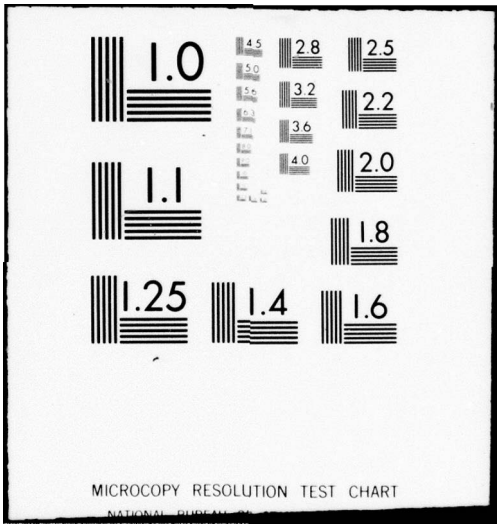
AD-A069 778 ILLINOIS UNIV AT URBANA-CHAMPAIGN COORDINATED SCIENCE LAB F/G 9/4
NEW CODES BEYOND THE ZYABLOV BOUND AND THE GOPPA-BASED JUSTESEN--ETC(U)
NOV 78 R J KLEINHENZ DAAB07-72-C-0259
UNCLASSIFIED R-833 NL

| OF |
AD
A069778

END
DATE
FILMED
7-79
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART

REPORT R-833 NOVEMBER, 1978 UILU-ENG 78-2226

CSL COORDINATED SCIENCE LABORATORY

LEVEL II

NEW CODES BEYOND THE ZYABLOV BOUND AND THE GOPPA-BASED JUSTESEN CODES

ROBERT JOSEPH KLEINHENZ

UNIVERSITY OF ILLINOIS - URBANA, ILLINOIS

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE *(and Subtitle)*

NEW CODES BEYOND THE ZYABLOV BOUND AND THE GOPPA-BASED JUSTESEN CODES,

5. TYPE OF REPORT & PERIOD COVERED

Technical Report

6. PERFORMING ORG. REPORT NUMBER

R-833, UILU-ENG-78-2226

7. AUTHOR(s)

Robert Joseph Kleinhenz

8. CONTRACT OR GRANT NUMBER(s)

DAAB-07-72-C-0259

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Joint Services Electronics Program

12. REPORT DATE

November 1978

13. NUMBER OF PAGES

61

14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*

Doctoral thesis,

15. SECURITY CLASS. *(of this report)*

UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Coding Theory
Asymptotically Good Codes
Concatenated Codes
Zyablov Bound

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

In this report, constructions are given for two new block codes. The first construction produces a class of asymptotically good codes that lie above both the Zyablov and SKHN bounds for certain rates. The counting techniques of Weldon are discussed and generalized and are used to compute a lower-bound on the distance-to-length ratio of the new codes. The codes themselves are constructed by concatenating an SKHN code (due to Sugiyama, et al.) with an interleaved code generated by a fixed $[n_0, k_0, d_0]$ base code

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

097 700

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

20. ABSTRACT (Continued)

having weight enumerator $w_0(x)$. The result of this concatenation is a collection of codes $\mathcal{K}^{(\ell)}$, $\ell = 1,2,3,\ldots$, that has rate $r^{(\ell)}$ and distance-to-length ratio $\Delta^{(\ell)}$ satisfying

$$\Delta^{(\ell)} \geq \frac{1}{n_0}\left(\sum_{\substack{j=1 \\ J=2^j}}^{\ell} \frac{1}{2^j \delta(J-1)}\right)^{-1}\left(1 - 2^{-\ell} - \frac{r^{(\ell)}}{r_0}\right),$$

where $r_0 = k_0/n_0$ and $\delta^{(r-1)}$ (together with $z^{(r-1)}$) is the unique solution of

$$2^{k_0/r} = z^{-\delta} w_0(z)$$

$$\delta = \frac{z w_0'(z)}{w_0(z)} .$$

The codes $\mathcal{K}^{(\ell)}$ generalize the codes of Justesen, Weldon, and Sugiyama, et al.

The second construction produces a class of codes J_G which lies on the Justesen bound. These codes arise from the concatenation of a maximum distance separable code with a set of Goppa Codes. If a sufficiently large number of Goppa Codes are used as inner codes, we produce a class J_G of codes that lie on the Justesen bound for all rates. By employing the methods of Justesen or Sarwate, the codes J_G can be made to correct the same number of errors in virtually the same time as a standard Justesen Code. We show, however, that by employing a fast algorithm to decode the inner Goppa Codes, we can substantially decrease the overall time required for decoding the codes J_G. Although we can improve the decoding time, we must sacrifice the asymptotically good nature of our codes to do so. The resulting quickly decoded version of J_G, although asymptotically bad, still corrects many more errors than a comparable BCH code and requires practically the same amount of time to decode.

UILU-ENG 78-2226

NEW CODES BEYOND THE ZYABLOV BOUND AND THE GOPPA-BASED JUSTESEN CODES

by

Robert Joseph Kleinhenz

This work was supported in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy and U.S. Air Force) under Contract DAAB-07-72-C-0259.

NEW CODES BEYOND THE ZYABLOV BOUND AND THE GOPPA-BASED JUSTESEN CODES

BY

ROBERT JOSEPH KLEINHENZ

B.S., University of Santa Clara, 1971
A.M., University of Illinois, 1973

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mathematics
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1978

Thesis Adviser: Professor F. P. Preparata

Urbana, Illinois

NEW CODES BEYOND THE ZYABLOV BOUND AND THE GOPPA-BASED JUSTESEN CODES

Robert Joseph Kleinhenz, Ph.D.
Coordinated Science Laboratory and
Department of Mathematics
University of Illinois at Urbana-Champaign, 1978

In this thesis, constructions are given for two new block codes. The first construction produces a class of asymptotically good codes that lies above both the Zyablov and SKHN bounds for certain rates. The counting techniques of Weldon are discussed and generalized and are used to compute a lower-bound on the distance-to-length ratio of the new codes. The codes themselves are constructed by concatenating an SKHN code (due to Sugiyama, et al.) with an interleaved code generated by a fixed $[n_0,k_0,d_0]$ base code having weight enumerator $w_0(x)$. The result of this concatenation is a collection of codes $\mathcal{X}^{(\ell)}$, $\ell = 1,2,3,\ldots$, that has rate $r^{(\ell)}$ and distance-to-length ratio $\Delta^{(\ell)}$ satisfying

$$\Delta^{(\ell)} \geq \frac{1}{n_0}\left(\sum_{\substack{j=1 \\ J=2^j}}^{\ell} \frac{1}{2^j \delta^{(J-1)}}\right)^{-1}\left(1 - 2^{-\ell} - \frac{r^{(\ell)}}{r_0}\right),$$

where $r_0 = k_0/n_0$ and $\delta^{(r-1)}$ (together with $z^{(r-1)}$) is the unique solution of

$$2^{k_0/r} = z^{-\delta} w_0(z)$$

$$\delta = \frac{z w_0'(z)}{w_0(z)} .$$

The codes $\mathcal{X}^{(\ell)}$ generalize the codes of Justesen, Weldon, and Sugiyama, et al.

