the checked variances. For the Preparata code we only determined the word-error probability, which is displayed in the last column of Table IV. The error probabilities are the same for each of the tested implementations, as they all perform complete soft-decision decoding.

Impl. 4 requires a number of column patterns to be stored. The maximum (max.) and the average (av.) number of patterns that have been stored during the simulation process depend on the code, its length, and on σ^2. The number of stored column patterns is shown in Table V.

ACKNOWLEDGMENT

The authors wish to thank the anonymous referees for their useful suggestions.

REFERENCES

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

[2] J. Snyders and Y. Be'ery, "Maximum likelihood soft decoding of binary block codes and decoders for the Golay codes," *IEEE Trans. Inform. Theory*, vol. 35, pp. 963–975, Sept. 1989.

[3] J.H. Conway and H.J.A. Sloane, "Soft decoding techniques for codes and lattices, including the Golay code and the Leech lattice," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 41–50, Jan. 1986.

[4] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 170–182, Jan. 1972.

[5] S.N. Litsyn, "Fast decoding algorithm of Reed–Muller codes," in *Proc. Fourth Joint Swedish–Soviet Int. Workshop Inform. Theory*, 1989, pp. 288–291.

[6] P.A.H. Bours, "Soft decision decoding," Master's thesis, Eindhoven Univ. of Technol., Eindhoven, The Netherlands, 1989.

[7] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes.* Amsterdam, The Netherlands: North Holland, 1977.

Bounds on the Trellis Size of Linear Block Codes

Yuval Berger and Yair Be'ery

***Abstract*— The size of minimal trellis representation of linear block codes is addressed. Two general upper bounds on the trellis size, based on the zero-concurring codewords and the contraction index of the subcodes are presented. The related permutations for attaining the bounds are exhibited. These bounds evidently improve the previous published general bound. Additional bounds based on certain code constructions are derived. We focus on the squaring construction and obtain specific constructive bounds for Reed–Muller and repeated-root cyclic codes. In particular, the recursive squaring construction of Reed–Muller codes is explored and the exact minimal trellis size of this design is obtained. Efficient permutations, in the sense of the trellis size, are also demonstrated by using shortening and puncturing methods. The corresponding bounds are specified.**

***Index Terms*—Trellis, maximum-likelihood decoding, soft-decision decoding, zero-concurring codewords, contraction index.**

Manuscript received June 12, 1991; revised January 6, 1992. This work was presented in part at the 21st Annual IEEE Communication Theory Workshop, Rhodes, Greece, June 30–July 6, 1991.

The authors are with the Department of Electrical Engineering-Systems, Tel-Aviv University, Tel-Aviv 69978, Israel.

IEEE Log Number 9203030.

I. INTRODUCTION

Trellis diagrams have been traditionally exploited for decoding convolutional codes. Wolf introduced in [1] a trellis design for representing linear block codes. The Viterbi algorithm was applied then for soft decision decoding of these codes. Forney [2] further analyzed the trellis diagram and utilized it for efficient decoding of some block codes. Muder [3] provided a formalization of the trellis design for block codes based on graph–theoretic approach.

Let $c = (c_1, c_2, \cdots, c_n)$ be an n-tuple over GF(q). Denote $c_p^{(i)} \triangleq (c_1, c_2, \cdots, c_i)$, $c_f^{(i-1)} \triangleq (c_i, c_{i+1}, \cdots, c_n)$, $i = 1, 2, \cdots, n$. For an (n, k) code C over GF(q), let $C_p^{(i)}$ be the linear code which consists of $c_p^{(i)}$ for all c such that $c \in C, c_f^{(i)} = (0, 0, \cdots, 0)$. Similarly, let $C_f^{(i)}$ be the linear code that consists of $c_f^{(i)}$ for all c such that $c \in C$, $c_p^{(i)} = (0, 0, \cdots, 0)$. Denote the dimensions of these codes by p_i and f_i respectively. Define $p_0 \triangleq 0, f_n \triangleq 0$. The trellis diagram for C consists of $n+1$ levels $V_i, i = 0, 1, \cdots, n$. Each level includes $|V_i|$ states. V_0 and V_n consist of a single state each, referred as the *initial* and *final* states, respectively. Each branch in the trellis connects states from successive levels and labeled by $0, 1, \cdots, q-1$. A path from level V_0 to level V_i represents $c_p^{(i)}$ for some $c \in C$. A trellis is called *minimal* if $|V_i|$ is minimal among all trellis representations of C for $i = 0, 1, \cdots, n$. The dimension of the *state space* at level i for minimal trellis is denoted by $s_i \triangleq \log_q |V_i|$. It is well known [2], [3] that the general trellis design for linear codes described by Forney [2] is minimal and unique, and s_i is identical in both C and its dual code. Hence, for a minimal trellis

$$s_i = k - p_i - f_i.$$

The *minimal trellis size* index, defined as $s \triangleq \max_i(s_i)$, depends on the order of the code coordinates [2], [3], i.e., the order of the columns in the generator matrix G or the parity check matrix H. The operation of reordering the columns of G or H is referred in the sequel as *permutation*. A slightly different parameter $s(C)$ is defined as the minimal attainable s index for *any* permutation of C, namely for any *equivalent* code of C. $s(C)$ is called the *absolute minimal trellis size*. A trellis of which $s = s(C)$ is called an *absolute minimal trellis*.

Clearly $s(C)$ reflects the decoding complexity of the code with respect to general decoding algorithms such as in [1]. Forney [2] recently introduced an alternative approach where the trellis is decoded in a section-by-section strategy. The sections themselves are efficiently decoded by a precomputation stage based on some type of "fast algorithms" schemes. The decoding complexity of such algorithm depends on the dimension of the state space s_i at the sections boundaries and the *branch complexity* in the sections. However, the choice of the sections boundaries for efficient decoding is involved with the code structure, such as in the four-section design of Reed–Muller (RM) codes [2], and generally unknown especially for long codes. Nevertheless, it appears from [2] that "good" permutations, in sense of $s(C)$ (which reflects the branch complexity at *every* bit level), are also suitable for section-by-section decoding, e.g., RM codes and the Golay (24, 12) code. Such permutations may as well be a key for designing an efficient trellis-based decoder. The corresponding trellis size parameter is "an intuitive measure of the decoding complexity of the code and appears to be a fundamental descriptive characteristic" [3]. It can be used for either comparing different permutations of the same code or

comparing the decoding complexity of different codes with respect to trellis-based decoding schemes.

