AN ALTERNATIVE SOFT-DECISION DECODER

Michael W . Josie
Department of Electrical and Computer Engineering
Brigham Young University
Provo, Utah

ABSTRACT

This paper describes an alterative soft-decision decoding algorithm to be used in hybrid-ARQ error control communication systems. This new scheme promises significant reductions in decoder complexity while exhibiting performance levels comparable to contemporary soft-decision decoders. The development of this algorithm is discussed briefly, its implementation is explained, and the results of computer analysis already performed are considered. It will be shown that the method presented is conducive to the use of a significantly reduced codeword searching algorithm, yielding the advantages of correlation decoding without the corresponding complexity. Keywords: soft-decision block decoding, hybrid- ARQ (Automatic-Repeat-Request) error control, correlation decoding, additive white Gaussian noise channel.

I. INTRODUCTION

The communication system which will be referred to in this paper is shown in Figure 1. Each block of k information bits is encoded to generate an n-bit codeword. Using binary antipodal signaling, codewords are transmitted across a channel where each bit is corrupted by additive white Gaussian noise. The input to the demodulator, then, is actually a stream of real values though the transmission was binary.

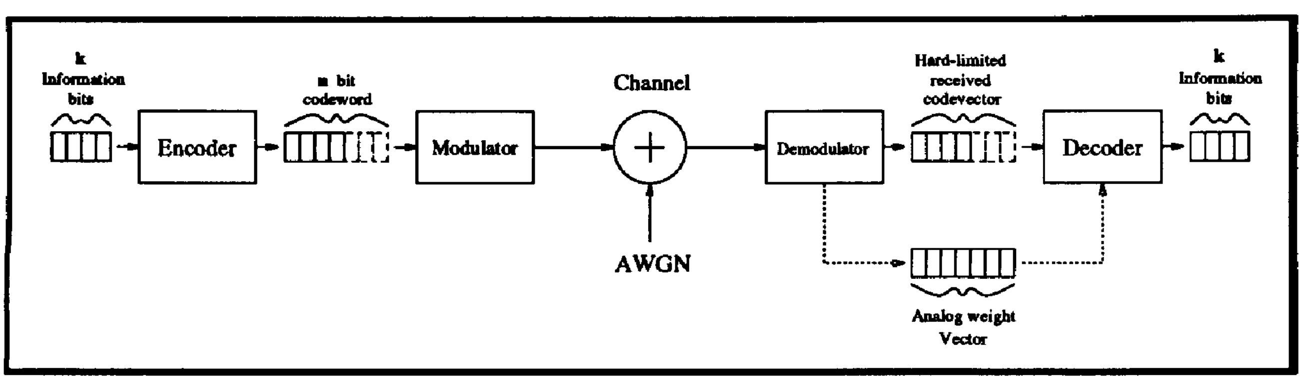

Figure 1. *Block code transmission across an additive white Gaussian noise channel.*

A *hard-decision* decoding scheme assumes that the input at the receiving end is binary. The demodulator makes a bit-by-bit decision on the incoming information and forms a hard-limited version of the received codevector. The actual analog measurements are ignored, and a binary decoder is implemented in an attempt to discern the transmitted codeword and, thereby, retrieve the original block of information.

Soft-decision decoding, on the other hand, utilizes this analog information to expand the error correcting capabilities of the code being implemented In addition to the hard-limited codevector, the demodulator generates a vector of *n* analog *weights* corresponding to the actual measured value of each incoming bit. These weights can be viewed as giving an indication of the probable accuracy of each received bit; the greater the magnitude of the analog weight, the more likely it is that the corresponding bit was received correctly.

In implementing forward error correction, soft-decision decoding has been shown to yield improvements on the order of 2 to 3 dB over hard-decision decoding on an AWGN channel [1]. The price paid is, of course, the comparative complexity of the decoder. The soft-decision decoding schemes employed to date implement a binary decoder in addition to the extra hardware and time required to store and make use of the analog weight vector [2,3]. However, the algorithm presented here bypasses the use of a binary decoder altogether, replacing it with a simple encoder and comparator. A brief explanation of the theory behind this scheme follows.