The second construction produces a class of codes J_G which lies on the Justesen bound. These codes arise from the concatenation of a maximum distance separable code with a set of Goppa Codes. If a

sufficiently large number of Goppa Codes are used as inner codes, we produce a class J_G of codes that lies on the Justesen bound for all rates. By employing the methods of Justesen or Sarwate, the codes J_G can be made to correct the same number of errors in virtually the same time as a standard Justesen Code. We show, however, that by employing a fast algorithm to decode the inner Goppa Codes, we can substantially decrease the overall time required for decoding the codes J_G. Although we can improve the decoding time, we must sacrifice the asymptotically good nature of our codes to do so. The resulting quickly decoded version of J_G, although asymptotically bad, still corrects many more errors than a comparable BCH code and requires practically the same amount of time to decode.

ACKNOWLEDGEMENT

In preparing a thesis many contributions are made by people surrounding the author. Their help is invaluable during this time and so a note of gratitude is appropriate here. My thanks go to the University of Illinois Coordinated Science Laboratory for the assistance they have given me in the preparation of this thesis. A special note of thanks goes to Professor Dilip V. Sarwate for his illuminating comments on this work. But most of all, I will always be indebted to my advisor, Professor Franco P. Preparata. His encouragement and understanding kept me going, even in the darkest of hours. Finally, I must thank my wife, Carolyn, for being so tolerant of me during the final stages of preparation. Without her moral support, this work might never have been completed. To all of these people I can only say thank you.

TABLE OF CONTENTS

CHAPTER 1

PRELIMINARIES

In any algebraic block code, the three parameters which serve to define the efficiency of the code are the length n, the information content k, and the minimum distance d. A current problem in the theory of algebraic codes is to produce a class of codes having the property that the ratio of minimum distance to length is bounded away from zero as the length tends to infinity and the rate k/n is held constant. Such a class of codes is referred to as being asymptotically good. We also wish the class of codes to be created constructively; that is, the amount of searching required to produce the class of asymptotically good codes is on the order of n, or even log n. Let us mention some of the results that have been achieved to date.

In the 1950's, Gilbert [1] and Varsharmov [2] established that there exist codes whose distance-to-length ratio Δ and rate r obey the following inequality asymptotically:

$$\Delta \geq H^{-1}(1 - r) .$$

This inequality is known as the Gilbert bound. To produce a code at the Gilbert bound, one begins with the entire codespace $A_0 = \{0,1\}^n$. A vector $\underline{x}_0$ is chosen at random from A_0, and all elements of A_0 (including $\underline{x}_0$) whose Hamming distance from $\underline{x}_0$ is less than d + 1 are removed from A_0 to form the set A_1. We then select a vector $\underline{x}_1$ from A_1, and remove all vectors from A_1 (including $\underline{x}_1$) whose Hamming distance from $\underline{x}_1$ is less than d + 1. This forms the set A_2. We continue in this fashion producing the sets $A_0, A_1, A_2, \ldots$ and the vectors $\underline{x}_0, \underline{x}_1, \underline{x}_2, \ldots$, and we stop when $A_k = \emptyset$ for some k.

The set $\{\underline{x}_0, \underline{x}_1, \ldots, \underline{x}_{k-1}\}$ is a code in the class of codes we are constructing. (Other codes arise from different choices of the parameters n and d.) We notice that in the worst case, we are excluding $\sum_{j=0}^{d} \binom{n}{j}$ vectors from the set A_i at the i-th stage of the algorithm. From this observation, it can be shown that the class of codes in question is asymptotically good. In fact, these codes are Gilbert bound codes.

The problem with this method is that ultimately we must look at every vector in A_0. While a certain amount of searching is inevitable in any code construction, it is undesirable to examine the whole codespace A_0.

As we have just seen, the construction of Gilbert and Varsharmov produces codes of length n and requires an amount of searching proportional to $|A_0| = 2^n$. In 1971, Zyablov [3] showed that by concatenating a maximum distance separable (MDS) code with a Gilbert bound code, one produces a class of codes whose rate $\mathcal{R}$ and distance-to-length ratio Δ satisfy

$$\Delta \geq \sup_{0 < r \leq 1} \{H^{-1}(1 - r)(1 - \frac{\mathcal{R}}{r})\} .$$

This bound, called the Zyablov bound, is inferior to the Gilbert bound. However, the method of Zyablov produces codes of length n which require an amount of searching proportional to n. Thus the Zyablov codes apparently trade asymptotic error tolerance for speed of construction.

Finally, in 1972 Justesen [4] produced via concatenation an asymptotically good class of codes for which the construction is compact. That is, a code of length $(2m)2^m$ is not appreciably harder to construct than a finite field of 2^m elements. Justesen's curve coincides with that of Zyablov for rates larger than .31, but for smaller rates it is a line

tangent to the Zyablov bound at .31. Subsequent compact constructions were made by Sugiyama, et al. [5, 8] and Weldon [6, 7].

In 1975 Weldon [7] made significant improvements over Justesen by creating codes whose curves lie above the Justesen bound but below the Zyablov bound for rates less than .31.

The most recent (and most dramatic) improvement to be made has been done by Sugiyama, et al. [8]. In 1978, Sugiyama, et al. produced a class of codes whose curves lies above the Zyablov bound but below the Gilbert bound for rates larger than .31. The technique used is a kind of generalized concatenation (described herein). In this paper, we extend the method of Sugiyama, et al. to produce a compactly constructed class of codes that lies above the Zyablov bound for rates larger than .205.

We now present a review of some elementary notions in algebraic coding theory together with an overview of the remainder of this thesis. At the time of this writing, all constructions that yield asymptotically good codes have relied at some point upon the method of concatenation. Since this work is no exception, we proceed with the idea of concatenation.

Concatenation is a construction whereby two codes are combined to form a new code. Specifically, the two codes, called the inner code and the outer code, are utilized as follows. The outer code is a block code whose symbols come from a symbol alphabet consisting of s_o symbols. Each symbol in each outer codeword is then encoded via the inner code. As shown in Figure 1, each concatenated codeword is a matrix whose dimensions are the lengths of the inner and outer codes, and whose i-th column is an inner codeword -- the result of encoding that symbol which appears in the i-th position of the outer codeword. We must note here that in order to accomplish the process of concatenation, it is necessary to have at

S_1^*	S_2^*	S_3^*		S_N^*
σ_{11}	σ_{12}	σ_{13}	. . .	σ_{1N}
σ_{21}	σ_{22}	σ_{23}	. . .	σ_{2N}
σ_{31}	σ_{32}	σ_{33}	. . .	σ_{3N}
.	.	.		.
.	.	.		.
.	.	.		.
σ_{n1}	σ_{n2}	σ_{n3}	. . .	σ_{nN}

Figure 1. A concatenated codeword. The vector $[S_1, S_2, \ldots, S_N]$ is a member of the outer code. The symbol S_i is encoded via the inner code to form the column $S_i^* = [\sigma_{1i}, \sigma_{2i}, \ldots, \sigma_{ni}]^T$. The matrix $[\sigma_{ij}]$ thus formed is a typical member of the concatenated code.

least as many inner codewords as there are symbols in the symbol alphabet of the outer code. Furthermore, we will always require that the symbol zero in the symbol alphabet of the outer code is encoded into the inner code's zero codeword.

As is evident from the construction, the length η, information content χ, and the minimum distance β of a concatenated code satisfy

$$\begin{aligned} \eta &= nN \\ \chi &= kK \\ \beta &\geq dD \end{aligned} \tag{1}$$

where N, K, and D are respectively the length, information content, and minimum distance of the outer code and n, k, and d are the same parameters for the inner code.