Several bounds on $s(C)$ were derived in [1]–[3]. The general Wolf bound [1] is

$$s(C) \leq \min(k,\, n-k).$$

Specific code constructions were utilized in [2] to improve this bound for the corresponding codes. In particular, the *cubic construction* for the binary Golay $(24, 12, 8)$ code evidently results in an absolute minimal trellis diagram since $s = s(\text{Golay}) = 9$ [3]. Similarly the *Squaring Construction* (SC) for RM codes yields a smaller trellis size then implied by (2). In the latter case, the trellis size index for the $(16, 5)$ and $(16, 11)$ codes was computed and indeed exhibited an improvement over (2).

In this study, we address particular permutations which lead to decreased trellis size and imply improved bounds on $s(C)$. Two general permutations and upper bounds are presented in Section II. The first is based on the *zero-concurring* (ZC) codewords that are defined and studied in [4], [5]. The concept of *contraction* is utilized for the derivation of the second permutation. This concept was introduced by Snyders and Be'ery [6] as part of a coset decoding algorithm, and further studied in [7]. In Section III, we address classes of codes whose structure is based on prescribed constructions. We focus in particular on the SC which has been utilized for constructing various codes such as Reed–Muller and repeated-root cyclic codes [2], [8], [9]. Upper and lower bounds on the trellis size of these codes are derived. Particularly the iterated SC design for RM codes [2] is explored and explicit formula of the minimal trellis size for this design is derived. This formula is used to bound $s(C)$ for RM and related BCH codes as well. Advantageous permutations and improved bounds are also obtained with the aid of shortening and puncturing methods.

II. General Upper Bounds on $s(C)$

In this section, we present two general improvements over the Wolf bound (2). The first is based on the ZC codewords. The second is based on the contraction concept which is in a sense a generalization of the ZC property.

The ZC codewords [4] are defined as follows. In any coordinate of the code, at most one of the ZC codewords has a nonzero component. These codewords are utilized for decoding algorithms in [4]. Vardy and Be'ery [5] study the problem of finding such words. They also provide the number of ZC codewords for many codes. Let $v^{(1)}, v^{(2)}, \cdots, v^{(J)}$ denote J zero-concurring codewords. Denote by $I_1, I_2, \cdots, I_J$ the sets of coordinates of the nonzero components in $v^{(1)}, v^{(2)}, \cdots, v^{(J)}$ respectively, i.e., $i_j^m \in I_j$ iff $v_{i_j^m}^{(j)} \neq 0$, $j = 1, 2, \cdots, J$, $m = 1, 2, \cdots, |I_j|$. I_0 is the set of coordinates in which all the ZC codewords have zero components. The following permutation orders the coordinates according to I_j.

Definition 1: Let a code contain J zero-concurring codewords. Define the permutation P_{ZC} by the following transformation on any coordinate i_j^m,

$$i_j^m \xrightarrow{P_{\text{ZC}}} \begin{cases} m, & j = 0. \\ \sum_{t=0}^{j-1} |I_t| + m, & j = 1, 2, \cdots, J. \end{cases}$$

Example 1: Suppose a cyclic ternary code contains the codeword (1002001000). Clearly this word and the two codewords obtained by bit rotation are ZC. Consequently the equivalent words defined by P_{ZC} are (0121000000), (0000121000), (0000000121).

Theorem 1: Suppose an (n, k, d) code C contains J zero-concurring codewords and its dual code contains $J^\perp$ zero-concurring codewords. Then,

$$s(C) \leq \min\Big(k - J + 1,\, n - k - J^\perp + 1\Big).$$

Proof: Consider the equivalent code $\tilde{C}$ defined by the permutation P_{ZC} on C. Obviously the ZC codewords are pairwise linearly independent. Assume that they are rows of the generator matrix G. For any position i of $\tilde{C}$, $i = 0, 1, \cdots, n$, $p_i + f_i \geq J - 1$. Then, from (1), $s_i \leq k - J + 1$. By Forney ([2], Appendix A), s_i is identical in $\tilde{C}$ and $\tilde{C}^\perp$. Applying the same argument for $\tilde{C}^\perp$ yields $s_i \leq n - k - J^\perp + 1$. □

Evidently, this bound improves (2), provided that $J > 1$ or $J^\perp > 1$. Be'ery and Synders proved in [4] that for RM(r, m) codes the maximal number of ZC codewords is $J = 2^r$. This implies an explicit upper bound on s(RM) (see Section III for improved bound for RM codes). A list of known values of J for various BCH codes and algorithms for locating the ZC codewords (and the required permutation P_{ZC}) are presented in [5].

Example 2: The BCH $(15, 7, 5)$ code contains $J = 3$ zero-concurring codewords [5]. From Theorem 1, $s(\text{BCH}(15, 7, 5)) \leq 5$. This bound is evidently tighter then the Wolf bound (2) which yields 7.

Example 3: The BCH $(63, 30, 13)$ code contains $J = 4$ zero-concurring codewords [5], which implies by Theorem 1 that $s(\text{BCH}(63, 30, 13)) \leq 27$. The previous bound (2) is 30.

Example 4: Consider the cyclic code $C(21, 15, 4)$ with roots $\{0, 3, 7\}$. It is well known [10, p. 196] that the dual code of C is equivalent to the cyclic code $C_1(21, 6)$ with roots $\{1, 5, 9\}$ generated by $g(x) = 1 + x^2 + x^4 + x^5 + x^8 + x^9 + x^{10} + x^{12} + x^{13} + x^{14} + x^{15}$. C_1 contains the following ZC codewords (found by a computer search):

$$\begin{pmatrix} 100100100100100100100 \\ 010010010010010010010 \\ 001001001001001001001 \end{pmatrix}.$$

Hence, from Theorem 1, $s(C(21, 15, 4)) \leq 6 - 3 + 1 = 4$. The previous bound (2) yields 6.

