ with several values of g, the guard-band. The delay distribution is seen to improve as g is decreased. This is expected because as g decreases, q, the probability of burst overlap, also decreases, forcing the right hand side of (7) to decrease. The best delay distribution will be obtained when $g = 1$ and q is the probability of bit overlap. On the other hand when $g \approx n$, MRQ clearly reduces to ARQ. For the data shown $g = 4$ results in the probability of long delay being improved by close to an order of magnitude.

The guard-band g can therefore be thought of as controlling the error correcting power of the RRC algorithm, more power being available at the expense of more computation.

B) Probability of Error

There are two sources of decoding errors in MRQ: (i) an undetected error in FEC decoding, and (ii) an undetected error in the RRC algorithm.

Let $r = n - k$ be the number of parity check bits in a codeword and let m be the number of error patterns that are FEC correctable into a given codeword. If we assume that each of the 2^n possible error patterns is equally likely, the probability of undetected FEC errors is

$$P_{\mathrm{FEC}} = \frac{m}{2^r}.$$

For a given channel, it is usually possible to design the code so that all the likely error patterns are detected; if this is done, the above estimate is conservative.

Let l be the number of bursts in B in some iteration of the RRC algorithm. The probability of undetected error for each of the 2^l patterns can be assumed to be $1/2^r$, giving an overall probability of $1/2^{(r-l)}$ for this iteration. We note that here again the estimate is conservative since the code can be designed so that the more likely patterns are detected. For any practical implementation of the RRC algorithm, some limit has to be placed on the number of bursts that can be allowed in B to prevent excessive computation. Let this limit be $l_{\max}$. We have

Pr (undected RRC errors per iteration)

$$\leqslant \frac{1}{2^{(r-l_{\max})}} = P_{\mathrm{RRC}}, \text{ say.}$$

The overall system probability may now be derived:

$$P_e \leqslant P_{\mathrm{FEC}} + (P_{\mathrm{FEC}} + (1 - P_{\mathrm{FEC}}) \cdot P_{\mathrm{RRC}})(p_1 + p_1p_2 + p_1p_2p_3 + \cdots)$$

replacing $(1 - P_{\mathrm{FEC}})$ by 1 and noting that p_1 is the largest of the p_i, we get

$$P_e < \frac{P_{\mathrm{FEC}} + p_1 P_{\mathrm{RRC}}}{(1-p_1)} = \frac{m + p_1 \cdot 2^{l_{\max}}}{(1-p_1)2^r} \tag{8}$$

The contribution to P_e due to RRC is determined by $l_{\max}$.

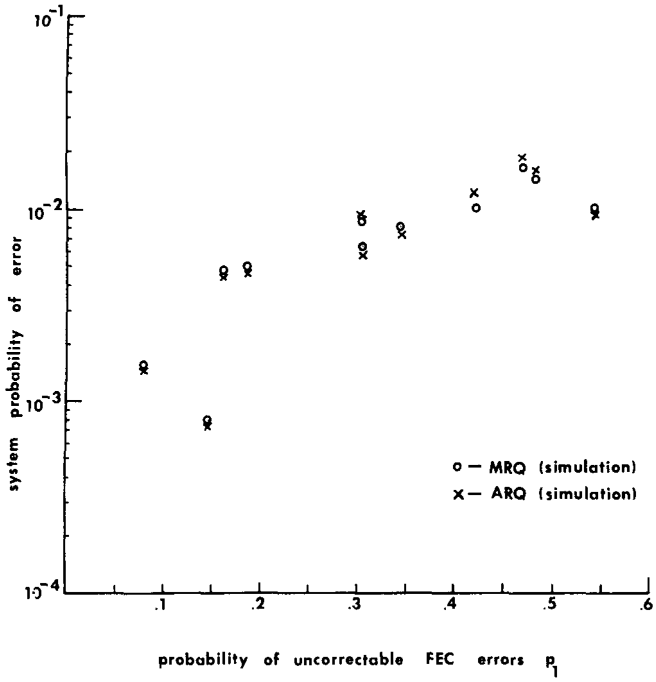

Fig. 7. System probability of error for ARQ and MRQ.