A technique that is employed frequently is the use of several inner codes, all of the same length, redundancy, and minimum distance. Justesen [4] employed this latter method in his construction. His outer code is a maximum distance separable code of length $N = 2^m-1$ over the finite field $GF(2^m)$, and the class of inner codes consists of randomly shifted codes of rate 1/2.

After the results of Justesen were announced, a method was presented by Weldon [7] in which the class of inner codes that Justesen used were replaced by a class of mutually disjoint interleaved codes generated by a fixed base code. At that time, this idea improved the best known constructive lower bound because it not only incorporates the ideas of distinctness that made the Justesen Codes asymptotically good, but also utilizes weight properties inherent in the base code. Since we will use this same modification on another code, we give here a thumbnail sketch

of the process of interleaving. (We are actually describing juxtaposition but since an interleaved and a juxtaposed code are isomorphic as codes, all algebraic properties of one are properties of the other.)

Let C_0 be a given $[n_0,k_0,d_0]$ linear code. The weight enumerator of C_0 is the polynomial $w_0(x) = \sum_{j=0}^{n_0} w_j x^j$ where w_j is the number of words in C_0 of weight j. An s-fold (or s-degree) interleaving of C_0 produces a code C_{s-1} whose generic word is $[\underline{c}_0,\underline{c}_1,\ldots,\underline{c}_{s-1}]$, where $\underline{c}_i \in C_0$. It is easy to see that the code C_{s-1} has length sn_0, information content sk_0, and minimum distance d_0. Furthermore, the weight enumerator $w_{s-1}(x)$ of C_{s-1} is related to $w_0(x)$ by $w_{s-1}(x) = [w_0(x)]^s$. The proof of this last identity is a simple combinatorial exercise and so is omitted.

In this thesis, we start with the basic construction of Sugiyama, et al. We take the codes of Sugiyama, et al. as the outer codes in a concatenation and use an s-degree interleaved code as the inner code. The length and rate of this new code can be calculated in the usual way (see Eq. (1)), however to compute the minimum distance we need to generalize a counting technique first established by Weldon [7].

To facilitate the proof of the main counting lemma, we need to rule out some types of codes from the discussion. We say that a code is <u>full</u> if the generator matrix of the code contains no column that is identically zero. Most of the standard binary codes (BCH, RM, etc.) are full. It is easy to establish that an $[n,k,d]$ code C is full by examining the weight enumerator w(x). A necessary and sufficient condition for fullness is that $w'(1) = n2^{k-1}$. This is an easy consequence of counting the total number of ones in the codewords of C. On one hand the total number of ones in all words of weight j ($0 \leq j \leq n$) is jw_j.

Thus the total number of ones in the codewords of C is w'(1). On the other hand, the fullness of the code guarantees that as we scan the codewords of C we see an equal number of zeros and ones in each position. Therefore the totality of ones in the code C is $n2^{k-1}$. Hence $w'(1) = n2^{k-1}$. If the code is not full, the second count of the total number of ones is strictly less than $n2^{k-1}$. Thus the condition is necessary and sufficient. This test for fullness is reasonable in light of our need for the weight enumerator in the calculation of our code's minimum distance.

CHAPTER 2

COUNTING TECHNIQUES

In this chapter we give a detailed and rigorous proof of the basic lemma used in establishing the minimum distance in our codes. We start with an important lemma.

Lemma of Partial Sums. Let C be any $[n,k,d]$ code with weight enumerator $w(x) = \sum_{j=0}^{n} w_j x^j$. The number of codewords in C of weight not greater than δ is itself less than $z^{-\delta}w(z)$ for all z in the interval $(0,1)$, and $\delta \leq 2^{-k}w'(1)$.

Proof. The number of codewords in C of weight not greater than δ is $\sum_{j=0}^{\delta} w_j$. The probability that the weight $wt(\underline{x})$ of a randomly selected codeword $\underline{x}$ is not greater than δ is $2^{-k} \sum_{j=0}^{\delta} w_j$.

Choose any number z in the interval $(0,1)$. The quantity $z^{wt(\underline{x})}$ is a random variable on the experiment that consists of choosing a word at random from C. The expected value $E[z^{wt(\underline{x})}]$ satisfies

$$E[z^{wt(\underline{x})}] = \sum_{\underline{x} \in C} 2^{-k} z^{wt(\underline{x})} = 2^{-k} w(z) \qquad (2)$$

since

$$w(z) = \sum_{j=0}^{n} w_j z^j = \sum_{\underline{x} \in C} z^{wt(\underline{x})} .$$

Since $0 < z < 1$ we may take $z = e^{-\lambda}$ where $\lambda > 0$. Equation (2) then becomes

$$2^{-k} w(z) = E[e^{-\lambda wt(\underline{x})}] . \qquad (3)$$

The quantity $wt(\underline{x})$ is a random variable in its own right. We take $\mu = E[wt(\underline{x})]$, and let η be any non-negative real number. By employing some simple algebra we may rewrite Eq. (3) as

$$2^{-k}w(z) = z^{\mu-\eta}E[e^{\lambda(\mu-wt(\underline{x})-\eta)}] . \tag{4}$$

The quantity $\mu - wt(\underline{x})$ is a random variable with zero mean, and so we may apply the Chernoff Bound [9] to the quantity $E[e^{\lambda(\mu-wt(\underline{x})-\eta)}]$. Equation (4) then becomes

$$2^{-k}w(z) = z^{\mu-\eta}E[e^{\lambda(\mu-wt(\underline{x})-\eta)}] \geq z^{\mu-\eta}P[\mu-wt(\underline{x}) \geq \eta] . \tag{5}$$

The probability on the right-hand side of this last inequality is $P[\mu-\eta \geq wt(\underline{x})]$, which by a previous observation is $2^{-k}\sum_{j=0}^{\mu-\eta} w_j$. Hence Eq. (5) can be transformed into $\sum_{j=0}^{\mu-\eta} w_j \leq z^{-(\mu-\eta)}w(z)$. Now set $\delta = \mu-\eta$. Notice that the restriction $\delta \leq 2^{-k}w'(1)$ given in the hypotheses allows us to do this since $\mu = E[wt(\underline{x})] = 2^{-k}w'(1)$ and η is arbitrary. □

The lemma is called the Lemma of Partial Sums since it gives an estimate of the partial sums of the sequence $w_0, w_1, \ldots, w_n$. In particular, if we take $w_j = \binom{n}{j}$, our lemma gives $2^{nH(\lambda/n)}$ as an upper-bound on the sum $\sum_{j=0}^{\lambda} \binom{n}{j}$.

We are going to use the Lemma of Partial Sums backwards. That is, we will estimate the size of $\sum_{j=0}^{\delta} w_j$ and then use the expression $z^{-\delta}w(z)$ to compute a value of δ. For this reason, it is desirable to have the inequality of the lemma as tight as possible. We can achieve this by minimizing the expression $z^{-\delta}w(z)$, for z in the interval (0,1).

Figure 2 indicates the usual shape of the graph of $z^{-\delta}w(z)$ for several values of δ. We wish to show that for a fixed constant K between 1 and w(1), there exists exactly one curve $\mathcal{C}(\delta)$ of the family $z^{-\delta}w(z)$ that is tangent to the line y = K and that the point of tangency is a local minimum of the curve $\mathcal{C}(\delta)$. We begin by establishing that for all δ between 0 and w'(1)/w(1), the function $z^{-\delta}w(z)$ has exactly one local minimum on the interval (0,1).

