Decoding with Multipliers

L. D. Baumert, R. J. McEliece, and G. Solomon[1]
Communications Systems Research Section

We present a general technique, called decoding with multipliers, that can be used to decode any linear code. The technique is applied to the (48,24) quadratic residue code and yields the first known practical decoding algorithm for this powerful code.

I. Introduction

It is widely believed that the next breakthrough in coding technology for a wideband Gaussian channel will come from *soft-decision decoding* of block codes (Ref. 1). Chase (Refs. 1 and 2) has devised an algorithm which allows reasonably efficient soft-decision decoding of any block code for which a hard-decision (i.e., binary) decoding algorithm is known, provided the block length is not too large. This motivates us to find good binary decoding algorithms for powerful short block codes. In this paper we shall describe a technique which is well-suited for this task; it is called *decoding with multipliers.*

In Section II, we define the notion of multiplier; in Section III, we give a general decoding algorithm; and in Section IV we devise a specific algorithm for decoding the powerful (48,24) quadratic residue code—a code whose performance on a Gaussian channel is likely to be very good.

II. Information Sets and Multipliers

Consider a set of j coordinates of an (n,k) linear code C over $GF(q)$. Let X be the set of all distinct j-tuples which appear in these coordinates in at least one of the codewords of C. Linearity guarantees that

$$|X| = q^s$$

for some s, $0 \leq s \leq j$, and establishes a many $(=q^{k-s})$ to one mapping

$$\phi : C \rightarrow X$$

Clearly, given any codeword c the corresponding j-tuple is easily found. Conversely, given any j-tuple of X the q^{k-s} codewords which are its preimages under ϕ can be constructed by means of linear algebra.

Since C has q^k codewords, $|X| \leq q^k$ and so $s \leq k$, independent of j. When $k = s \leq j$, the j coordinates are said to constitute an *information set* for the code since any

[1] Consultant from University of California at Los Angeles.

codeword can be uniquely determined from its values at these j coordinates. All linear codes have information sets of size $j = k$.

We associate with any subset J of the numbers $1, \ldots, n$ its incidence vector m, i.e.

$$m_i = \begin{cases} 1 \text{ if } i \epsilon J \\ 0 \text{ otherwise} \end{cases}$$

for $1 \leq i \leq n$. We shall call such a vector a *multiplier*. For decoding purposes we shall usually be interested in multipliers which specify an information set of the code under consideration; they will be called *proper* multipliers.

III. Decoding with Multipliers

Let C be an (n,k) linear code over $GF(q)$, which is capable of correcting all patterns of e or fewer errors, i.e., one whose minimum distance d satisfies $d \geq 2e + 1$. Let $M = \{m_1, m_2, \ldots, m_N\}$ be a collection of N multipliers for C with the property that for each e-tuple of codeword coordinates there is at least one $m_i \epsilon M$ which is zero at each of these coordinates. Then we can use M to decode C as follows:

Given a received vector $y = (y_1, \ldots, y_n)$—we assume it is a codeword which has suffered $\leq e$ errors—for each $m \epsilon M$ we form the vector $m \times y$ which is defined by

$$(m \times y)_i = \begin{cases} y_i & \text{if } m_i = 1 \\ 0 & \text{if } m_i = 0 \end{cases}$$

Let us temporarily assume the multipliers $m \epsilon M$ are all proper. This means that the vector $m \times y$ can be uniquely extended to a codeword of C. If the vector y contains at most e errors, then by the property of M cited above, at least one of these codewords will be the one transmitted. Thus if we compare y to each of the N (not necessarily distinct) generated codewords, the one closest to y will be the one sent.

If some or all of the multipliers are improper, the decoding procedure is similar, except that in general it will be possible to extend the vectors $m \times y$ to codewords in several ways: if m is a multiplier with $k - s = i$ (we say m has *defect* i), this extension can be done in q^i ways. Thus if M contains N_i multipliers of defect i, then the decoding process will generate $N_0 + qN_1 + q^2N_2 + \ldots$ codewords, each of which must be compared to the received vector y.

In the next section we will apply these general considerations to the (48,24) quadratic residue code.

IV. Multipliers for the (48,24) Quadratic Residue Code

The $(n,k) = (48,24)$ quadratic residue code over $GF(2)$ has minimum distance 12 and so can be used to correct $e = 5$ errors. We would like to find a minimal set of multipliers for this code. Since all 5-tuples of coordinates from $1, \ldots, 48$ must be covered by 0's in some multiplier, in order to minimize $|M|$ we want each multiplier to have as many 0's as possible. On the other hand, we would like the multipliers to be proper; this implies that each multiplier must have at least $k = 24$ nonzero entries. Let us assume then, for the time being, that each multiplier has 24 1's and 24 0's. Thus we have the combinatorial problem of covering all 5-tuples from $1, \ldots, 48$ with 24-tuples in such a way as to use the least possible number of 24-tuples.

