Transactions Letters

Class of Majority Decodable Real-Number Codes

Jiun Shiu and Ja-Ling Wu

Abstract—**A majority decoding algorithm for a class of real-number codes is presented. Majority decoding has been a relatively simple and fast decoding technique for codes over finite fields. When applied to decode real-number codes, the robustness of the majority decoding to the presence of background noise, which is usually an annoying problem for existing decoding algorithms for real-number codes, is its most prominent property. The presented class of real-number codes has generator matrices similar to those of the binary Reed–Muller codes and is decoded by similar majority logic.**

I. Introduction

ERROR control codes defined over a field of real or complex numbers [1]–[4] could have advantages in some aspects over those defined over a finite field, including: 1) using real/complex operations which are widely available in standard programmable digital signal processors, 2) not restricting to certain block lengths, 3) defining codes which can simultaneously correct errors and reduce data rate [5], and 4) making error control coding more accessible to signal processing engineers. In addition, real-number codes have also been applied to the problem of algorithm-based fault tolerant (ABFT) design [6]–[9]. However, there are problems for real-number codes which are not presented in codes defined over a finite field. For example, in addition to some impulse errors, the elements of a received code vector will also be contaminated by some unavoidable minor errors due to inaccurate computation or channel noise. This background noise is not correctable and will affect decoding procedures. Therefore, either some modifications should be applied to existing decoding algorithms, or new decoding algorithms should be devised specially for real-number codes. In [3], the conventional error trapping decoding technique with a threshold set for identifying if the syndrome equals zero has been shown to be applicable to the class of cyclic DFT codes. In [10], the Berlekamp–Massey algorithm with two thresholds set for testing equality of zero was applied for decoding a class of real-number codes based on the discrete cosine transform (DCT). When these thresholds are properly set, the background noise is implicitly filtered out. Hence, these conventional decoding algorithms could still work in the presence of small error in every received data. Some other unique decoding techniques for real-number codes such as Reed–Solomon voting [1] and rank reduction [2] have also been investigated. In this paper, the first time in the literature, to our best knowledge, a class of majority decodable real-number codes will be presented.

Paper approved by S. G. Wilson, the Editor for Coding Theory and Applications of the IEEE Communications Society. Manuscript received October 20, 1993; revised January 26, 1995.

The authors are with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, 10764, Taiwan, ROC.

Publisher Item Identifier S 0090-6778(96)01777-1.

The first class of majority decodable codes, the Reed–Muller (RM) codes, was discovered by Muller and later Reed devised the decoding algorithm. The Reed algorithm is unusually simple as compared to other decoding algorithms. It recovers the information directly from the received data by some additions and majority logic. No computation for syndromes is required. As a decoding scheme for real number codes, majority decoding algorithms are thus expected to be more robust to the presence of background noise which is usually a major problem of all existing decoding algorithms for real-number codes. These features could also be important when applying real-number codes to design ABFT schemes since any error-correcting operations based on current decoding algorithms for real-number codes might even cost more than the original computation.

There have been some classes of real-number codes which are closely related to codes defined over finite fields. In fact, as proved in [3], for each single-error correcting code over a finite field $GF(p)$, where p is a prime, there exists a single-error correcting real-number code which is defined by the same parity check matrix. This result has been extended to the fact that for any t-error detecting code defined over a prime finite field, there exists a corresponding real number code, with the same generator matrix and the same parity check matrix, whose detectability is at least t [8]. Similarly, it will be shown true in this paper that a class of real-number codes can be defined with generator matrices which are similar to those of the binary RM codes and can be decoded by similar majority logic.

II. The Class of Binary Reed–Muller Codes

As described in [11, sec. 3.7], a binary RM code is defined as follows. The generator matrix for an rth-order RM code of blocklength 2^m, usually denoted as an (r, m) RM code, is defined as an array of blocks

$$G = \begin{bmatrix} G_0 \\ G_1 \\ \vdots \\ G_r \end{bmatrix}$$

where G_0 is the vector of length $n = 2^m$ containing all ones; G_1, an $m \times 2^m$ matrix, has each binary m-tuple appearing once as a column; and G_l has its rows from all possible products[1] of l rows of G_1. It is clear that such an RM code is an $(2^m, k)$ linear block code, where $k = \sum_{i=0}^{r} \binom{m}{i}$ and $\binom{m}{i}$ is a binomial coefficient. A rth-order RM code has minimum distance $2^{m-r} - 1$ and can correct $2^{m-r} - 2$ errors by some majority logic [12]. Since the generator matrix is defined as an array of blocks, it is convenient to write an information vector $a = [a_0, a_1, \cdots, a_{k-1}]$ correspondingly, that is, $a = [A_0, A_1, \cdots, A_r]$. The encoding can then be represented as a sum of some vector-matrix products, i.e.,

$$\begin{aligned} c &= aG && (1)\\ &= A_0G_0 + A_1G_1 + \cdots + A_rG_r. && (2) \end{aligned}$$