The derivative of $z^{-\delta}w(z)$ can be written as $z^{-\delta-1}w(z)[\frac{zw'(z)}{w(z)} - \delta]$. We see that the only place where a local extremum can occur is that value of z which solves

$$\delta = \frac{zw'(z)}{w(z)} .$$

It is not clear that the equation $\delta = \frac{zw'(z)}{w(z)}$ has any solution at all; and if the equation does possess a solution, there is no guarantee that this solution produces a local minimum of the function $z^{-\delta}w(z)$. These concerns are needless in light of the following proposition.

Proposition 1. If w(x) is a polynomial with nonnegative coefficients, deg(w) $\geq$ 1, and w(0) = 1 then the function $\delta_w(z)$ defined by

$$\delta_w(z) = \frac{zw'(z)}{w(z)}$$

is a strictly increasing function of z on (0,1).

Proof. We will prove this proposition by induction on the degree of w(x). If deg(w) = 1, the hypotheses allow us to write w(x) = 1 + ax where a > 0. A quick calculation reveals $\delta_w(z) = \frac{az}{1 + az}$, which is a strictly increasing function of z on (0,1). Next, assume the truth of the

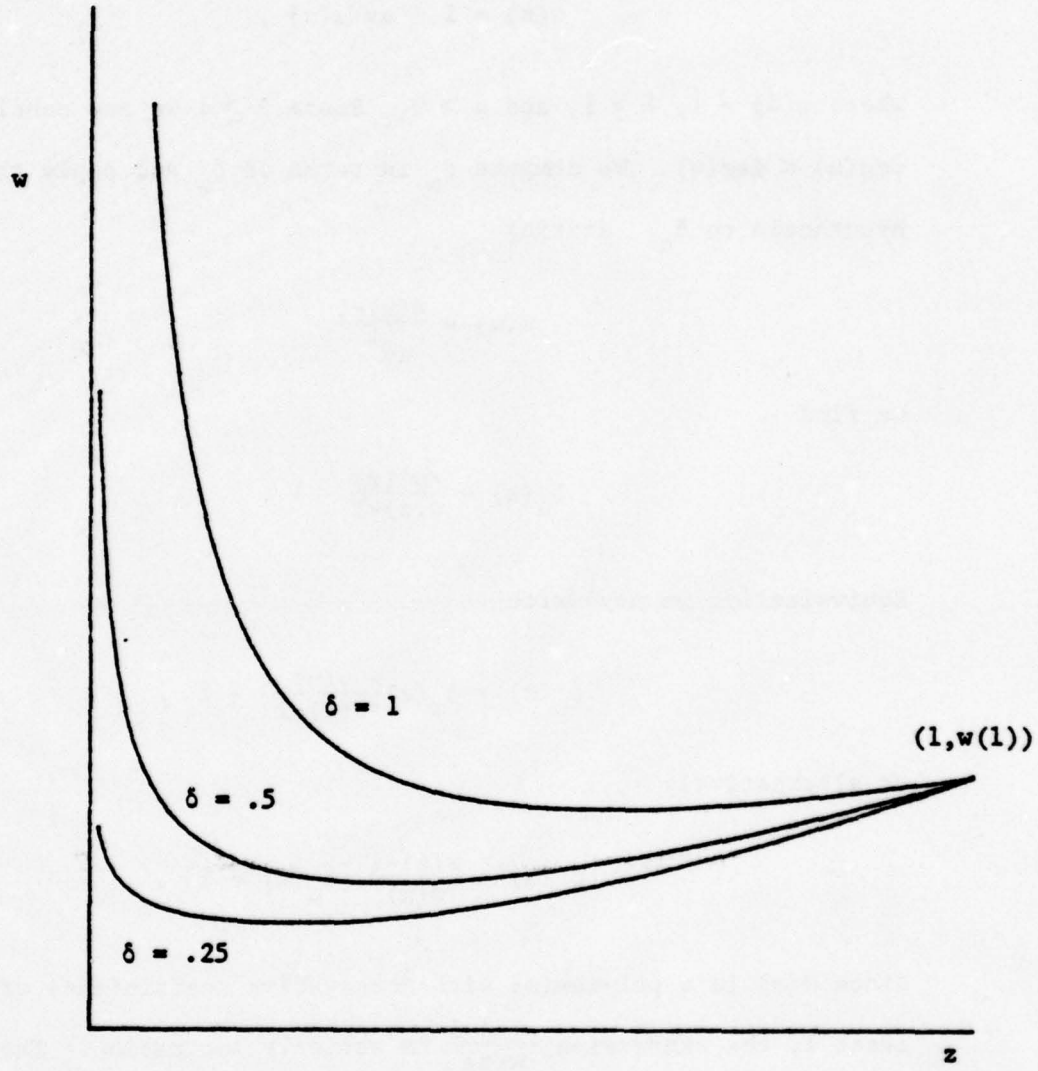


Figure 2. The family of curves $z^{-\delta}w(z)$. The function $w(z)$ is a polynomial with non-negative coefficients. The graphs shown are three members of this family corresponding to $\delta = .25, .5$, and 1.

proposition for all polynomials whose degree is less than k, where $k \geq 2$. We may then write w(x), a polynomial of degree k, as

$$w(x) = 1 + ax^{\lambda}u(x) ,$$

where $u(0) = 1$, $\lambda \geq 1$, and $a > 0$. Since $\lambda \geq 1$ we may conclude that $\deg(u) < \deg(w)$. We compute δ_u in terms of δ_w and apply the induction hypothesis to δ_u. Writing

$$u(x) = \frac{w(x)-1}{ax^{\lambda}}$$

we find

$$\delta_u(z) = \frac{zw'(z)}{w(z)-1} - \lambda .$$

Equivalently, we may write

$$\delta_u(z) = \delta_w(z)\{\frac{w(z)}{w(z)-1}\} - \lambda ,$$

or alternatively

$$\delta_w(z) = \frac{w(z)-1}{w(z)} \{\delta_u(z) + \lambda\} .$$

Since w(z) is a polynomial with nonnegative coefficients of degree at least 1, the expression $\frac{w(z)-1}{w(z)}$ is strictly increasing. The induction hypothesis guarantees that $\delta_u(z) + \lambda$ is strictly increasing provided $\deg(u) \geq 1$. If $\deg(u) = 0$ then $\delta_u(z) \equiv 0$. In either case, $\delta_w(z)$ is a strictly increasing function on (0,1). □

We note here that the condition $w(0) = 1$ in the last proposition can be omitted by observing that we may write

$$w(x) = ax^{t}w^{*}(x) ,$$

where $a > 0$, $t \geq 0$, and $w^*(0) = 1$; and that for any two differentiable functions f and g

$$\delta_{fg}(z) = \delta_f(z) + \delta_g(z) \ .$$

(This last equation is an easy consequence of $\delta_w(z) = z \cdot \frac{d}{dz} \log(w(z))$.)