Let $C(n, k)$ be a code with generator matrix G. Let C_1 be a subcode of C with dimension J_ν generated by G_1. Denote by $\tilde{G}_1$ a matrix which consists of maximal number of pairwise linearly independent columns of G_1 over GF(q). The code generated by $\tilde{G}_1$, denoted as $\tilde{C}_1$, is called a *contracted code* of C_1 or a *contracted subcode* of C. $\tilde{C}_1$ is of dimension J_ν and length $J_\nu + \nu$ (or $J_\nu + \nu + 1$ when $\tilde{G}_1$ incudes a zero column), hereby the *contraction index* ν is defined. The contraction concept was introduced in [6] as part of a *coset decoding* method for block codes. The problem of estimating the maximal J_ν was further analyzed in [7] where several bounds on the contraction index were derived.

Theorem 2: If a subcode of dimension J_ν and contraction index ν exists for code C, $\nu \leq J_\nu - 1$, and similarly a subcode of dimension $J_{\nu\perp}$and contraction index $\nu^\perp$ exists for the dual code $C^\perp$, $\nu^\perp \leq J_{\nu\perp} - 1$, then

$$s(C) \leq \min\Big(k - J_\nu + \nu + 1, n - k - J_{\nu\perp} + \nu^\perp + 1\Big).$$

Proof: Consider a permutation which changes C into $\tilde{C}$ with generator matrix $\tilde{G}$ such that

$$\tilde{G} = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix}^{\} J_\nu},$$

$$G_1 = \left(b_0 b_0 \cdots b_0,\, b_1^1 b_1^2 \cdots b_1^{n_1}, \cdots, b_r^1 b_r^2 \cdots b_r^{n_r}\right),$$

where $r = J_\nu + \nu$ and b_{j1}^u, b_{j2}^t represent either pairwise linearly dependent columns (for $j_1 = j_2$) or pairwise linearly independent

columns ($j_1 \neq j_2$) of G_1, $j_{1,2} \in \{1, 2, \cdots, r\}$. b_0 is the possible zero column. Denote by $C_{1-}^{(i)}$, $C_{1+}^{(i)}$ the codes whose generator matrices consist of columns at positions 1, $2, \cdots, i$, and $i+1$, $i+2, \cdots, n$ of C_1, respectively. There are two distinguishable cases.

1) If the column at position i of G_1 is b_j^u, $j = 0, 1, \cdots, J_\nu$ $(1 \leq i \leq n)$, then there are j pairwise linearly independent columns in the generator matrix of $C_{1-}^{(i)}$. Thus, the dimension of $C_{1-}^{(i)}$ is at most j, and the dimension of the span of codewords that are all-zero at the first i positions is at least $J_\nu - j$, i.e., $f_i \geq J_\nu - j$. Similarly, the dimension of $C_{1+}^{(i)}$ is at most $\min(J_\nu, J_\nu + \nu + 1 - j)$, therefore $p_i \geq J_\nu - \min(J_\nu, J_\nu + \nu + 1 - j)$. From (1),

$$s_i \leq k + j - 2J_\nu + \min(J_\nu, J_\nu + \nu + 1 - j).$$

The right side of (3) attains its maximum for $j \geq \nu + 1$, and thus,

$$s_i \leq k - J_\nu + \nu + 1.$$

2) The second case is for $j > J_\nu$. The proof for this case follows immediately by applying the previous proof for $j > J_\nu$ in reversed columns order of $\tilde{G}$, i.e., for indices $\tilde{i}$, $\tilde{j}$, instead of i, j, where $\tilde{i} \triangleq n - i$, $\tilde{j} \triangleq J_\nu + \nu - j + 1$, $1 \leq \tilde{j} \leq J_\nu$. Hence, we have proved inequality (4) also for any i that corresponds to j such that $\nu + 1 \leq j \leq J_\nu + \nu$.

Since $\nu \leq J_\nu - 1$, we have evidently covered all the columns of C by now, and (4) is proved for $i = 1, 2, \cdots, n$. Applying the foregoing for the dual code concludes the proof. □

Note that the ZC codewords clearly span a subcode with contraction index $\nu = 0$, $J = J_0$. In this case, the upper bounds of Theorems 1 and 2 are indeed identical. Nevertheless, the ZC codewords are relatively simple to locate [5], and the constructive derivation of Theorem 1 is justified.

Example 5: The extended BCH $(18, 9, 6)$ code contains a subcode with contraction index $J_1 = 4$ [6]. Then from Theorem 2, $s(\text{BCH}(18, 9, 6)) \leq 7$. The Wolf bound (2) yields 9.

Example 6: The binary Golay $(24, 12, 8)$ code includes a subcode with $J_1 = 5$ [7]. From Theorem 2, $s(\text{Golay}) \leq 9$. This is in fact the exact absolute minimal trellis size [3].

Example 7: Consider the BCH code $C(35, 28, 4)$ with roots $\{5, 7\}$. As in Example 4, we explore first the cyclic code $C_1(35, 7)$, generated by $g(x) = 1 + x^3 + x^4 + x^7 + x^8 + x^9 + x^{10} + x^{11} + x^{13} + x^{15} + x^{16} + x^{17} + x^{19} + x^{20} + x^{22} + x^{26} + x^{27} + x^{28}$, which is equivalent to the dual code of C. A subcode of C_1 is generated by (found by a computer search)

$$\begin{pmatrix} 00011000110001100011000110001100011 \\ 00110001100011000110001100011000110 \\ 01100011000110001100011000110001100 \\ 11000110001100011000110001100011000 \end{pmatrix}.$$

Consequently $J_1 = 4$ and from Theorem 2 $s(35, 28, 4) \leq 7 - 4 + 1 + 1 = 5$. The previous bound (2) is 7. A proper permutation for attaining the bound is defined by interleaving the coordinates, namely the coordinated are joined into the sets $\{1, 6, 11, 16, 21, 26, 31\}$, $\{2, 7, 12, 17, 22, 27, 32\}, \cdots, \{5, 10, 15, 20, 25, 30, 35\}$. It can be easily verified that the minimal distance of C_1 is $d_1 = 12$. The general upper bound $J \leq \lfloor n/d \rfloor$ [4] implies for C_1 $J \leq \lfloor 35/12 \rfloor = 2$ and similarly for C $J \leq \lfloor 35/4 \rfloor = 8$. Hence, in this example, any ZC set in either the code or its dual code implies an inferior bound with respect to the foregoing result.

III. Bounds Based on Code Constructions

In the foregoing section, the appropriate permutations were obtained by utilizing general properties of codes. In this section, we address codes whose structure is based on prescribed constructions and present bounds on s and $s(C)$. Some basic structures of G are first referred in behalf of the main results that follow.