This number can be interpreted to be the extra parity bits needed in an MRQ system to attain the same P_e as the corresponding ARQ system that does not use RRC. If l_{max} is small compared to k, the number of message bits per transmitted block, the drop in throughput due to l_{max} is negligible, and is easily offset by the increase due to a reduction in $\bar{N}$.

This is illustrated in Fig. 7 which shows the system probability of error P_e for ARQ and MRQ for various values of the channel probability of error p_1. It is clear that there is almost no deterioration of P_e in going from ARQ to MRQ. This is also borne out by (8): For the code and FEC capability used, $m = 640 + 639 = 1279$, $r = 20$, and the value of l_{max} was 10. For $p_1 = 0.1$, the contribution due to RRC is approximately 1.1×10^{-4}, or roughly one tenth the P_e of the system without RRC.

V. CONCLUSION

The main attempt in this paper has been to show how repetition redundancy can be used for effective error control, and to demonstrate that the technique developed is a significant improvement over ARQ systems not using memory. The particular algorithm used for error correction has been shown to operate close to optimal with respect to the upper bound on throughput for a particular channel. Further, it has been shown that the optimality is maintained over most of the useful range of channel error probability.

The improvement in throughput over ARQ is seen to increase with increasing channel error probability. This result is interesting, because it is for the large values of channel error probability that one has to fall back upon FEC because of throughput considerations, and tolerate the relatively poor reliability that FEC has to offer. In the region of channel error probability where FEC is a little better in throughput than ARQ, MRQ offers both better rate and better reliability than FEC. For lower channel error probabilities also MRQ is superior to FEC, since ARQ is superior to FEC [3] and because the performance of MRQ is lower bounded by that of ARQ.

We have seen that the use of MRQ considerably improves message delay characteristics—something that is especially important in a network that has interactive users. An added advantage is the ability to easily trade correction power for computation complexity by controlling the guard-band. This allows adaptive operation, where only that much correction power is used as is needed to attain a specified throughput. Further, all these advantages are available with almost no sacrifice in system reliability compared to ARQ.

MRQ can be incorporated in existing ARQ systems and in packet broadcasting systems with little added complexity and almost no changes in protocol. The higher than average level of errors in a packet broadcasting channel makes the use of MRQ even more attractive because of the higher gains in throughput possible.

Along with all the advantages of ARQ, MRQ inherits some of its disadvantages as well. It cannot, for example, be used in situations where retransmission is not practical because of the large round-trip time. Whether implementing MRQ in a given situation is worth the increase in throughput obtained is, of course, dependent upon current technology, but it is clear that with the rapidly decreasing cost of memory and the availability of cheap and versatile computing elements, MRQ is a viable error control technique.

ACKNOWLEDGMENT

The author wishes to thank Dr. Norman Abramson for the encouragement and guidance that he provided throughout the project which resulted in this paper.