II. DEVELOPMENT OF THE NEW ALGORITHM

Any given block encoding scheme will generate 2^k unique codewords. In implementing soft-decision decoding techniques, these codewords can be viewed as forming a set of vertices of the *n*-dimensional hyper-cube centered about the origin in *n*-dimensional Euclidean space. Each codeword, then, would be a vector consisting of *n* elements in the set {+1,-1}. When encoding, bits are considered to take on values of either 1 or 0; but for nearly all other purposes of the process to be described, the binary set {+1,-1} will be employed, where the mapping between the two sets should be apparent.

Each real-valued codevector coming into the receiver can be plotted in this *n*-dimensional space. A complete forward error correction (FEC) correlation decoder would map this received vector to the codeword nearest to it geometrically. In this setting, the decoding region for each codeword would be the hyper-pyramid centered about it with vertex at the origin as illustrated in 3-dimensional Euclidean space in Figure 2a. The reliability of such a mapping, however, decreases as the distance between the received vector and the corresponding codeword increases. A hybrid-ARQ decoding scheme should decrease the decoding region for each codeword in order to ensure a required level of decoder accuracy. The scheme at hand reduces each of these regions into a hyper-cone extending from the origin as shown in Figure 2b [4]. Thus, received vectors falling outside any decoding region are not decoded and a retransmission is requested. This has the effect of decreasing system throughput while increasing decoder reliability.

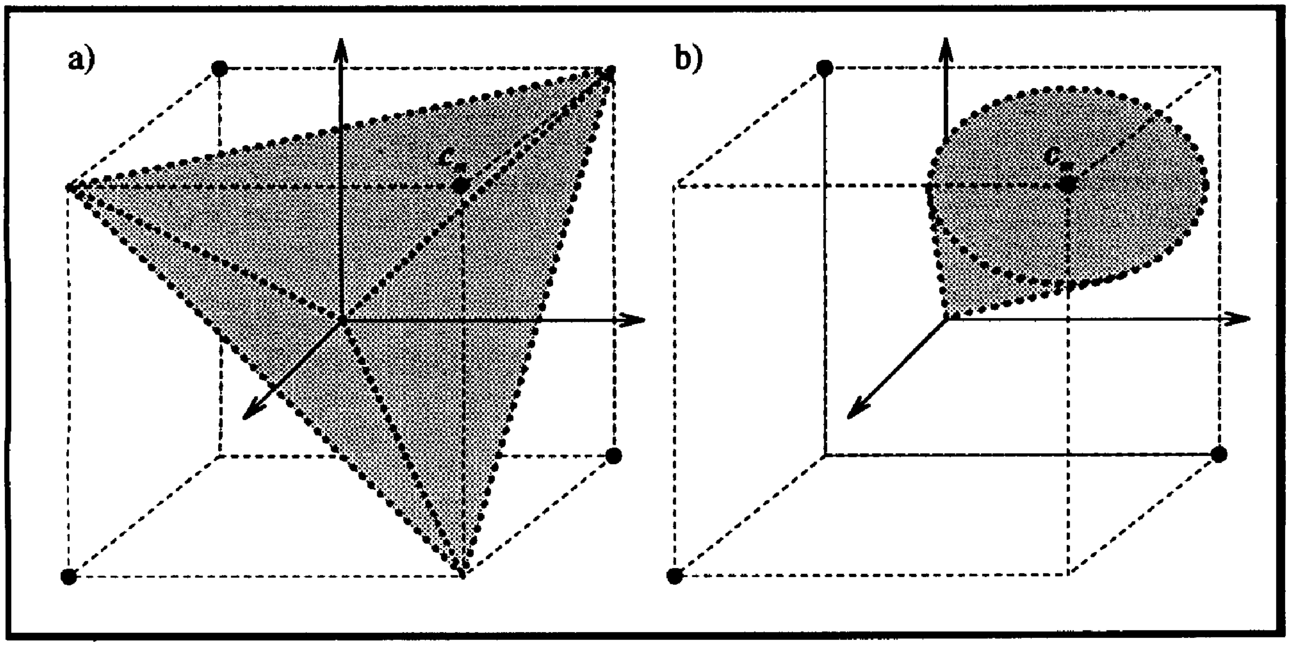

Figure 2. *Corresponding decoding regions of the codeword c_m for a complete correlation decoder (a), and for the present algorithm (b).*