As an example, let $m = 4$, $n = 16$, $r = 2$. Then

$$G_0 = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$$

$$G_1 = \begin{bmatrix} 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1\\ 0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1\\ 0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1\\ 0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1\\ 0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&1\\ 0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&1\\ 0&0&0&0&0&0&1&1&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&1\\ 0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1 \end{bmatrix}.$$

For decoding a_{10}, the last information bit in A_2, four check sums are formed

$$\begin{aligned} a_{10}^{(1)} &= v_0 + v_1 + v_2 + v_3\\ a_{10}^{(2)} &= v_4 + v_5 + v_6 + v_7\\ a_{10}^{(3)} &= v_8 + v_9 + v_{10} + v_{11}\\ a_{10}^{(4)} &= v_{12} + v_{13} + v_{14} + v_{15}. \end{aligned}$$

It can be seen that each of the above check sums are a modulo-2 sum of four different received bits. Since $c_0 = a_0$, $c_1 = a_0 + a_4$, $c_2 = a_0 + a_3$, and $c_3 = a_0 + a_3 + a_4 + a_{10}$, $a_{10}^{(1)}$ is clearly equal to a_{10} in the absence of errors. The other three check sums can be similarly checked. If only one error occurs, only one of these check sums will be wrong. Majority voting on these four check sums gives the correct answer.

III. A Majority Decodable Class of Real-Number Codes

The generator matrix G for an rth-order binary RM code, when viewed as a matrix over the real field, can also generate a corresponding real-number code. However, the above majority voting scheme fails to decode this real-number code because the check sums will not equal the desired information digits under the addition defined over the real field. Fortunately, there is an easy way to modify the generator matrix for generating a real-number code which preserves necessary conditions for the code to be majority decodable.

Since every row of the generator matrix has even weight, one can change the sign of half of these nonzero elements in order to produce null sum. Now the received digits for a check sum are selected such that they contribute a_{k-1} (or $-a_{k-1}$) for one and only one time, and every other information digit zero or an even number of times in which half the contributions are of plus sign while the others are of minus sign. Then the check sums are again equal to the desired information digit.

Continuing the above example, a modified generator matrix for the corresponding real-number code is shown at the top of the next page. The relation between G_r and G'_r is clear. It should be noted that there are other ways for constructing G'_r. However, they are all equivalent if permutations of information digits and codeword digits are allowed. The four check sums for a_{10} become

$$a_{10}^{(1)} = v_0 + v_1 + v_2 + v_3 \tag{3}$$
$$a_{10}^{(2)} = -(v_4 + v_5 + v_6 + v_7) \tag{4}$$
$$a_{10}^{(3)} = v_8 + v_9 + v_{10} + v_{11} \tag{5}$$
$$a_{10}^{(4)} = -(v_{12} + v_{13} + v_{14} + v_{15}). \tag{6}$$

If no error occurs, it can be checked again that each of the above four equations will still produce a_{10}. Let us take $a_{10}^{(1)}$ as an example. Since it is evident from G' that

$$\begin{aligned} v_0 &= a_0\\ v_1 &= -a_0 + a_4\\ v_2 &= a_0 + a_3\\ v_3 &= -a_0 - a_3 - a_4 + a_{10}. \end{aligned}$$

The first check sum $a_{10}^{(1)}$ will then be equal to a_{10}. If one error occurs, at most one of the four check sums in (3) will be wrong, a majority-voting algorithm with a threshold value, say T, for the testing of equality between these check sums will find the correct value for a_{10}. The testing of equality between these check sums can be done faster by first sorting them in order and then testing these check sums sequentially to find the major one.

If two unequal errors occur, the voting algorithm still works. Clearly, this voting algorithm will fail when two equal errors or more than two errors occur. Two errors equal or not are also determined by the same threshold T. Hence, this (16, 11) real-number code can correct all patterns of single error and most patterns of double errors. The resultant correcting capacity of this code is quite good although not optimal when considering that there are five parity digits in a codeword.

IV. Discussion and Conclusions

In this letter, a majority-decodable class of real-number codes is presented. It should be noted that this kind of majority voting algorithm has little to do with the voting algorithm proposed by Wolf [1].