We are now ready to establish the aforementioned properties of $z^{-\delta}w(z)$. The derivative of this last expression is $z^{-\delta-1}w(z)[\delta_w(z)-\delta]$. Since $\delta_w(z)$ is a strictly increasing function of z on (0,1), there is a unique solution to $\delta = \delta_w(z)$ provided that δ is between $\delta_w(0) = 0$ and $\delta_w(1) = \frac{w'(1)}{w(1)}$. Call this solution z^*. To the left of z^* the derivative of $z^{-\delta}w(z)$ is negative, and to the right of z^* it is positive. Hence z^* is a number in (0,1) that minimizes $z^{-\delta}w(z)$ for $0 \leq \delta \leq \frac{w'(1)}{w(1)}$.

Turning back to our main problem, we wish to show that there is a unique curve in the family $z^{-\delta}w(z)$ tangent to the line $y = K$. If we restrict the value of K to lie between 1 and w(1) we show that δ assumes a value between 0 and $\frac{w'(1)}{w(1)}$, so the point of tangency is the local minimum of the curve $z^{-\delta}w(z)$. The following proposition summarizes our ideas.

Proposition 2. If $1 \leq K \leq w(1)$ and w(x) satisfies the hypotheses of Proposition 1, then there exist unique real numbers z and δ (depending on both K and w(x)) satisfying $0 < z < 1$ and $0 < \delta < \frac{w'(1)}{w(1)}$ such that

$$K = z^{-\delta}w(z)$$

$$\delta = \frac{zw'(z)}{w(z)} \ .$$

Furthermore, this particular z minimizes the expression $z^{-\delta}w(z)$ for the value δ.

Proof. We may solve the system of equations

$$K = z^{-\delta} w(z)$$

$$\delta = \frac{zw'(z)}{w(z)}$$

for z to obtain the following equation:

$$0 = \log_2 z \left\{ \delta_w(z) - \frac{\log w(z) - \log K}{\log z} \right\} .$$

For convenience, let us refer to the right-hand side of this last equation as $F_w(z,K)$. It is easy to establish that for any two differentiable functions f and g and positive constants K and L,

$$F_{fg}(z,KL) = F_f(z,K) + F_g(z,L) ,$$

and

$$\frac{d}{dz} F_w(z,K) = \log_2 z \cdot \frac{d}{dz} \delta_w(z) .$$

Proposition 1 implies that $\frac{d}{dz} \delta_w(z) \geq 0$, so on (0,1) the function $\frac{d}{dz} F_w(z,K)$ is less than or equal to zero. Notice that the function $\delta_w'(z)$ is analytic as a function of the complex variable z along the line (0,1). Therefore, $\delta_w'(z)$ can equal zero only a finite number of times on (0,1). Thus on (0,1) the function $F_w(z,K)$ is strictly decreasing. Two further calculations reveal that

$$F_w(0,K) = \lim_{z \to 0^+} F_w(z,K) = \log_2 K - \log_2 w(0)$$

and

$$F_w(1,K) = \lim_{z \to 1^-} F_w(z,K) = \log_2 K - \log_2 w(1) .$$

Thus if $1 < K < w(1)$ we have $F_w(0,K) > 0$ and $F_w(1,K) < 0$. Hence a unique value of z exists on (0,1) that solves $0 = F_w(z,K)$. Furthermore, the corresponding value of δ given by $\frac{zw'(z)}{w(z)}$ lies between 0 and $\frac{w'(1)}{w(1)}$. These particular z and δ are the desired numbers. As $0 \leq \delta \leq w'(1)/w(1)$, the value z, in light of our previous discussion, clearly minimizes $z^{-\delta}w(z)$. □

The graphs in Figure 3 illustrate the functions $\delta_w(z)$ and $F_w(z,K)$ in the situation most frequently encountered (w(x) a polynomial satisfying Proposition 1 and $1 < K < w(1)$). To emphasize the fact that the numbers z and δ of Proposition 2 depend on both the value K and the polynomial w(x) we denote these quantities by z(w,K) and δ(w,K).

We are interested in applying Proposition 2 in the case that w(x) is the weight enumerator of an interleaved code. In this instance, we know that $w(x) = [w_0(x)]^s$, where $w_0(x)$ is the weight enumerator of the base code and s is the interleaving degree. There is a connection between the quantities $z(w,K^s)$ and $z(w_0,K)$ and between the quantities $\delta(w,K^s)$ and $\delta(w_0,K)$. This relationship is given by the following proposition.

Proposition 3. If $w_0(x)$ and K satisfy the hypothesis of Proposition 2 then

$$z(w_0^s,K^s) = z(w_0,K)$$

$$\delta(w_0^s,K^s) = s\delta(w_0,K) .$$

Proof. The number $z(w_0^s,K^s) = z^*$ is the unique solution of $0 = F_W(z,K^s)$, where $W = w_0^s$. From the remarks made at the time $F_w(z,K)$ was defined, we know that $F_w(z,K)$ is "logarithmic". Hence,