It can be shown that any vector falling within the hyper-conic decoding region of a given codeword satisfies the threshold [4]

$$\frac{\mathbf{c}_m \cdot \mathbf{r}}{\sqrt{\mathbf{r} \cdot \mathbf{r}}} > \frac{n - \rho d_{min}}{\sqrt{(n - \rho d_{min}) - \rho(1 - \rho) d_{min}}} \qquad (1)$$

where *D* = an adjustable decoding parameter unique to this algorithm

$\mathbf{c}_m$ = the codeword central to the hyper-conic decoding region (1 # *m*# 2^k)

$\mathbf{r}$ = the received real-numbered vector

d_{min} = the minimum Hamming distance of the code being used

The parameter *D* is restricted to the real interval (0,1]. A *D* of 1.0 causes the hyper-conic decoding regions to become as large as possible while remaining disjoint [5]. Decreasing *D* serves to decrease the decoding regions, thereby increasing decoder reliability while decreasing throughput.

One significant aspect of this decoding algorithm is that there is at most one codeword, $\boldsymbol{c}_m$, satisfying the condition above. Therefore, once such a codeword is found, it is accepted as that which was transmitted and the next incoming bit stream is received. A complete correlation decoder, on the other hand, can not make a decision until the received vector $\boldsymbol{r}$ is compared against every possible $\boldsymbol{c}_m$. As will be shown, sufficiently concatenated codeword searching algorithms can be employed with this scheme to attain much of the advantage of correlation decoding while avoiding the complexity thereof.

III. IMPLEMENTATION OF THE NEW ALGORITHM

In implementing this algorithm, the actual measured value of each received bit is stored to form a vector of $\boldsymbol{n}$ real numbers. In addition, a bit-by-bit decision is made to form a hard-limited binary codevector from the analog measurements. These two vectors will be used in the decoding process as shown in Figure 3. Only the k information bits are used from the hard-limited codevector. The redundant check bits are discarded and the information vector is re-encoded to form a new codeword. This, then, represents the codeword most likely to have actually been transmitted.

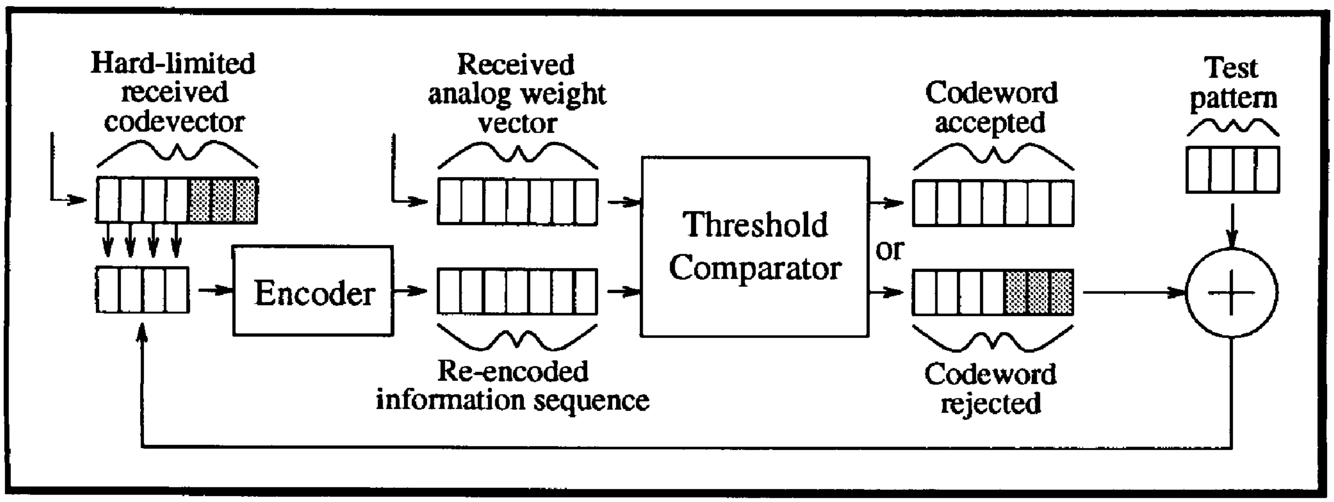

Figure 3. *A block diagram of the decoder implementation.*

The newly formed codeword and the analog weight vector are used to evaluate Equation (1). If the equation is satisfied, the new codeword is indeed that which forms the vector in $\boldsymbol{n}$-dimensional Euclidean space which is central to the $\boldsymbol{n}$-dimensional hyper-cone enclosing the received analog weight vector. Whether or not this was actually the transmitted codeword cannot be definitely discerned; but it is, by design, accepted as such since the reliability criteria have been met. The check bits are then stripped away and the k information bits are retrieved in proper sequence.