REFERENCES

[1] H. O. Burton, and D. D. Sullivan, "Errors and Error Control," *Proceedings of the IEEE,* Vol. 60, No. 11, November 1972.
[2] R. J. Benice and A. H. Frey, Jr., "An Analysis of Retransmission Systems," *IEEE Trans. Commun. Technol.,* Vol. COM-12, December 1964.
[3] R. J. Benice and A. H. Frey, Jr., "Comparisons of Error Control Techniques," *IEEE Trans. Commun. Technol.,* Vol. COM-12, December 1964.
[4] A. B. Fontaine and R. G. Gallager, "Error Statistics and Coding for Binary Transmission Over Telephone Circuits," *Proc. IRE,* Vol. 49, pp. 1059–1065, June 1961.
[5] J. J. Metzner and K. C. Morgan, "Coded Binary Decision-Feedback Communication Systems," *IRE Trans. Commun. Syst.,* Vol. CS-8, pp. 101–113, June 1960.
[6] W. R. Cowell and H. O. Burton, "Computer Simulation of the Use of a Group Code With Retransmission on a Gilbert Burst Channel," *A.I.E.E. Trans.,* January 1962, Pt. 1, No. 58, pp. 577–580.
[7] N. Abramson, "THE ALOHA SYSTEM" in *Computer Communication Networks,* Prentice-Hall, 1973, pp. 501–518.
[8] P. Sindhu, "A Retransmission Error Control Technique for Packet Broadcasting Channels," ALOHA System Technical Report.
[9] D. Wax, "Status Report on UH/ALOHA Participation in the ATS-1 Computer Communication Experiment," ALOHA System Technical Report B74-8, September 1974.
[10] S. Lin, *An Introduction to Error Correcting Codes,* Prentice-Hall, 1970.
[11] W. W. Peterson, and E. J. Weldon, Jr., *Error Correcting Codes,* 2nd edition, the MIT Press, Cambridge, Massachusetts, 1970.

★

Pradeep S. Sindhu (S'76) was born in Bombay, India on September 4, 1952. He received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1974 and the M.S. degree in electrical engineering from the University of Hawaii, Honolulu, in 1976. He is currently working toward the Ph.D. degree in computer archtecture at Carnegie-Mellon University, Pittsburgh, PA.

From 1974 to 1976 he was a Teaching Assistant at the University of Hawaii, and was also associated with the ALOHA System project in computer communication, where he worked on error control techniques for packet broadcasting systems. Since September 1976 he has been a Research Assistant in the Department of Computer Science at Carnegie-Mellon University.

Mr. Sindhu is a student member of ACM and a member of the IEEE Computer Society.

Optimum Quantization for Signal Detection

SALEEM A. KASSAM, MEMBER, IEEE

Abstract—**Optimum quantization of data, primarily for signal detection applications, is considered. It is shown that two useful detection criteria lead to quantization which gives the minimum mean-squared error between the quantized output and the locally optimum nonlinear transform for each data sample. This criterion is an extension of the usual minimum distortion criterion for optimum quantizers. Numerical results show that it leads to optimum quantizers which can be considerably better in their performance for non-Guassian inputs than the minimum-distortion quantizers.**

I. INTRODUCTION

THE PROBLEM of optimally quantizing analog waveforms or data into a finite number of discrete levels for data compression, transmission, and storage has been considered by several authors. Most of this previous work has been concerned with optimum quantization under minimum mean-squared-error or entropy-based criteria [1]-[9]. In addition, for such criteria, adaptive quantizers for nonstationary inputs (e.g., [10]) and robust quaantizers for uncertain input statistics [11] have been studied.

In this paper we approach the quantization problem with a different objective in mind; specifically, we consider optimum quantization of data where the quantized data are to be used to form a test of hypotheses for signal detection. Using quantization performance criteria appropriate for this objective, the optimum quantizer is derived and compared with the minimum mean-squared-error quantizer. As can be expected, for certain types of input statistics the quantizer derived here performs much better in the hypothesis testing applications than the minimum mean-squared-error quantizer.

It is shown that the performance criteria used here in deriving the optimum quantizers lead to quantizers which can be interpreted as optimal solutions of simple generalizations of the mean-squared-error criterion. The results in this paper indicate that in the design of quantizers for data to be used in signal detection (as, for example, in certain telemetering systems) the use of an appropriate performance criterion is important to give an optimum design. The performance criteria we use in this paper are related to the local power of detection in the asymptotic case (small signal, large sample-

Paper approved by the Editor for Communication Theory of the IEEE Communications Society for publication without oral presentation. Manuscript received May 28, 1976; revised October 6, 1976. This research was supported in part by the Air Force Office of Scientific Research under Grant AFDSR-77-3154.

The author is with the Department of Systems Engineering, Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, PA 19174.