Lemma 1: Let $C(n, k)$ be a code with generator matrix

$$G = \begin{pmatrix} G_1 \\ \vdots \\ G_L \end{pmatrix}.$$

Let $s(i)$ be the minimal trellis size index of subcode i generated by $G_i, i = 1, 2, \cdots, L$. Then,

$$s(C) \leq s \leq \sum_{i=1}^{L} s(i).$$

Proof: Denote by $p_j(i), f_j(i), s_j(i)$ the trellis parameters of subcode i in level j, $j = 0, 1, \cdots, n$. Clearly $p_j \geq \sum_i p_j(i)$, $f_j \geq \sum_i f_j(i)$, and by (1) $s_j \leq \sum_i (k(i) - p_j(i) - f_j(i))$. Thus,

$$s_j \leq \sum_{i=1}^{L} s_j(i).$$

By definition, $s(C) \leq s = \max_j(s_j) \leq \sum_i \max_j(s_j(i)) = \sum_i s(i)$. □

Consider a code C which is a *direct sum* of subcodes $C_i, i = 1, 2, \cdots, L]$. Using the same notations previously stated, clearly $s_j = s_t(i)$, where column j of G is the tth column of G_i adjusted with zeros. This implies the following Lemma.

Lemma 2: Let $C(n, k)$ be a code with generator matrix

$$G = \begin{pmatrix} G_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & G_L \end{pmatrix},$$

(G is the *direct sum* of $G_1, G_2, \cdots, G_L$). $s(i), s^*(i)$ are the minimal trellis size and absolute minimal trellis size indices of G_i. Then $s = \max_i(s(i))$ and $s(C) = \max_i(s^*(i))$.

Lengthening a code by joining additional columns to its generator matrix can never decrease the dimensions of its state spaces s_i. The next Lemma is immediately implied.

Lemma 3: Let $C(n, k)$ be a code with generator matrix $G = (G_1 G_2 \cdots G_L)$. Then, $\max_i(s(i)) \leq s$ and $\max_i(s^*(i)) \leq s(C)$.

Consider a code which consists of two subcodes as described in Lemma 1, and G_1 is a direct sum of L subcodes as described in Lemma 2. Suppose that the coordinates order of the subcodes of G_1 is such that $s(i) = s^*(i)$. An obvious result is the following Lemma.

Lemma 4: Let $C(n, k)$ be a code with generator matrix

$$G = \begin{pmatrix} G_1 & & \mathbf{0} \\ & \ddots & \\ \underbrace{\mathbf{0}}_{G_{L+1}} & & G_L \end{pmatrix},$$

where $k(i)$, $s^*(i)$ are the dimension and absolute minimal trellis size index of the subcode generated by G_i, $i = 1, 2, \cdots, L$. Then, $s(C) \leq k - \sum_{i=1}^{L} k(i) + \max_{i=1}^{L}(s^*(i))$.

Such structure exists in the Reed–Solomon codes [11] and was employed for bit-level soft decoding of these codes.

A. The Squaring Construction Bound

Theorem 3: Let $C(n,k)$ be a code generated by the squaring construction [2], also defined as $c=(u|u+v)$, $c\in C$, $u\in U$, $v\in T$, where $T\subseteq U$. $k(T),k(U)$ denote the dimension of T, U, respectively. Then,

$$s(C)\le \min\left(k-2k(T)+s(T),2k(U)-k+s(U)\right).$$

Proof: In compliance with the construction definition, the generator matrix of C may be exhibited in the form

$$G=\begin{pmatrix} G_T & 0\\ G_U & G_U\end{pmatrix},$$

G_U and G_T generate the subcodes U and T, respectively. $T\subseteq U$, therefore define $G_{U/T}$ such that $G_U=\binom{G_{U/T}}{G_T}$. By row operations, G may be changed to the form

$$\begin{pmatrix} G_T & 0\\ 0 & G_T\\ G_{U/T} & G_{U/T}\end{pmatrix}.$$

Applying Lemma 4 with this structure yields $s(C)\le k-2k(T)+s(T)$.

The dual code $C^{\perp}$ is $(u|u+v)$, $u\in T^{\perp}$, $v\in U^{\perp}$ [2]. Thus,

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

Also $U^{\perp}\subseteq T^{\perp}$ and similarly by row operations H may be changed to

$$\begin{pmatrix} H_U & 0\\ 0 & H_U\\ H_{T/U} & H_{T/U}\end{pmatrix},$$

where $H_{T/U}\triangleq G_{T^{\perp}/U^{\perp}}$, and $s(C)\le n-k-2(\frac{n}{2}-k(U))+s(U)=2k(U)-k+s(U)$. □

B. Bounds for the Reed–Muller Codes

Any $\mathrm{RM}(r,m)$ code can be generated by the squaring construction [2] $\mathrm{RM}(r,m)=|\mathrm{RM}(r,m-1)/\mathrm{RM}(r-1,m-1)|^2$. With the notations of Theorem 3, $U\equiv\mathrm{RM}(r,m-1)$, $T\equiv\mathrm{RM}(r-1,m-1)$. Denote the dimensions and absolute minimal trellis size parameters of $\mathrm{RM}(r,m)$ by $k(r,m)$, $s(r,m)$, respectively. Recall [10, p.376] that $k(r,m)=\sum_{i=0}^{r}\binom{m}{i}$, then from Theorem 3,

$$\begin{aligned} s(r,m)&\le s(r-1,m-1)k(r,m)-2k(r-1,m-1)\\ &=s(r-1,m-1)+k(r,m-1)-k(r-1,m-1)\\ &=s(r-1,m-1)+\sum_{i=0}^{r}\binom{m-1}{i}-\sum_{i=0}^{r-1}\binom{m-1}{i}\\ &=s(r-1,m-1)+\binom{m-1}{r}. \end{aligned}$$

Inequality (8) is a recursive expression, so that

$$\begin{aligned} s(r,m)&\le s(r-2,m-2)+\binom{m-1}{r}+\binom{m-2}{r-1}\\ &\le s(r-3,m-3)+\binom{m-1}{r}+\binom{m-2}{r-1}+\binom{m-3}{r-2}\\ &\ \ \vdots\\ &\le s(0,m-r)+\sum_{i=1}^{r}\binom{m-i}{r-i+1}\\ &=1+\sum_{i=1}^{r}\binom{m-1-r+i}{i}\\ &=\sum_{i=0}^{r}\binom{m-1-r+i}{i}. \end{aligned}$$

Using the identity [12, p. 159] $\sum_{k=0}^{b}\binom{a+k}{k}=\binom{a+b+1}{b}$, it follows that $s(r,m)\le\binom{m}{r}$. A similar procedure for the dual code, which is $\mathrm{RM}(m-r-1,m)$ ([10, p. 376]), yields

$$s(r,m)\le\min\left[\binom{m}{r},\binom{m}{m-r-1}\right].$$

However, the unique recursive SC design for the RM codes may be further exploited. In fact, the exact minimal trellis size for this design is obtained, hereby improving (9).