If Equation (1) is not satisfied, the codeword in question is rejected. In this case, a series of error patterns $\boldsymbol{E_m}$ (1 # m # 2^k) is used to alter the hard-limited information sequence. Each error pattern is a vector of k bits with a 1 in every position where the information vector is to be changed [2]. The original vector of information bits is added, modulo-2, to an error pattern, forming a new series of information bits each time as shown below (note here that the information sequence is once again being regarded as containing elements in the set {1,0}).

$$\boldsymbol{I_m = I_o} \text{ ¾ } \boldsymbol{E_m}$$

This resultant information vector is then re-encoded, forming a new codeword to be compared against the original analog weight vector.

This process is repeated until either a codeword is found satisfying Equation (1), or it is determined to be disadvantageous to continue. As should be expected, the likelihood of obtaining a correct decoding will decrease with each successive error patter used. However, as can be seen statistically and through computer simulation, the probability of making an erroneous decoding will drop off even more rapidly. The harm, therefore, in carrying out the decoding process thoroughly lies solely in the corresponding decrease in decoder throughput.

There is an obvious upper bound of 2^k possible error patters with which to alter each received codevector in an attempt to obtain the transmitted codeword (as this represents all possible codewords for any (n,k) block coding scheme). Computer simulation has been performed to give an indication of how far to take the decoding process and to determine the most efficient order in which to implement the error patterns. Some of the results of this analysis are included in the next section.

IV. COMPUTER SIMULATION

Software was developed to emulate the communication system illustrated previously in Figure 1. In order to cover a relatively wide spectrum, the performance of this new soft-decision decoding algorithm was analyzed with respect to three diverse linear (n,k) cyclic block encoding schemes: the (7,4) Hamming code with a minimum Hamming distance of 3, the (15,7) BCH code with a minimum Hamming distance of 5, and the (23,12) Golay code with a minimum Hamming distance of 7. Consistent trends were noticed for all three codes.

Data files were compiled for 100,000 randomly generated codevectors for each of the three encoding schemes with values of D ranging from 0.7 to 1.0 and signal-to-noise ratios from 0 dB to 10 dB. Each received codevector was realized as a $(1 \times n)$

dimensional array of randomly generated Gaussian variables with mean ±1 and variance [(n / k) × (1 / (2 × SNR)], where *SNR* represents the signal-to-noise ratio for each bit being transmitted across the communication channel. Consistent with hybrid-ARQ decoding techniques, three possible outcomes were considered for each codevector received: ***C***, indicating a correct decoding; ***E***, indicating an erroneous decoding; or ***R***, if all error patterns failed, indicating that a request for retransmission should be made.

Determining the Most Efficient Error Patterns

Optimally, the decoder being described would successfully decode each and every codevector received which falls within any of the hyper-conic decoding regions discussed in Section III. This is an impractical task, however, for block codes employing codewords of any practical length, since a possible 2^k error patterns exist for each received codeword. Realistically, then, the decoder should implement the error patterns in an order which will result in the most correct decodings at the earliest possible stages of the decoding process, requesting a retransmission if continuing appears to be futile. Knowing the order of the analog weights, certain error patterns become obvious candidates for early implementation. However, the exact order to use is not obvious from inspection .

In order to gain a full view of the behavior of all possible error patterns, every pattern was tested on each codevector received, until either the threshold equation was satisfied or all patterns had been eliminated. From this comprehensive analysis, statistics were compiled revealing the most efficient choice of error patterns to employ, and the order in which to use them.

