Computing the Spectrum of a Binary Group Code*

By M. M. BUCHNER, JR.

(Manuscript received December 10, 1965)

The weight distribution of the code vectors of a binary group code has been referred to as the spectrum of the code. This paper presents a technique for calculating the spectrum of such a code, the spectra of shortened codes obtainable from the code, and what are defined as the level weight structures of the code.

The method is conceptually straightforward and readily adaptable to digital computers. It involves operations no more complex than the addition of two $(n - k)$-tuples, the determination of the weight of certain $(n - k)$-tuples, and the ordinary addition of certain integers. Its computational complexities are independent of the code parameters. In principle, it may be used for any binary group code, but it is particularly useful for codes in which the number of parity check positions per code vector is rather small although the number of information positions may be large.

I. INTRODUCTION

The need for reliable data transmission systems has prompted the investigation of various coding techniques which attempt to detect and/or correct transmission errors. Because of the relative ease with which binary codes can be implemented, these codes have received special attention. It is with certain properties of these codes that this paper is concerned.

In general, the encoder receives a block of k binary symbols (called a message) from a message source from which it determines $(n - k)$ binary parity check symbols (called an ending). The message symbols and the ending symbols may be interleaved or transmitted sequentially thus forming a block of length n (called a code vector). Because any

* The material presented in this paper formed Appendix II of the dissertation "Coding for Numerical Data Transmission" submitted by the author to The Johns Hopkins University in conformity with the requirements for the degree Doctor of Philosophy.

code in which these symbols are interleaved is equivalent[1] to a code in which the message and ending are transmitted sequentially, attention may be restricted to the latter situation.

The elements 0 and 1 form a field. Two vectors (or n-tuples) whose components are these field elements may be added by adding modulo 2 the corresponding components of each vector. The symbol $\oplus$ will be used to denote this addition of vectors.

The set of all possible n-tuples forms a vector space V_n of dimension n over the field of two elements. A subset V is said to form a group code if the n-tuples in the subset form a group. Over the field of two elements, a set of vectors that forms a group is a subspace of V_n. Therefore, the vectors of a group code form a subspace of V_n.

The weight of a vector u is the number of nonzero components in u and is denoted by $w[u]$. The distance[1] between two code vectors u and v is $w[u \oplus v]$. Because the code vectors form a group, there exists a code vector $t = u \oplus v$. The distance between u and v is thus equal to $w[t]$.

Because of this relationship between code vector weights and distances between code vectors, it is useful in evaluating the error detecting and/or correcting capabilities of group codes to be able to determine the number of code vectors of each possible weight — i.e., from 0 to n. This information has been called the spectrum of a code and can in principle be obtained by calculating in detail each of the possible 2^k code vectors and then determining the weight of each of these code vectors. However, this method is not computationally feasible for values of k which are most often of interest.

MacWilliams[2] has determined a system of linear equations which relate the set of integers that forms the spectrum of a given code to the set of integers that forms the spectrum of its dual code. The method is particularly effective for codes in which the dimension of the dual code is relatively small so that the spectrum of the dual code is readily obtained.

The method presented herein enables the direct computation of the spectrum of a code without the actual formation of every code vector. The technique also gives both the spectrum of each of the possible shortened codes which may be obtained from the given code and the level weight structures of the given code. The level weight structures (which are defined later in this paper) have proved useful in the study of the effectiveness of error-correcting codes for numerical data transmission[3] and may indeed be of interest in other areas of code evaluation.

The method is conceptually quite simple, readily implemented on a digital computer, and does not depend upon the solution of any equa-

tions. In fact, the only operations involved are the component by component modulo 2 addition of $(n - k)$-tuples, the determination of the weight of certain $(n - k)$-tuples and the ordinary addition of certain integers.

II. COMPUTATIONAL TECHNIQUE

Let k denote the dimension of the code space V and let E_j $(1 \leqq j \leqq k)$ denote the k basis vectors of V. Take E_j in the usual systematic form

$$E_j = e_j \mid C_j \tag{1}$$

where the message e_j is the k-tuple with a 1 in position j and all other positions 0 and C_j is the $(n - k)$-tuple ending assigned to the message e_j. Note that if the code is specified by a parity check matrix[1] in the form

$$H = (h_1 h_2 \cdots h_k I_{n-k}) \tag{2}$$

where h_i $(1 \leqq i \leqq k)$ is the column of H in the ith position and I_{n-k} is the $(n - k) \times (n - k)$ identity matrix, then C_j is simply the transpose of h_j and E_j is readily obtainable.