For most of existing decoding algorithms for real-number codes, background noise in the received words has always

[1] The product of two vectors is defined as the componentwise multiplications of the two vectors.

$$G'_0 = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

$$G'_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \end{bmatrix}$$

$$G'_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix}.$$

been an annoying problem. Background noise comes from both the computations involving with digital data and imprecise presentation of the elements in generator matrices (encoding) or parity check matrices (decoding), and the quantization of the calculated codewords before transmitted on a digital channel.

For the class of real number codes proposed here, since their generator matrices take only 0, 1, or -1 as elements and are very sparse, the computations for encoding not only are free from the computation error caused by imprecise representation of the coefficients, but also require relatively fewer additions and subtractions. Therefore, the encoding of this class of real-number codes can be done both accurately and fast. On the other hand, the decoding algorithm only needs to calculate 2^{m-r} check sums of the 2^r received digits for a majority voting on these check sums. It can also be performed more accurately and faster than those algorithms that need syndrome computation. Above all, this class of real-number codes suffers less from background noise caused by computation.

Because of the simplicity of both encoding and decoding, if one knows how the information digit is represented (that is, fixed-point or floating-point, precision, etc.), the induced computation error can be easily written out quantitatively. If one knows further how the computed codewords are quantized, the magnitude of background noise presented at the receiving end is known. The threshold T can then be set accordingly to filter out those minor errors caused by the background noise.

For a real-number code derived from an (r, m) RM code, suppose that the information digits are represented as a p-bit fixed-point real number, that is, the information digits are represented with an error 2^{-p-1}. After the addition or subtraction of two digits, the error will become 2^{-p}. From the generator matrix G' presented above, we know that the error of a digit in a codeword will be at most 2^{-p-1+m}. If the codeword is further quantized to q-bit precision where q is usually smaller than $p-m$, the error of a digit in a codeword will be 2^{-q-1}. Hence, we know that the error of computed check sums will be 2^{-q-1+r}. We can then set the threshold T slightly higher than that value to filter out the minor error caused by background noise.

ACKNOWLEDGMENT

The authors wish to acknowledge the comments of the anonymous reviewers.

REFERENCES

[1] J. K. Wolf, "Redundancy, the discrete Fourier transform, and impulse noise cancellation," *IEEE Trans. Commun.*, vol. COM-31, no. 3, pp. 458–461, Mar. 1983.

[2] R. Kumaresan, "Rank reduction techniques and burst error-correction decoding in real/complex fields," in *Proc. 19th Asilomar Conf. Circuits Syst.*, Pacific Grove, CA, Nov. 1985, pp. 457–461.

[3] T. G. Marshall, Jr., "Coding of real-number sequences for error correction: A digital signal processing problem," *IEEE J. Select. Areas Commun.*, vol. SAC-2, no. 2, pp. 381–391, Mar. 1984.

[4] ——, "Codes for error correction based upon interpolation of real-number sequences," in *Proc. 19th Asilomar Conf. Circuits Syst.*, Pacific Grove, CA, Nov. 1985, pp. 202–206.

[5] ——, "Codes and algorithms for simultaneous error correction and rate reduction implementable with standard digital signal processor," in *Proc. IEEE Global Telecommun. Conf.*, Miami, FL, Nov. 1982.

[6] K. H. Huang and J. A. Abraham, "Algorithm-based fault detection for matrix operations," *IEEE Trans. Comput.*, vol. C-33, no. 6, pp. 518–528, June 1984.

[7] C. J. Anfinson and F. T. Luk, "A linear algebraic model of algorithm-based fault tolerance," *IEEE Trans. Comput.*, vol. 37, no. 12, pp. 1599–1604, Dec. 1988.

[8] V. S. S. Nair and J. A. Abraham, "Real-number codes for fault-tolerant matrix operations on processor arrays," *IEEE Trans. Comput.*, vol. 39, no. 4, pp. 426–435, Apr. 1990.

[9] A. L. N. Reddy and P. Banerjee, "Algorithm-based fault detection for signal processing application," *IEEE Trans. Comput.*, vol. 39, no. 10, pp. 1304–1308, Oct. 1990.

[10] J.-L. Wu and J. Shiu, "Real-valued error control coding by using DCT," in *Proc. Inst. Elec. Eng.*, vol. 139-I, no. 2, pp. 133–139, Apr. 1992.

[11] R. E. Blahut, *Theory and Practice of Error Control Codes.* Reading, MA: Addison-Wesley, 1983.

[12] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications.* Englewood Cliffs, NJ: Prentice-Hall, 1983.