The relative contributions of the ten most efficient error patterns proved to be almost perfectly consistent for all scenarios tested. According to the simulation, the most efficient ordering of error patterns to be used in conjunction with this decoding algorithm is shown in Figure 4, where the following interpretation applies: Assume a received real-valued information vector with an increasing sequence of absolute values from left to right, such that the left-most information bit is most likely to be in error and the right-most bit is least likely to be.

The performance of several subsets of the error patter sequence shown above is illustrated in Figure 5 for the (23,12) Golay code and the (15,7) BCH code, using values of 1.0 and 0.9 for ***D***. The left scale represents the percentage of correct decodings attained out of the total possible (were all error patterns to be implemented). Any of the subsets might be appropriate depending on the circumstances.

$E_1 = 00000....$ *	$E_5 = 00100....$	$E_9 = 00001....$
$E_2 = 10000....$	$E_6 = 00010....$	$E_{10} = 10010....$
$E_3 = 01000....$	$E_7 = 10100....$	$E_{11} = 11100....$
$E_4 = 11000....$	$E_8 = 01100....$	⋮

**Obviously this error pattern has no effect on the received information vector, and is not actually implemented in the decoding process, but is included here to represent the initial "encode and compare" step required for each received codevector.*

Figure 4. *The most efficient ordering of error patterns to be used in conjunction with the decoding algorithm being discussed.*

Factors affecting the decision of how many error patterns should be employed include codeword length, the value of D being used, and the expected signal-to-noise ratio for each bit being transmitted across the channel. Obviously codes that generate longer codewords will result in a greater variation of possible error patterns than shorter codes. Also, decreasing D eliminates the more scattered codevectors from the decoding regions, serving to decrease the variety of error patterns necessary in implementation. These observations are verified in the plots of Figure 5 (no plots are presented for analysis done with , the (7, 4) Hamming code, or for values of $D < 0.9$, as no more than 4 error patterns were necessary for any of these scenarios).

It can be seen that for signal-to-noise ratios above about 7 dB, implementing any more than 2 error patterns might not be worth the overhead. However, even for expectedly noisy channels the vast majority of correct decodings can be attained using only a few patterns. Even with a bit SNR as low as 0 dB and $D = 1.0$ for the (23,12) Golay code, 97% of all codewords which could possibly be correctly decoded, will be correctly decoded using only 7 error patterns. This represents a *maximum* of only 7 "encode and compare" steps in the decoding process of each received codevector (remember that once Equation (1) is satisfied for a given codevector, further decoding is unnecessary). Even slightly better results were obtained with only 4 error patterns for the shorter BCH code. Likewise, employing only 2 patterns yields comparable results for the (7,4) Hamming code.

V. CONCLUSIONS

An alternative approach to soft-decision decoding for hybrid-ARQ error control communication systems has been presented. The new method bypasses the use of a binary decoder, replacing it by little more than a simple encoder and comparator. The decoder maps a received codevector to a codeword if Equation (1) can be satisfied. This rule allows for correlation decoding employing a reduced codeword searching