The vectors $E_1, E_2, \cdots, E_l$ generate a subspace of V of dimension l which we shall denote as Γ_l. Γ_k is the code itself and Γ_l is the set of code vectors in which information positions $l + 1, l + 2, \cdots, k$ are 0. Γ_0 is defined as consisting exclusively of the all 0 code vector.

Let $\Lambda_l = \Gamma_l - \Gamma_{l-1} \cdot \Lambda_l$, which is called the l-level of the code, is the set of code vectors in which information positions $l + 1, l + 2, \cdots, k$ are 0 and information position l is 1. Any code vector in Λ_l is the sum of E_l and some vector in Γ_{l-1}.

The basic idea is to form for Γ_l an ending-weight matrix $S^{(l)}$. For convenience we shall deviate from usual practice and number the rows and columns of $S^{(l)}$ beginning with 0. The entry $s_{\alpha,\beta}{}^{(l)}$ in row α and column β of $S^{(l)}$ denotes the number of code vectors in Γ_l of weight α whose endings are the $(n - k)$-bit binary representation of β (denoted by $B(\beta)$). There must be $(n + 1)$ rows in $S^{(l)}$ to allow for all possible code vector weights and 2^{n-k} columns to allow for all of the possible $(n - k)$-bit endings. Therefore, $S^{(l)}$ is an $(n + 1) \times 2^{n-k}$ matrix.

The utility of this technique lies in the ease with which $S^{(l)}$ may be obtained from $S^{(l-1)}$. Suppose that $S^{(l-1)}$ is known. The code vectors of Λ_l are formed by adding E_l to the code vectors of Γ_{l-1}. However, the special form of E_l makes this operation equivalent to placing a 1 in information position l of each vector in Γ_{l-1} and, at the same time, adding C_l to the ending of each code vector in Γ_{l-1}.

Any code vector of weight α in Γ_{l-1} whose ending is $B(\beta)$ becomes a vector in Λ_l with ending $B(\beta) \oplus C_l$ and of weight γ where

$$\gamma = \alpha + 1 + w[B(\beta) \oplus C_l] - w[B(\beta)]. \tag{3}$$

For those values of α $(0 \leqq \alpha \leqq n)$ and γ $(0 \leqq \gamma \leqq n)$ for which (3) may be satisfied,

$$s_{\gamma, B^{-1}[B(\beta)\oplus C_l]}^{(l)} = s_{\gamma, B^{-1}[B(\beta)\oplus C_l]}^{(l-1)} + s_{\alpha,\beta}^{(l-1)} \tag{4}$$

where B^{-1} (the inverse of B) is the operator such that $\psi = B^{-1}B[\psi]$.

In general, it is not possible to satisfy (3) for every γ $(0 \leqq \gamma \leqq n)$. However, because all code vectors in Γ_{l-1} whose endings are $B(\beta)$ (i.e., the code vectors giving rise to the nonzero entries in column β of $S^{(l-1)}$) become code vectors in Λ_l of weight in the range 0 through n, all values of α corresponding to nonzero entries in column β of $S^{(l-1)}$ produce values of γ such that $0 \leqq \gamma \leqq n$. For these values of γ, (4) may be applied.

On the other hand, values of γ which would require values of α outside of the range $0 \leqq \alpha \leqq n$ in order to satisfy (3) are those values of γ for which it is impossible to have code vectors in Γ_{l-1} of weight α whose endings are $B(\beta)$. For these values of γ,

$$s_{\gamma, B^{-1}[B(\beta)\oplus C_l]}^{(l)} = s_{\gamma, B^{-1}[B(\beta)\oplus C_l]}^{(l-1)}. \tag{5}$$

The column numbers referred to in (4) and (5) are independent of α. Furthermore, as α increases, (4) and (5) simply refer to different elements in the same column. For this reason, these results may be expressed as column operations thus leading to a conceptually simple result.

Let $s_\beta^{(l)}$ denote column β in $S^{(l)}$. Define $\sigma^{(j)}$ to be a shifting operator which, when applied to $s_\beta^{(l)}$, shifts each element of $s_\beta^{(l)}$ by j positions filling in any resulting blank positions with zeros. For example, if

$$s_\beta^{(l)} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

then

$$\sigma^{(2)} \cdot s_\beta^{(l)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

and

$$\sigma^{(-1)} \cdot s_\beta^{(l)} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

In terms of $\sigma^{(j)}$, the relationships expressed in (3), (4), and (5) may be conveniently expressed as

$$s_{B^{-1}[B(\beta)\oplus C_l]}^{(l)} = s_{B^{-1}[B(\beta)\oplus C_l]}^{(l-1)} + \sigma^{(w[B(\beta)\oplus C_l]-w[B(\beta)]+1)} \cdot s_\beta^{(l-1)}. \quad (6)$$

Clearly all columns of $S^{(l)}$ are obtained by successively applying (6) as β runs from 0 to $2^{n-k} - 1$.