While the answer to the above problem is unknown, there is a general result, due to Schönheim (Ref. 3), which provides lower bounds for such questions. In this case, Schönheim tells us that we need at least 62 such 24-tuples. On the other hand, as we shall see, a covering which uses 63 sets is possible. So

$$63 \geq \min |M| \geq 62$$

In terms of decoding effort there is little to choose between 63 and 62 and since there is no guarantee that 62 is even possible, we would be quite content with $|M| = 63$.

A set M of 63 multipliers, whose zeros cover every set of 5 coordinates from $1, \ldots, 48$, is given by the nonzero codewords of a binary (48,6) punctured Solomon–Stiffler (Ref. 4) code. Given any 5 coordinates, there are at most $2^5 = 32$ distinct 5-tuples appearing in these coordinates in the codewords and, by linearity, each 5-tuple which does occur appears equally often. Since this code has dimension $k = 6$ this means that, in particular, 00000 occurs at least twice, and so there is at least one nonzero codeword which has 0's in the desired 5 coordinates. So the nonzero codewords do cover the 0's properly. There are 60 codewords of weight 24 and 3 codewords of weight 32 in this collection. Furthermore, 8 1's can be removed from each of the words of weight 32 without sacrificing the property that they specify information sets for the (48,24) quadratic residue code. Unfortunately, however, in the representations of the Solomon–Stiffler code so far tested not all the words of weight 24 yield proper multipliers for the quadratic residue code. In fact, for the best case yet found, of the 63 multipliers so obtained, 37 are proper, 24 have defect 1, and 2 have defect 2. Thus while $M = 63$ there could be as many as $37 + 2.24 + 4.2 = 93$ codewords to be compared with y in the decoding process. Since any of the 48! coordinate permutations of the Solomon–Stiffler code also has the desired 0's covering property, we conjecture that this bound of 93 can be reduced—but not, of course, below 62.

The complexity of the decoding process is at worst linear in the number of codewords to be compared with y, so 93 (vs 62) represents less than a factor of 2 in decoding time. Thus, if no better multiplier set is discovered, it would be feasible to decode the (48,24) quadratic residue code using this set M.

For definiteness, this multiplier set and the parity check matrix for the (48,24) quadratic residue code are given in the Appendix.

References

1. Baumert, L. D., and McEliece, R. J., "Performance of Some Block Codes on a Gaussian Channel," *Proc. 1975 International Telemetering Conf.*, Washington, D.C., pp. 189–195.
2. Chase, D., "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information," *IEEE Trans. Inform. Th.*, IT-18, pp. 170–182, 1972.
3. Schönheim, J., "On Coverings," *Pacific J. Math.*, 14, pp. 1405–1411, 1964.
4. Solomon, G., and Stiffler, J. J., "Algebraically Punctured Cyclic Codes," *Inform. Contr.* 8, p. 170–179, 1965.

Appendix

The multiplier set discussed above consists of all the nonzero codewords of a particular representation of the Solomon–Stiffler (48,6) code. Since this code is linear it suffices to list generators for the code. These are:

$$g_1 = 110100\ 011011\ 101111\ 111010\ 111000\ 101110\ 110101\ 101111$$

$$g_2 = 101111\ 111110\ 011100\ 011111\ 101111\ 010011\ 001010\ 011101$$

$$g_3 = 010110\ 001010\ 100100\ 000100\ 111111\ 111011\ 010101\ 010001$$

$$g_4 = 010111\ 111000\ 001001\ 101101\ 101100\ 111000\ 000011\ 100110$$

$$g_5 = 010101\ 001101\ 111001\ 010110\ 010101\ 001111\ 000010\ 000111$$

$$g_6 = 000110\ 010010\ 111000\ 000111\ 100010\ 101101\ 111010\ 101011$$

The (48,24) quadratic residue code is cyclic with an overall parity check adjoined. Thus its parity check matrix is a cyclic 47 × 47 matrix with one extra row and column added:

$$\begin{bmatrix}
110010\ 100100\ 110110\ 011000\ 100000\ 000000\ 000000\ 000000 \\
011001\ 010010\ 011011\ 001100\ 010000\ 000000\ 000000\ 000000 \\
\vdots \qquad\qquad \vdots \qquad\qquad\qquad \vdots \\
001010\ 010011\ 011001\ 100010\ 000000\ 000000\ 000000\ 000110 \\
100101\ 001001\ 101100\ 110001\ 000000\ 000000\ 000000\ 000010 \\
111111\ 111111\ 111111\ 111111\ 111111\ 111111\ 111111\ 111111
\end{bmatrix}$$