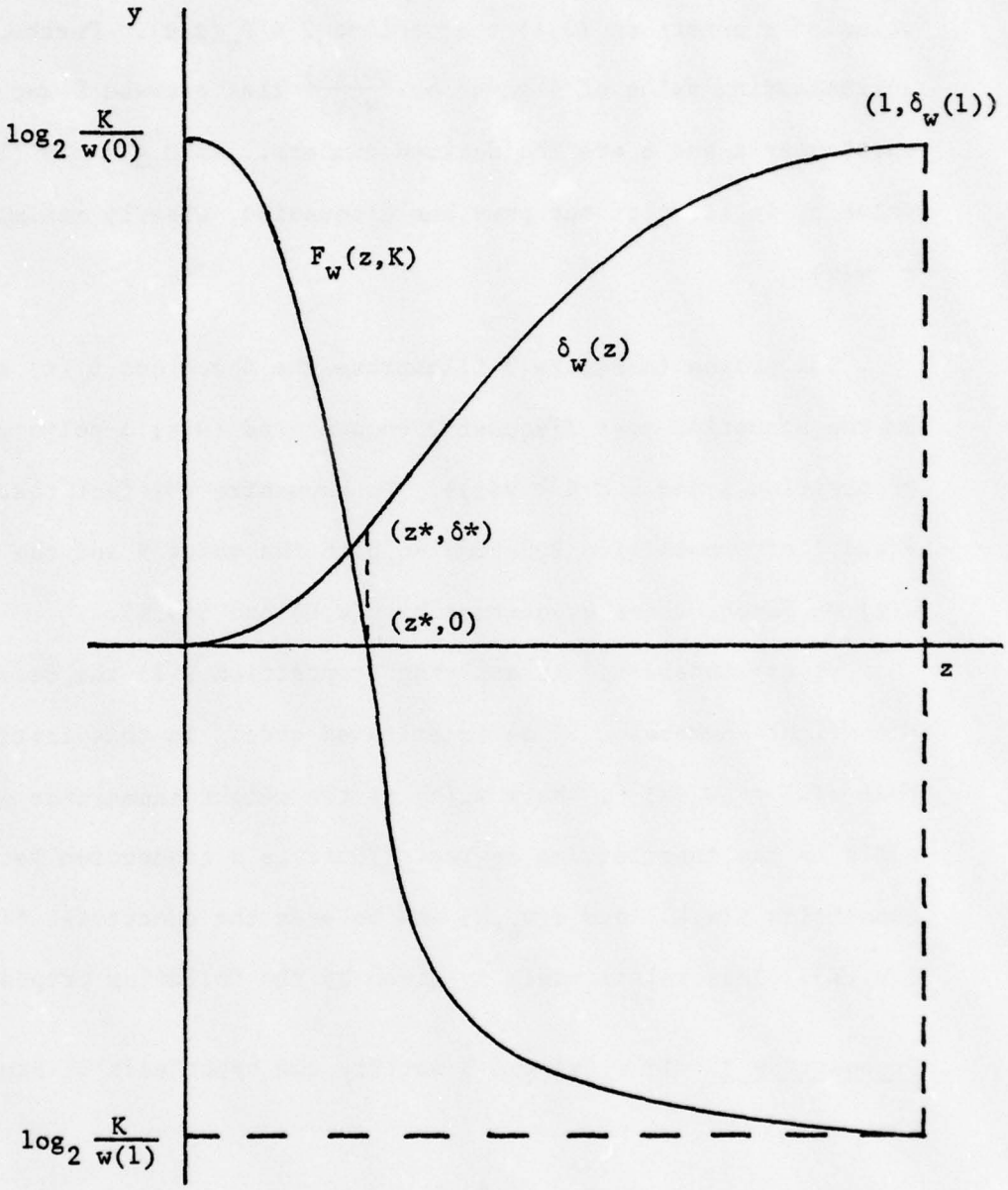


Figure 3. A graph showing the typical behaviour of the functions $F_w(z,K)$ and $\delta_w(z)$. The numbers z^* and δ^* are the values whose existence is guaranteed by Proposition 2.

$$F_W(z,K^s) = sF_{w_0}(z,K) .$$

The uniqueness of both $z(w_0^s,K^s)$ and $z(w_0,K)$ imply that $z(w_0^s,K^s) = z(w_0,K)$. Furthermore, the function $\delta_w(z)$ is also "logarithmic". Hence we have

$$\delta_W(z) = s\delta_{w_0}(z) .$$

Therefore

$$\delta(w_0^s,K^s) = \delta_W(z(w_0^s,K^s)) = s\delta_{w_0}(z(w_0,K)) = s\delta(w_0,K) . \quad \square$$

We are now ready to prove our main counting theorem. The devices used in the proof were first discovered by Weldon in his aforementioned paper [7] although the actual statement of this theorem was never given. We generalize Weldon's results as well as simplify his construction somewhat. Of course, the problems we encounter at the end are similar too, since all of our results contain the quantity $\delta(w_0,K)$ for certain specific values of the parameter K. The drawback of this method is that $\delta(w_0,K)$ has a vague combinatorial significance. It would be desirable to express $\delta(w_0,K)$ in terms of more concrete variables such as k_0 or d_0 but at this writing, no strong estimates of $\delta(w_0,K)$ are known to the author.

Weldon's Counting Lemma. Let m be a positive integer and let each α in $GF(2^m)$ be represented by a binary column vector of length m. Suppose that for $1 \leq i \leq \lambda$ and $2 \leq r$ the vectors $\underline{x}_i = [\alpha_{i,1},\alpha_{i,2},\ldots,\alpha_{i,r}]^T$ form a collection of λ distinct column vectors of length r, where each $\alpha_{i,j}$ is an element of $GF(2^m)$; and that A is a generator matrix for a full $[n_0s,k_0s]$ interleaved code having weight enumerator $[w_0(x)]^s$. If $m = k_0s$ and $\lambda = \gamma 2^m$ where γ is a rational number between 0 and 1, then as $s \to \infty$ the

total weight of the λ vectors $\underline{y}_i = [A\alpha_{i,1}, A\alpha_{i,2}, \ldots, A\alpha_{i,r}]^T$ is at least $\gamma sr\delta^{(r-1)}2^m(1 + o(1))$, where $\delta^{(r-1)}$ (together with $z^{(r-1)}$) is the unique solution of

$$2^{k_0/r} = z^{-\delta}w_0(z)$$

$$\delta = \frac{zw_0'(z)}{w_0(z)} .$$

Proof. Let C_{s-1} denote an s-fold interleaved code generated by an $[n_0,k_0]$ base code C_0, and let A denote a generator matrix for C_{s-1}. Each vector $\underline{y}_i$ is an element of the r-fold juxtaposition of C_{s-1}. Keeping in mind that we are not distinguishing between interleaved or juxtaposed codes, we may write $\underline{y}_i \in C_{rs-1}$. The distinctness of the vectors $\underline{x}_i$ implies that the vectors $\underline{y}_i$ are also distinct. Let us now order the elements of C_{rs-1} by weight, breaking ties arbitrarily. Consider the first $\lfloor\frac{\lambda}{m}\rfloor$ "low weight" vectors in C_{rs-1}. We lower-bound the weight of each of these vectors by zero and lower-bound the weight of the next $\lambda - \lfloor\frac{\lambda}{m}\rfloor$ vectors by δ^*, where δ^* is the weight of the $\lfloor\frac{\lambda}{m}\rfloor$-th vector. The weight enumerator of C_{rs-1} is

$$w^{rs}(x) = \sum_{j=0}^{rsn_0} W_j x^j = W(x) ,$$

and the number δ^* is the largest integer satisfying

$$\sum_{j=0}^{\delta^*} W_j \leq \frac{\lambda}{m} .$$

Since $m = sk_0$ and $W(1) = 2^{rsk_0}$ we have (for large m)

$$1 < \lfloor\frac{\lambda}{m}\rfloor = \lfloor\frac{\gamma}{m} 2^m\rfloor < 2^{rm} = W(1) .$$

Therefore for large m we can apply Proposition 2 to produce the numbers $z(W,\lfloor\frac{\lambda}{m}\rfloor)$, $\delta(W,\lfloor\frac{\lambda}{m}\rfloor)$ that solve the system

$$\lfloor\frac{\lambda}{m}\rfloor = z^{-\delta}W(z)$$

$$\delta = \frac{zW'(z)}{W(z)} .$$

To employ the Lemma of Partial Sums we must guarantee that $\delta(W,\lfloor\frac{\lambda}{m}\rfloor) \leq W'(1)/W(1)$. This last inequality is a consequence of the monotonicity of $\delta_W(z)$ over $(0,1)$. Specifically, we have

$$\delta(W,\lfloor\frac{\lambda}{m}\rfloor) = \delta_W(z(W,\lfloor\frac{\lambda}{m}\rfloor)) \leq \delta_W(1) = \frac{W'(1)}{W(1)} .$$

The Lemma of Partial Sums guarantees that for our choice of z and δ (namely $z = z(W,\lfloor\frac{\lambda}{m}\rfloor)$, $\delta = \delta(W,\lfloor\frac{\lambda}{m}\rfloor)$) we have

$$\sum_{j=0}^{\delta} W_j \leq z^{-\delta}W(z) = \lfloor\frac{\lambda}{m}\rfloor .$$

Since δ^* is the largest integer for which the last inequality holds, we conclude $\delta^* \geq \delta(W,\lfloor\frac{\lambda}{m}\rfloor)$.