It is important to notice the great simplicity of (6). In practice, it involves shifting one column of $S^{(l-1)}$ and then combining by simple addition the elements of this column with those of another column of $S^{(l-1)}$ to obtain a column of $S^{(l)}$. Determining the number of positions that $s_\beta^{(l-1)}$ should be shifted and the column with which $s_\beta^{(l-1)}$ should be combined is extremely easy. In particular, the operations in (6) are readily adapted to digital computer operations.

If $S^{(0)}$ is known, the remaining ending-weight matrices can be successively obtained. The only code vector in Γ_0 is the all 0 vector. Therefore,

$$s_{00}^{(0)} = 1$$

and

$$s_{\alpha,\beta}^{(0)} = 0$$

for all other values of α and β.

Now that the method for constructing the ending-weight matrices has been presented, the following will serve to indicate how the desired information is extracted.

(*i*) Spectrum of the code: The total number of code vectors of weight α in the code is

$$\sum_{\beta=0}^{2^{n-k}-1} s_{\alpha,\beta}^{(k)}. \quad (7)$$

The spectrum of the code is obtained from $S^{(k)}$ by using (7) for each value of $\alpha (0 \leqq \alpha \leqq n)$.

(*ii*) Spectra of shortened codes: Let k' denote the number of information positions in the shortened code — i.e., $k - k'$ information positions are deleted. Assume that the deleted posi-

tions are information positions $k' + 1, k' + 2, \cdots, k$. The total number of code vectors of weight α in the shortened code is

$$\sum_{\beta=0}^{2^{n-k}-1} s_{\alpha,\beta}{}^{(k')}. \tag{8}$$

The spectrum of the shortened code is obtained from $S^{(k')}$ by using (8) for each value of $\alpha (0 \leqq \alpha \leqq n - \mathrm{k} + k')$.

(*iii*) Level weight structure: The set of code vectors Λ_l has been referred to as the l-level code vectors and the weight distribution of these code vectors as the l-level weight structure.[3] Note that the l-level weight structure is the difference between the spectrum of the shortened code consisting of l information positions and the spectrum of the shortened code consisting of $(l - 1)$ information positions.

Let $n_{l,\alpha}$ denote the number of code vectors of weight α on the l-level. The number of code vectors of weight α in Γ_{l-1} is

$$\sum_{\beta=0}^{2^{n-k}-1} s_{\alpha,\beta}{}^{(l-1)}.$$

Similarly, the number of code vectors of weight α in Γ_l is

$$\sum_{\beta=0}^{2^{n-k}-1} s_{\alpha,\beta}{}^{(l)}.$$

It follows that

$$n_{l,\alpha} = \sum_{\beta=0}^{2^{n-k}-1} s_{\alpha,\beta}{}^{(l)} - \sum_{\beta=0}^{2^{n-k}-1} s_{\alpha,\beta}{}^{(l-1)}. \tag{9}$$

III. CONCLUSIONS

The spectrum of any group code, the spectrum of any shortened code, and all level weight structures are obtainable in a straightforward manner by means of operations no more complex than the addition of two $(n - k)$-tuples (to determine the columns to combine), the computation of the weight of certain $(n - k)$-tuples, and the repeated addition of integers two at a time (to actually combine the columns). The number of computations does depend upon the parameters n and k but the method has the advantage that the complexity of the operations is invariant. Because the number of computations and the number of computer storage locations required for the ending-weight matrix are sensitive to changes in $(n - k)$ but rather insensitive to changes in k, the method is most effective for codes in which $(n - k)$ is moderate although k may be quite large.

As presented, the method treats each of the k ending-weight matrices in a similar manner by determining all of the $(n + 1)\cdot 2^{n-k}$ entries of each matrix. Computing time can be saved by realizing that the maximum possible weight of a vector in Γ_l is $l + n - k$ and, thus, that it is only necessary to compute the first $l + n - k$ rows of $S^{(l)}$ because the remaining rows contain zero entries exclusively. Additional programming sophistications, including processing only those columns of $S^{(l-1)}$ which contain nonzero entries in obtaining $S^{(l)}$ (particularly for the smaller values of l), improve the computing efficiency of the method.