Theorem 4: For any $\mathrm{RM}(r,m)$ code, define $\tilde{r}\triangleq\min(r,m-r-1)$. Then, $s(r,m)\le\sum_{i=0}^{\tilde{r}}\binom{m-2i-1}{r-i}$, and the bound is the minimal trellis size for the SC design.

Proof: Consider the SC design for $\mathrm{RM}(r,m)$. We seek the maximal state space dimension s_i of the code, denoted as $s_{\mathrm{sc}}(r,m)$, from all the trellis levels V_i, $i=0,1,\cdots,n$, where $n=2^m$. From the structure of the generator matrix (6), clearly the left and right halves of the trellis, corresponding to positions $0,1,\cdots,n/2$, and $n/2,n/2+1,\cdots,n$, respectively, are symmetric in sense of the state space dimensions. Thus, we may seek the maximal s_i at either of these sections. Suppose we choose the left half. Obviously, there is a level with state space dimension $s_{\mathrm{sc}}(r,m)$ at the left or right half of this section (or in both). Again, we select that half. We may proceed recursively to divide each section into left and right subsections and select the one in which the maximal state space dimension equals $s_{\mathrm{sc}}(r,m)$. Such a recursive course is referred in the sequel as a *valid course*. In the Appendix, it is proved that an alternating left–right course with maximum $2r$ steps is a valid course.

Denote by $k_{i-}(r,m),k_{i+}(r,m)$ the dimensions of the codes generated by the columns in positions $1,2,\cdots,i$ and $i+1,i+2,\cdots,n$ of G, respectively. Denote by $s_i(r,m)$ the state space dimension at level V_i. Consider first the case where $m\ge 2r+1$ and consequently $\tilde{r}=r$. Then, from (6),

$$k_{i+}(r,m)=\begin{cases} k_{j+}(r,m-1), & 2^{m-1}\le i\le 2^m,\quad j=i-2^{m-1},\\ k_{j+}(r-1,m-1)+k(r,m-1), & \\ \quad 0 & \le i\le 2^{m-1},\quad j=i,\end{cases}$$

$$k_{i-}(r,m)=\begin{cases} k_{j-}(r-1,m-1)+k(r,m-1), & 2^{m-1}\le i\le 2^m,\quad j=i-2^{m-1},\\ k_{j-}(r,m-1), & \\ \quad 0 & \le i\le 2^{m-1},\quad j=i.\end{cases}$$

Notice that in (10) position 2^{m-1} is equivalently accounted in both subsections. By the definition of p_i and f_i, clearly

$$P_i(r,m)=k_{i-}(r,m)-s_i(r,m),$$
$$f_i(r,m)=k_{i+}(r,m)-s_i(r,m);$$

thus from (1) and (11) it follows that

$$s_i(r,m) = k_{i-}(r,m) + k_{i+}(r,m) - k(r,m).$$

Recursive substitution of (10) in (12) according to a $2r$-step alternating left–right course, starting from left, yields for proper i

$$\begin{aligned} s_i(r,m) &= k_{j_1-}(r,m-1) + k_{j_1+}(r-1,m-1) - k(r,m) \\ &\quad + k(r,m-1) \\ &= k_{j_2-}(r-1,m-2) + k_{j_2+}(r-1,m-2) - k(r,m) \\ &\quad + k(r,m-1) + k(r,m-2) \\ &\quad \vdots \\ &= k_{j_{2r}-}(0,m-2r) + k_{j_{2r}+}(0,m-2r) - k(r,m) \\ &\quad + \sum_{t=0}^{r-1} k(r-t,m-2t-1) + k(r-t,m-2t-2), \end{aligned}$$

where i and $j_1, j_2, \cdots, j_{2r}$ are the appropriate positions within the inner left or right subsections that correspond to a common coordinate of $\mathrm{RM}(r,m)$. If follows that

$$\begin{aligned} s_{\mathrm{sc}}(r,m) &= 2 - k(r,m) + \sum_{t=0}^{r-1} k(r-t,m-2t-1) \\ &\quad + (r-t,m-2t-2). \end{aligned}$$

Let $j \triangleq t-1$. Thus (13) becomes

$$\begin{aligned} s_{\mathrm{sc}}(r,m) &= 2 - k(r,m-1) - k(r-1,m-1) \\ &\quad + \sum_{j=0}^{r-2} k\,(r-j-1,m-2j-3) \\ &\quad + k(r-j-1,m-2j-4) \\ &\quad + k(r,m-1) + k(r,m-2) \\ &= s_{\mathrm{sc}}(r-1,m-2) - k(r,m-2) - k(r-1,m-1) \\ &\quad + k(r,m-1) + k(r,m-2) \\ &= s_{\mathrm{sc}}(r-1,m-2) + \binom{m-1}{r}. \end{aligned}$$

The explicit formula for $s_{\mathrm{sc}}(r,m)$ immediately follows:

$$\begin{aligned} s_{\mathrm{sc}}(r,m) &= s_{\mathrm{sc}}(r-1,m-2) + \binom{m-1}{r} = s_{\mathrm{sc}}(r-2,m-4) \\ &\quad + \binom{m-1}{r} + \binom{m-3}{r-1} \\ &= \cdots = s_{\mathrm{sc}}(0,m-2r) + \sum_{j=0}^{r-1} \binom{m-2j-1}{r-j} \\ &= \sum_{j=0}^{r} \binom{m-2j-1}{r-j}. \end{aligned}$$

Until now, we have considered the case $m \geq 2r+1$. Otherwise, $\tilde{r} = m-r-1$, but this is the order of the dual RM code for which the foregoing proof similarly holds. □

Table I contains a summary of upper bounds on $s(\mathrm{RM})$ obtained by Theorem 4, as compared to the SC bound (9) (appears in parenthesis to the right of the best bound) and to the general Wolf bound (2) (appears in parenthesis below the best bound).