We can now compute a lower bound for the total weight of the λ vectors $\underline{y}_i$. If this total weight is $\mathcal{W}$ we have

$$\mathcal{W} \geq \lfloor\frac{\lambda}{m}\rfloor \cdot 0 + (\lambda - \lfloor\frac{\lambda}{m}\rfloor)\,\delta^* \geq \lambda(1 - \tfrac{1}{m})(\delta(W,\lfloor\frac{\lambda}{m}\rfloor)) .$$

Observe that $\delta(W,\lfloor\frac{\lambda}{m}\rfloor) = s\delta(w_0^r,K)$, where $K = (\lfloor\frac{\lambda}{m}\rfloor)^{1/s}$. This gives $\mathcal{W} \geq \lambda(1 - \frac{1}{m})(s\delta(w_0^r,K))$. The number $z(w_0^r,K)$ is the unique solution of

$$0 = F_a(z,K) = \log_2 z \cdot \delta_a(z) - \log_2 a(z) + \tfrac{1}{s}\log_2\lfloor\tfrac{\lambda}{m}\rfloor ,$$

where $a(x) = [w_0(x)]^r$. As $s \to \infty$ the term

$$\frac{1}{s}\log_2\lfloor\frac{\lambda}{m}\rfloor = \frac{1}{s}\{k_0 s + \log_2\frac{\gamma}{sk_0} + o(1)\}$$

tends to k_0. By virtue of the continuity of F and the implicit function theorem, we have for $s \to \infty$

$$z(a,K) \to z(a,2^{k_0})$$

$$\delta(a,K) \to \delta(a,2^{k_0}) .$$

Hence we may write $\delta(a,K) = \delta(a,2^{k_0})(1 + o(1))$ as $s \to \infty$. The lower bound on $\mathcal{W}$ then becomes, as $s \to \infty$,

$$\mathcal{W} \geq \lambda(1 - \frac{1}{m})\delta(a,2^{k_0})(1 + o(1)) .$$

Noting that $\delta(a,2^{k_0})$ does not depend on s, that $\lambda = \gamma 2^m$, that $a(x) = [w_0(x)]^r$, and that $\delta(w_0^r,2^{k_0}) = r\delta^{(r-1)}$, we may write, as $s \to \infty$,

$$\mathcal{W} \geq \gamma sr\ \delta^{(r-1)}2^m(1 + o(1)) ,$$

where $\delta^{(r-1)}$ together with $z^{(r-1)}$ is the unique solution to

$$2^{k_0/r} = z^{-\delta}w_0(z)$$

$$\delta = \frac{zw_0'(z)}{w_0(z)} . \qquad \square$$

This is the Weldon Counting Lemma.

The next three propositions establish some elementary properties of the numbers $\delta^{(r-1)}$. In particular we show that $\delta^{(1)} \leq \frac{n_0}{4}$, that $\delta^{(i)} < \delta^{(j)}$ if and only if $i > j \geq 1$, and that for the code $\{0,1\}$, the numbers $\delta^{(r-1)}$ can be computed explicitly.

Proposition 4. If $w_0(x)$ is the weight enumerator for an $[n_0,k_0,d_0]$ linear code, then $\delta^{(1)} \leq \frac{n_0}{4}$.

Proof. Let m be a positive integer, $N = 2^m - 1$, and $GF(2^m)$ a finite field. Suppose α is a primitive element of $GF(2^m)$ and $[a_1,a_2,\ldots,a_N]$ is an element of a Reed-Solomon (RS) code of rate R over $GF(2^m)$ of length N. Build a Justesen Code letting a generic element have the form

$$J = \begin{bmatrix} a_1 & a_2 & a_3 & \cdots & a_N \\ \alpha a_1 & \alpha^2 a_2 & \alpha^3 a_3 & \cdots & \alpha^N a_N \end{bmatrix} .$$

Encode each of these 2N symbols with an s-fold interleaved code generated by the given $[n_0,k_0,d_0]$ code, choosing s so that $m = sk_0$. All of the non-zero columns of J are distinct, and we are guaranteed to have at least (1-R)N of them. The Weldon Counting Lemma can now be applied with $\lambda = (1-R)\frac{N}{N+1}2^m$ and r = 2 to lower bound the weight of J. We get

$$wt(J) \geq (1-R)\frac{N}{N+1} \cdot s \cdot \delta^{(1)} \cdot 2^m(1 + o(1)) .$$

Next we observe that $[1,1,\ldots,1]$ is in all RS codes of length N over $GF(2^m)$ so that every Justesen Code contains the vector

$$J^* = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \alpha & \alpha^2 & \alpha^3 & \cdots & \alpha^N \end{bmatrix} .$$

Choose an encoding scheme in such a way that the symbol 1 in $GF(2^m)$ is encoded into a binary vector of weight d_0. The second row of J*, after encoding, runs over all non-zero codewords in the s-fold interleaved code. Thus the total weight of J* is

$$d_0N + [w_0^s(x)]'|_{x=1} ,$$

or equivalently

$$d_0N + sw_0^{s-1}(1)w_0'(1) .$$

Under the standing hypothesis that the $[n_0,k_0,d_0]$ code is full, we have

$$wt(J^*) = d_0N + s2^{k_0(s-1)} \cdot n_0 2^{k_0-1} .$$

Hence

$$(1-R)\frac{N}{N+1} s \cdot 2 \cdot \delta^{(1)} \cdot 2^m(1 + o(1)) \leq wt(J^*) = d_0N + sn_0 2^{k_0-1} .$$

We may rewrite this (recalling that $m = sk_0$) as

$$2 \cdot \frac{N}{N+1} (1-R)\delta^{(1)} \cdot 2^m(1 + o(1)) \leq \frac{d_0N}{s2^m} + \frac{n_0}{2} .$$

Let $s \to \infty$ to get

$$2(1-R)\delta^{(1)} \leq \frac{n_0}{2} .$$

Since $\delta^{(1)}$ and n_0 are independent of R we conclude that

$2\delta^{(1)} \leq \frac{n_0}{2}$. □

Proposition 5. The quantity $\delta^{(i)} < \delta^{(j)}$ if and only if $i > j \geq 1$.

Proof. The numbers $\delta^{(i)}$ and $z^{(i)}$ form the unique solution of

$$2^{k_0/(i+1)} = z^{-\delta} w_0(z)$$

$$\delta = \frac{zw_0'(z)}{w_0(z)}$$

under the standing hypothesis that $w_0(x)$ is the weight enumerator of an $[n_0,k_0,d_0]$ full code. Furthermore $z^{(i)}$ is the unique solution to $F_{w_0}(z,K_i) = 0$ where $K_i = 2^{k_0/(i+1)}$. We observe that if $1 \leq j < i$ then

$$F_{w_0}(z,K_j) > F_{w_0}(z,K_i) .$$

Hence $F_{w_0}(z^{(i)},K_j) > 0$, which in turn implies $z^{(i)} < z^{(j)}$ since $F_{w_0}(z,K_j)$ is a strictly decreasing function of z on (0,1). The relation $\delta^{(i)} = \delta_{w_0}(z^{(i)})$ and the increasing nature of $\delta_{w_0}(z)$ as a function of z conclude the proof since

$$\delta^{(i)} = \delta_{w_0}(z^{(i)}) < \delta_{w_0}(z^{(j)}) = \delta^{(j)} . \qquad \square$$

To conclude the analysis of the counting procedure we present an evaluation of the numbers $\delta^{(r-1)}$ in the case $w_0(x) = 1 + x$ and $r \geq 2$. This corresponds to the trivial base code $\{0,1\}$. The values of these numbers will be used to check later computations against known results.

Proposition 6. If $w_0(x) = 1 + x$ and $r \geq 2$ then $\delta^{(r-1)} = H^{-1}(\frac{1}{r})$ where $H(\cdot)$ is the binary entropy function.