This technique was originally developed for computing the level weight structures of certain codes. Thus, if the level weight structures and/or the spectra of the shortened codes are desired, this method offers a straightforward and effective means of obtaining such information. However, if all that is desired is the spectrum of the code, then under some conditions the method developed by MacWilliams[2] may be preferable from a computing time point of view although the conceptual simplicity of this method is still appealing. In any case, the relative advantages of the two methods should be considered before deciding which to use for a specific application.

The method has been used successfully to compute the level weight structures and the spectra of the (15,11), (31,26), and (63,57) Hamming perfect single error-correcting codes. In each case the information was obtained on an IBM 7094 digital computer in less than 0.01 hours.

IV. NUMERICAL EXAMPLE

The parity check matrix for a (7,4) Hamming code is

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

The basis vectors for this code are

$$\begin{aligned} E_1 &= 1000 \quad 111 \\ E_2 &= 0100 \quad 110 \\ E_3 &= 0010 \quad 101 \\ E_4 &= \underbrace{0001}\ \underbrace{011} \\ &\qquad \text{message} \quad \text{ending.} \end{aligned}$$

The sets of code vectors referred to as Γ_3 and Λ_4 are listed in Table I. The appropriate values of α and β are given next to each vector.

TABLE I

Γ_3	α	β	A_4	α	β
0000 000	0	0	0001 011	3	3
1000 111	4	7	1001 100	3	4
0100 110	3	6	0101 101	4	5
1100 001	3	1	1101 010	4	2
0010 101	3	5	0011 110	4	6
1010 010	3	2	1011 001	4	1
0110 011	4	3	0111 000	3	0
1110 100	4	4	1111 111	7	7

Tabulating this information yields $S^{(3)}$ and $S^{(4)}$.

$S^{(3)}:\alpha$ \ β

	0	1	2	3	4	5	6	7
0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	1	1	0	0	1	1	0
4	0	0	0	1	1	0	0	1
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

$S^{(4)}:\alpha$ \ β

	0	1	2	3	4	5	6	7
0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	1	1	1	1	1	1	1	0
4	0	1	1	1	1	1	1	1
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	1

We now turn to use the method herein developed to obtain $S^{(4)}$ from $S^{(3)}$. Specifically, we use (6) first with $\beta = 0$ and then successively increase β until $\beta = 7$.

When $\beta = 0$, $B^{-1}[B(0) \oplus C_4] = 3$. Thus, (6) reduces to

$$s_3^{(4)} = s_3^{(3)} + \sigma^{(3)} \cdot s_0^{(3)} = \begin{bmatrix}0\\0\\0\\0\\1\\0\\0\\0\end{bmatrix} + \begin{bmatrix}0\\0\\0\\1\\0\\0\\0\\0\end{bmatrix} = \begin{bmatrix}0\\0\\0\\1\\1\\0\\0\\0\end{bmatrix}$$

which is indeed correct.

Now let $\beta = 1$. Then $B^{-1}[B(1) \oplus C_4] = 2$ so (6) yields

$$s_2^{(4)} = s_2^{(3)} + \sigma^{(1)} \cdot s_1^{(3)} = \begin{bmatrix}0\\0\\0\\1\\0\\0\\0\\0\end{bmatrix} + \begin{bmatrix}0\\0\\0\\0\\1\\0\\0\\0\end{bmatrix} = \begin{bmatrix}0\\0\\0\\1\\1\\0\\0\\0\end{bmatrix}.$$

The remaining columns of $S^{(4)}$ are obtained in a like manner as β increases to 7.

The spectrum of the code is obtained by summing across the rows of $S^{(4)}$. The spectrum of the shortened code resulting from the deletion of the fourth information position is obtained by summing across the rows of $S^{(3)}$. The 4-level weight structure is the difference between these spectra. This information is tabulated in Table II.

TABLE II

Weight	Code Spectrum	Shortened Code Spectrum	4-Level Weight Structure
0	1	1	0
1	0	0	0
2	0	0	0
3	7	4	3
4	7	3	4
5	0	0	0
6	0	0	0
7	1	0	1

For an illustrative example, it was necessary to confine ourselves to a code in which k is small. However, it should be realized that the true utility of the method lies in the fact that it can, without modification or additional complexity, be used for codes in which k is quite large.

REFERENCES

1. Peterson, W. W., *Error Correcting Codes,* M.I.T. Press and John Wiley and Sons, 1961.
2. MacWilliams, J., A Theorem on the Distribution of Weights in a Systematic Code, B.S.T.J., *42,* January, 1963, pp. 79–94.
3. Buchner, M. M., Jr., Coding for Numerical Data Transmission, Ph.D. Dissertation, The Johns Hopkins University, Baltimore, Maryland, 1965.