TABLE I
UPPER BOUNDS ON $s(\mathrm{RM}(r,m))$

$r \backslash m$	1	2	3	4	5	6	7	8
0	$1_{(1)}{}^{(1)}$	$1_{(1)}{}^{(1)}$	$1_{(1)}{}^{(1)}$	$1_{(1)}{}^{(1)}$	$1_{(1)}{}^{(1)}$	$1_{(1)}{}^{(1)}$	$1_{(1)}{}^{(1)}$	$1_{(1)}{}^{(1)}$
1		$1_{(1)}{}^{(1)}$	$3_{(3)}{}^{(3)}$	$4_{(5)}{}^{(4)}$	$5_{(6)}{}^{(5)}$	$6_{(7)}{}^{(6)}$	$7_{(8)}{}^{(7)}$	$8_{(9)}{}^{(8)}$
2			$1_{(1)}{}^{(1)}$	$4_{(5)}{}^{(4)}$	$9_{(16)}{}^{(10)}$	$14_{(22)}{}^{(15)}$	$20_{(29)}{}^{(21)}$	$27_{(37)}{}^{(28)}$
3				$1_{(1)}{}^{(1)}$	$5_{(6)}{}^{(5)}$	$14_{(22)}{}^{(15)}$	$29_{(64)}{}^{(35)}$	$49_{(93)}{}^{(56)}$
4					$1_{(1)}{}^{(1)}$	$6_{(7)}{}^{(6)}$	$20_{(29)}{}^{(21)}$	$49_{(93)}{}^{(56)}$
5						$1_{(1)}{}^{(1)}$	$7_{(8)}{}^{(7)}$	$27_{(37)}{}^{(28)}$
6							$1_{(1)}{}^{(1)}$	$8_{(9)}{}^{(8)}$
7								$1_{(1)}{}^{(1)}$

The small numbers at the right and bottom correspond to (9) and (2), respectively.

Muder [3] derives the following lower bound on $s(C)$ for any code $C(n,k,d)$:

$$s(C) \geq \max\Big(\min(k, n-k-2\Delta), \min(n-k, k-2\Delta^{\perp}), 0\Big),$$

where $\Delta \triangleq n-k-d+1$, $\Delta^{\perp} \triangleq k-d^{\perp}+1$. By substitution of the explicit expression of n, k and d in (15) and by Theorem 4, it follows that

$$m-1 \leq s(\mathrm{RM}(1,m)) = s(\mathrm{RM}(m-2,m)) \leq m.$$

C. Bounds for Punctured and Shortened Codes

Lemma 5: Let the absolute minimal trellis size index $s(C)$ of a code C be bounded such that $x \leq s(C) \leq y$, then for the *punctured* code $\overline{C}$ (obtained by deleting a coordinate) $x-1 \leq s\left(\overline{C}\right) \leq y$.

Proof: Let $G, \overline{G}$ be the generator matrices of $C, \overline{C}$ respectively. Suppose that the last coordinate of C is deleted. By elementary row operations, G may be exhibited as

$$G = \begin{pmatrix} & & w \\ & & 0 \\ \overline{G} & & 0 \\ & & \vdots \\ & & 0 \end{pmatrix},$$

where w represents an element of $\mathrm{GF}(q)$. For *any* permutation of C, $\overline{C}$ may have state spaces with dimensions $\overline{s}_i$ such that $s_i - 1 \leq \overline{s}_i \leq s_i$. Thus by proper permutations on C and $\overline{C}$, the bounds $s\left(\overline{C}\right) \leq s(C)$, $s(C)-1 \leq s\left(\overline{C}\right)$ are respectively obtained. □

By Lemma 5, the lower and upper bound for $s(C)$ derived for the RM codes imply such bounds for the BCH codes which are the punctured RM codes. In particular, applying Lemma 5 on (16) yields for binary Hamming $(2^m-1, 2^m-m-1, 3)$ codes $(m \geq 2)$,

$$m-2 \leq s(\mathrm{Hamming}) \leq m.$$

The lower bound in (17) improves the general lower bound for perfect codes of [3], in the case of binary Hamming codes. The upper bound in (17) meets the original bound (2).

Good codes may be obtained by *shortening* other codes, namely, taking a cross-section of the code [10, p. 29]. Denote by $\overline{C}$ the shortened code obtained from a code C. Obviously from Lemma 3, $s\left(\overline{C}\right) \leq s(C)$. Such a simple conclusion may be quite powerful, as demonstrated in the following example.

Example 8: The Y1 construction [10, p. 592] is a shortening method in which an $(n-d^{\perp}, k-d^{\perp}+1, \geq d)$ code is obtained from $C(n,k,d)$, $d^{\perp}$ is the Hamming distance of the dual code of C. Consider the binary code $(105, 43, \geq 21)$ obtained from the BCH $(127, 64, 21)$ by the Y1 construction [13]. Then from Lemmas 3, 5 and Theorem 4, $s(105, 43) \leq 29$. Another example is the binary code $(207, 116, \geq 25)$ similarly obtained from the BCH $(255, 163, 25)$. By the same arguments $s(207, 116) \leq 49$. Both codes are currently considered as best codes [10, p. 675]. Clearly these results significantly improve those obtained by (2).

D. Bounds for Repeated-Root Cyclic Codes

Theorem 3 may be effectively applied for some classes of repeated-root cyclic codes [8], [9]. Consider the binary repeated-root cyclic code $C_{r1}(2^{m+1}-2, 2^{m+1}-m-4, 4)$, $m \geq 2$, with generator polynomial $g(x) = (x+1)^2 m_1(x)$, where $m_1(x)$ is a primitive polynomial of elements in GF(2^m). Due to Van Lint [8], $C_{r1} = |U/T|^2$, where U, T are the BCH codes with generators $g_U(x) = x+1$, $g_T(x) = (x+1)m_1(x)$. T is the expurgated Hamming code [10, p. 29] with parity check matrix $H_T = \binom{H_1}{11\cdots 1}$, where H_1 is the parity check matrix of the Hamming $(2^m-1, 2^m-1-m, 3)$ code. From the structure of H_T and (17), $m-2 \leq s(T)$. From (7), followed by an elementary row operation, we get

$$H = \begin{pmatrix} 11\cdots 1 & 00\cdots 0 \\ H_1 & H_1 \\ 00\cdots 0 & 11\cdots 1 \end{pmatrix}.$$