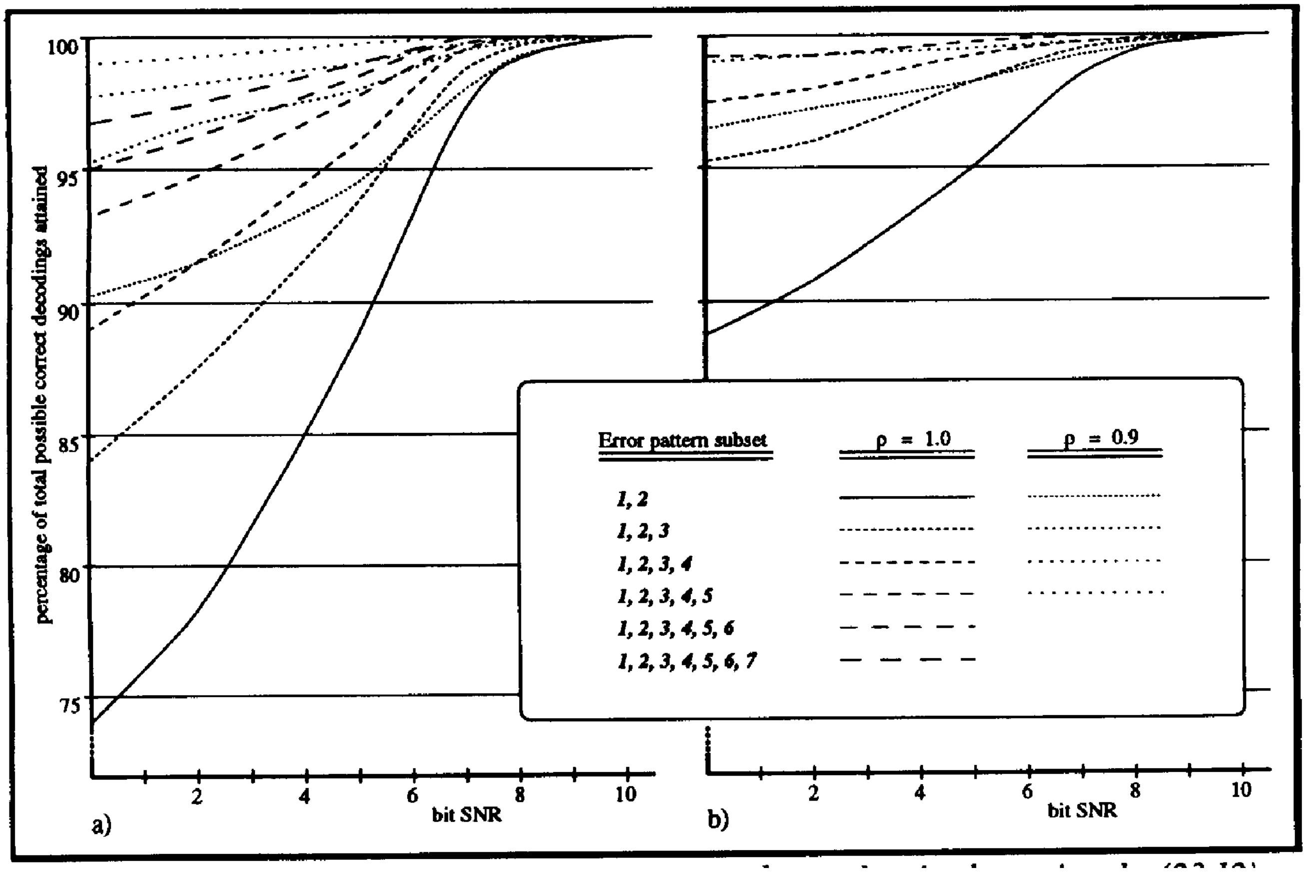

Figure 5. T*he performance of various error pattern subsets when implementing the (23,12) Golay code (a), and the (15, 7) BCH code (b).*

algorithm, further diminishing the complexity of the decoder. The parameter D of this equation is adjustable to provide the reliability-throughput curve desired by the particular system.

Software analysis was performed to emulate this new decoding scheme, verify its potential, and derive the most efficient set of error patterns to be used in the decoding process. Significant results were obtained using only small subsets of these patterns. This decoding algorithm is the subject of ongoing research, and as of yet the relative complexity and performance potential, as compared to contemporary soft-decision schemes, has not been precisely determined. Nonetheless, the current results promise a substantial decrease in decoder complexity without performance loss in implementing this scheme.

ACKNOWLEDGEMENTS

The concept of the alterative soft-decision block decoding scheme presented herein is the brainchild of Dr. Michael Rice, faculty member of the Electrical and Computer Engineering Department at Brigham Young University.

REFERENCES

[1] John G. Proakis. *Digital Communications.* McGraw-Hill, New York, second edition, 1989.

[2] David Chase. "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information". *IEEE Transactions on Information Theory*, IT-18(1):170-182, January 1972.

[3] Hiroshi Miyakawa and Toshinobu Kaneko. "Decoding Algorithm of Error-Correcting Codes by Use of Analog Weights". *Electronics and Communications in Japan*, 58-A(1):18-27, 1975.

[4] Michael Rice. "Performance of Soft-Decision Block Decoded Hybrid-ARQ Error Control". In *Proceedings of the International Telemetering Conference,* Las Vegas, NV, October 1993.

[5] G. David Forey. *Concatenated Codes*. Research Monograph No. 37. M.I.T. Press, Cambridge, Massachusetts, 1966.