Then, from Lemma 3 (applied on the dual code) and (18), $s(C_{r1}) \geq m-2$. Clearly, $s(T) \leq s(\text{Hamming}) + 1 \leq m+1$. Also $s(U) = 1$. Then, from Theorem 3, $s(C_{r1}) \leq m+1$. Thus,

$$m-2 \leq s(C_{r1}) \leq m+1.$$

Another class of binary repeated-root cyclic codes $C_{r2}(2^{m+2}-4, 2^{m+2}-m-7, 4)$, $m \geq 2$, is obtained [8] by $\left||U/T|^2/|T/V|^2\right|^2$, where U, T, V are cyclic codes generated by the polynomials $g_U(s) = 1$, $g_T(x) = x+1$, $g_V(x) = (x+1)m_1(x)$, respectively. $m_1(x)$ is a primitive polynomial of elements in GF(2^m). Clearly, $s(|U/T|^2) = 1$ and $|T/V|^2 \equiv C_{r1}$. From (7), (19), and Theorem 3 it can be similarly shown that

$$m-2 \leq s(C_{r2}) \leq m+2.$$

Both results improve the previously known bounds for these codes obtained by (2).

IV. Conclusion

Significant progress in the theory of trellis diagrams for block codes was achieved recently following the original work of Wolf [1]. It includes the study of minimal trellis representation by Forney [2], which was then formalized an generalized in [3]. It was recognized [2], [3] that the coordinate ordering of the code plays a major role in the trellis size. Yet, no general algorithm for finding the optimal permutation is currently known. The first progress in this line is the use of cubic construction and squaring construction for Golay Code $(24, 12)$ and RM codes, respectively [2]. These constructions were also utilized for efficient section by section trellis-based decoding.

In this study, the possibilities in shrinking the trellis of various codes are further explored. First, two improved general upper bounds on $s(C)$ along with the corresponding permutations are given for any linear block code. Certain code constructions are also investigated. In particular, a general bound for SC design is derived and an explicit formula of the trellis size for the recursive SC of RM codes is obtained. This formula, as compared (in Table I) to the general SC bound and the Wolf bound, suggests (not surprisingly) that intensive use of code properties has a greater potential for minimizing the trellis. Known constructive techniques such as shortening and puncturing are used to improve the current bounds on $s(C)$ for various codes, e.g., a lower bound for binary Hamming codes. Still, no essential structural attributes of the codes are involved with the lower bounds.

The results in this work, concerning long codes as well as short codes, supports the use of the trellis size as a decoding complexity measure. The related permutations contribute to practical design of trellis-based decoders.

Appendix

We prove by induction that an alternating left–right recursive course in the trellis, started from left, with maximum $2r$ steps is a *valid course*. First, consider a course through u subsections, t of which are "left type" subsections. It can be readily shown by recursive substitution of (10), starting with (12), that

$$s_i(r,m) = k_{\tilde{i}-}(r-(u-t), m-u) + k_{\tilde{i}+}(r-t, m-u) + A,$$

where $\tilde{i}$ is the corresponding column of i within the final subsection in the selected course, and A is a certain sum of RM(r,m) subcodes dimensions. The basis of the induction is the first 1-step course in which the left subsection of the code is selected. As previously mentioned, this is obviously a valid course due to the symmetric structure of G. Assume now that a u-step alternating course, which is a valid course due to the induction hypothesis, has been chosen, started from left. The next subsection is now selected. Distinguish between two cases.

1) u is odd, i.e., the course consists of $t = (u+1)/2$ left subsections and $t-1$ right subsections, $u < 2r$. Selecting the next subsection at the left yields for appropriate i_1:

$$s'_{i_1}(r,m) = k_{\tilde{i}-}(r-t+1, m-2t) + k_{\tilde{i}+}(r-t-1, m-2t) + k(r-t, m-2t) + A,$$

while selecting the right subsection yields for the corresponding positions:

$$s''_{i_2}(r,m) = k_{\tilde{i}-}(r-t, m-2t) + k_{\tilde{i}+}(r-t, m-2t) + k(r-t+1, m-2t) + A.$$

We must prove that the maximal $s''_{i_2}(r,m)$ is not less than the maximal $s'_{i_1}(r,m)$. If $t = r$ and $m = 2r+1$ then $\max\{s'_{i_1}(r,m)\} = 2+0+1+A = 3+A$ and $\max\{s''_{i_2}(r,m)\} = 1+1+2+A = 4+A$. Else, let us examine first the right half of both subsections, i.e., for $2^{m-2t-1} \leq \tilde{i} \leq 2^{m-2t}$. From (10), (A.2) becomes

$$s'_{i_1}(r,m) = k_{\tilde{j}-}(r-t, m-2t-1) + k_{\tilde{j}+}(r-t-1, m-2t-1) + k(r-t, m-2t) + k(r-t+1, m-2t-1) + A.$$

$$s''_{i_2}(r,m) = k_{\tilde{j}-}(r-t-1, m-2t-1) + k_{\tilde{j}+}(r-t, m-2t-1) + k(r-t+1, m-2t) + k(r-t, m-2t-1) + A.$$

The maximal sum of the first two elements in (A.3a), (A.3b) is identical due to symmetry. Subtracting the remaining (constant) part of (A.3a) from the corresponding part of (A.3b) yields $k(r-t+1, m-2t) - k(r-t, m-2t) + k(r-t, m-2t-1) - k(r-t+1, m-2t-1) = \binom{m-2t}{r-t+1} - \binom{m-2t-1}{r-t+1} > 0.$

Consequently, at the right half evidently $\max\{s''_{i_2}(r,m)\} > \max\{s'_{i_1}(r,m)\}$. Consider now the left half of (A.2a) and (A.2b). Similarly, (A.2) becomes

$$s'_{i_1}(r,m) = k_{\tilde{j}_-}(r-t+1, m-2t-1) + k_{\tilde{j}_+}(r-t-2, m-2t-1) + k(r-t, m-2t) + k(r-t-1, m-2t-1) + A.$$

$$s''_{i_2}(r,m) = k_{\tilde{j}_-}(r-t, m-2t-1) + k_{\tilde{j}_+}(r-t-1, m-2t-1) + k(r-t+1, m-2t) + k(r-t, m-2t-1) + A.$$

Recall that $\mathrm{RM}(r-t-2, m-2t-1) \subset \mathrm{RM}(r-t-1, m-2t-1)$. Therefore, $k_{\tilde{j}_+}(r-t-1, m-2t-1) \geq k_{\tilde{j}+}(r-t-2, m-2t-1)$, and also $k(r-t, m-2t-1) > k(r-t-1, m-2t-1)$. Subtracting the remaining part of (A.4a) from the remaining of (A.4b) yields $\delta \triangleq k_{\tilde{j}_-}(r-t, m-2t-1) + k(r-t+1, m-2t) - k_{j-}(r-t+1, m-2t-1) - k(r-t, m-2t)$. Clearly, $k_{\tilde{j}_-}(r-t, m-2t-1) = k(r-t, m-2t-1) - f_{\tilde{j}}(r-t, m-2t-1)$ and similarly $k_{\tilde{j}_-}(r-t+1, m-2t-1) = k(r-t+1, m-2t-1) - f_{\tilde{j}}(r-t+1, m-2t-1)$. Also, $f_{\tilde{j}}(r-t+1, m-2t-1) \geq f_{\tilde{j}}(r-t, m-2t-1)$ and consequently, $\delta \geq k(r-t, m-2t-1) - k(r-t+1, m-2t-1) + k(r-t+1, m-2t) - k(r-t, m-2t) = \binom{m-2t}{r-t+1} - \binom{m-2t-1}{r-t+1} > 0$.

2) The second case is when the course consists of even steps, i.e., $t = u/2$ left subsections and also t right subsections. Selecting the next subsection at the left yield

$$s'_{i_1}(r,m) = k_{\tilde{i}_-}(r-t-1, m-2t-1) + k_{\tilde{i}_+}(r-t, m-2t-1) + k(r-t, m-2t-1) + B,$$

while selecting the right subsection yields

$$s''_{i_2}(r,m) = k_{\tilde{i}_-}(r-t, m-2t-1) + k_{\tilde{i}_+}(r-t-1, m-2t-1) + k(r-t, m-2t-1) + B.$$

The symmetry between $k_{\tilde{i}_+}$ and $k_{\tilde{i}_-}$ implies that the maximal values of $s'_{i_1}(r,m)$ and $s''_{i_2}(r,m)$ in (A.5) are identical, and the induction is completed.

ACKNOWLEDGMENT

The authors wish to thank A. Vardy for helpful discussions and for providing software for finding zero-concurring codewords and contractible subcodes. They also thank the referees for simplifying the final form of (9) and for many other suggestions which improved the presentation of this correspondence.

REFERENCES

[1] J.K. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 76–80, 1978.

[2] G.D. Forney, Jr., "Coset codes—Part II: Binary lattices and related codes," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1152–1187, Sept. 1988.

[3] D.J. Muder, "Minimal trellises for block codes," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1049–1053, Sept. 1988.

[4] Y. Be'ery and J. Snyders, "Optimal soft decision block decoders based on fast Hadamard transform," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 355–364, May 1986.

[5] A. Vardy and Y. Be'ery, "On the problem of finding zero-concurring codewords," *IEEE Trans. Inform. Theory*, vol. 37, pp. 180–187, Jan. 1991.

[6] J. Snyders and Y. Be'ery, "Maximum likelihood soft decoding of binary block codes and decoders for the Golay codes," *IEEE Trans. Inform. Theory*, vol. 35, pp. 963–975, Sept. 1989.

[7] A. Vardy, J. Snyders, and Y. Be'ery, "Bounds on the dimension of codes and subcodes with prescribed contraction index," *Linear Algebra Appl.*, vol. 142, pp. 237–261, 1990.

[8] J.H. van Lint, "Repeated-root cyclic codes," *IEEE Trans. Inform. Theory*, vol. 37, pp. 343–345, Mar. 1991.

[9] G. Castagnoli, J.L. Massey, Ph. Schoeller, and N. von Seemann, "On repeated-root cyclic codes," *IEEE Trans. Inform. Theory*, vol. 37, pp. 337–342, Mar. 1991.

[10] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North Holland, 1977.

[11] A. Vardy and Y. Be'ery, "Bit-level soft-decision decoding of Reed–Solomon codes," *IEEE Trans. Commun.*, vol. 39, pp. 440–445, 1991.

[12] R.L. Graham, D.E. Knuth, and O. Patashnik, *Concrete Mathematics*. Reading, MA: Addison-Wesley, 1989.

[13] H.J. Helgert and R.D. Stinaff, "Shortened BCH codes," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 818–820, 1973.

Constructions for Perfect Mixed Codes and Other Covering Codes

Tuvi Etzion, *Member IEEE,* and Gadi Greenberg

***Abstract*—A construction for an infinite family of perfect mixed codes with covering radius 2 is presented. These are the first known nontrivial perfect mixed codes with covering radius greater than 1. Based on mixed codes constructions for binary covering codes that lead to a considerable improvement of upper bounds on the sizes of covering codes are presented. These codes and some other codes can be obtained by the blockwise direct sum construction. Two infinite families of codes are of a special interest, they are quasi-perfect, nonlinear, union of their disjoint translates covers the space, and their density as covering codes is remarkably low.**

***Index Terms*—Combining codes, covering codes, Hamming code, mixed code, normal code, perfect code, Preparata code, quasi-perfect code.**

I. INTRODUCTION

In this correspondence, we study two important problems on covering radius of codes. The first one is the existence of perfect mixed codes with covering radius greater than 1. The second is the upper bounds on $K(n,R)$, the minimum cardinality of any binary code of length n and covering the radius R.

Let $r > 1$ be an integer, $k_1, k_2, \cdots, k_r$ be distinct integers greater than 1, and $n_1, n_2, \cdots, n_r$ integers greater than 0. Let Z_{k_i} be the group of integers modulo k_i, and $Z_{k_i}^{n_i}$ is the set of all words of

Manuscript received November 15, 1991. This work was supported in part by the Technion V.P.R. Fund.

T. Etzion is with the Computer Science Department, Technion—Israel Institute of Technology, Haifa 32000, Israel.

G. Greenberg is with the Mathematics Department, Technion—Israel Institute of Technology, Halfa 32000, Israel.

IEEE Log Number 9203015.