Proof. We must first find the corresponding value of $z^{(r-1)}$. This number is the solution of

$$0 = z \frac{\log_2 z}{1+z} - \log_2(1+z) + \frac{1}{r} .$$

We may rearrange this last equation to produce

$$(1+z) \cdot \frac{1}{r} = (1+z)\log_2(1+z) - z \log_2 z$$

$$= (1+z)H(\frac{1}{1+z}) .$$

Thus $H(\frac{z}{1+z}) = \frac{1}{r}$. However, we observe that $\delta_{1+x}(z) = \frac{z}{1+z}$ and $\delta^{(r-1)} = \delta_{1+x}(z^{(r-1)})$. We therefore conclude that $\delta^{(r-1)} = H^{-1}(\frac{1}{r})$. □

We made many attempts to lower-bound $\delta^{(r-1)}$ by quantities having a more concrete combinatorial significance. The best universal lower-bound we discovered is

$$\delta^{(r-1)} > \frac{d_0}{r} - \frac{d_0}{k_0} \quad ,$$

which relates the information content and minimum distance of an $[n_0,k_0,d_0]$ error correcting code to its corresponding values of $\delta^{(r-1)}$ for $r = 2,3,\ldots$. In many individual cases examined, we found that $\delta^{(1)}$ was usually close to the number $\frac{d_0}{2}$. In a few spectacular cases, $\delta^{(1)}$ was as large as d_0. In view of this evidence, some restriction on the weight distribution of the $[n_0,k_0,d_0]$ code is evidently required to force $\delta^{(1)}$ to assume values consistently larger than $\frac{1}{2}\, d_0$.

CHAPTER 3

THE GENERALIZED SKHN CONSTRUCTION AND THE CODES $\mathcal{K}^{(\ell)}$

In this chapter we present a class of codes which not only generalizes the class of codes produced by the SKHN construction, but also generalizes the classes of Justesen [4] and Weldon [7] as well. To this end, we reproduce the basic results of Sugiyama, et al. We start with a definition.

Definition. A vector $\underline{y} = (y_1, y_2, \ldots, y_N)$ is generated by an element $\beta \in GF(2^m)$ if there exist elements $x_1, x_2, \ldots, x_{N-1}$ in $GF(2^m)$ such that

$$y_k = \begin{cases} x_1 & \text{if } k = 1 \\ \beta\, x_{k-1} + x_k & \text{if } 2 \le k \le N-1 \\ \beta\, x_{N-1} & \text{if } k = N\,. \end{cases}$$

From this definition the following propositions are immediate.

Proposition 7. If $\underline{a} = (a_1, a_2, \ldots, a_N)$, $\underline{b} = (b_1, b_2, \ldots, b_N)$, and $\underline{c} = (c_1, c_2)$ are generated by β, then so is

$$\underline{y} = (a_1, a_2, \ldots, a_{N-1},\ a_N + c_1, c_2 + b_1, b_2, \ldots, b_N)\,.$$

Proof. We may write

$$a_k = \begin{cases} u_1 & \text{if } k = 1 \\ \beta\, u_{k-1} + u_k & \text{if } 2 \le k \le N-1 \\ \beta\, u_{N-1} & \text{if } k = N\,, \end{cases}$$

$$b_k = \begin{cases} v_1 & \text{if } k = 1 \\ \beta\, v_{k-1} + v_k & \text{if } 2 \le k \le N-1 \\ \beta\, v_{N-1} & \text{if } k = N\ , \end{cases}$$

and

$$c_k = \begin{cases} w_1 & \text{if } k = 1 \\ \beta w_1 & \text{if } k = 2\ , \end{cases}$$

with u_i, v_i, and w_1 in $GF(2^m)$. The vector $\underline{y}$ has the proper form in the first N-1 and last N-1 coordinates. It suffices to check the middle two positions. We observe that $a_N + c_1 = \beta u_{N-1} + w_1$ and $c_2 + b_1 = \beta w_1 + v_1$, and therefore $\underline{y}$ has the proper form. □

Proposition 8. In the notation of the last proposition, $\underline{c} \neq \underline{0}$ implies $\underline{y} \neq \underline{0}$.

Proof. If $\underline{c} \neq \underline{0}$ and $\underline{y} = \underline{0}$, then $a_k = 0$ for $1 \le k \le N-1$ and $b_k = 0$ for $2 \le k \le N$. This implies that $u_k = 0$ for $1 \le k \le N-1$ and $v_k = 0$ for $1 \le k \le N-1$. Furthermore

$$0 = a_N + c_1 = \beta u_{N-1} + w_1 = \beta \cdot 0 + w_1 = w_1\ .$$

Therefore $\underline{c} = \underline{0}$ contrary to hypothesis. □

Proposition 9. Let $\underline{y}_1, \underline{y}_2, \ldots, \underline{y}_{M_0}$ be a collection of N-component vectors. Suppose that each vector $\underline{y}_i$ is generated by $\beta_i \in GF(2^m)$ and that the β_i are distinct. If $N \ge 2$ (to avoid trivialities), then at most N-1 of the non-zero $\underline{y}_i$ are identical.

Proof. Since $\underline{y}_i$ is generated by β_i we can write $\underline{y}_i = (y_{i,1}, y_{i,2}, \ldots, y_{i,N})$, where

$$y_{i,k} = \begin{cases} x_{i,1} & \text{if } k = 1 \\ \beta_i x_{i,k-1} + x_{i,k} & \text{if } 2 \le k \le N-1 \\ \beta_i x_{i,N-1} & \text{if } k = N \ , \end{cases}$$

and $x_{i,k} \in GF(2^m)$ for $1 \le i \le M_0$ and $1 \le k \le N$. If we evaluate the expression $\sum_{j=0}^{N-1} y_{i,N-j}\beta_i^j$ by replacing $y_{i,k}$ with $x_{i,k}$ via the above equations we find that $\sum_{j=0}^{N-1} y_{i,N-j}\beta_i^j = 0$. Hence the quantity β_i is a root of the polynomial $p_i(z) = \sum_{j=0}^{N-1} y_{i,N-j}z^j$ over $GF(2^m)$. If N or more vectors $\underline{y}_i$ are identical and non-zero then the polynomial $p_i(z)$ is not identically zero and has N or more roots. This contradiction leads us to the conclusion that no more than N-1 of the $\underline{y}_i$ are identical. □

These three propositions are the foundation upon which the code construction lies. They were first presented in the paper of Sugiyama, et al. and are repeated here for the sake of completeness. Of the three, the last proposition is the most crucial to the success of the construction, for it limits the number of repetitions of low weight inner codewords in our final concatenation.

We describe our code in two stages for ease of comprehension. Initially we describe the code $\mathcal{X}^{(2)}$, and then the codes $\mathcal{X}^{(\ell)}$ for $\ell = 1,2,3,\ldots$.

<u>The Code $\mathcal{X}^{(2)}$</u>. Let $M = 2^m$. We begin with two maximum distance separable (MDS) codes over the symbol alphabet $GF(2^m)$. The first MDS code $\mathcal{M}_1$ has length M and rate $R^{(1)}$ and the second MDS code $\mathcal{M}_2$ has length M/2 and rate $R^{(2)}$. Choose M/2 distinct non-zero elements from $GF(2^m)$ and call them $\alpha_1, \alpha_2, \ldots, \alpha_{M/2}$. The codewords of $C^{(2)}$ (the Sugiyama, et al. code) are divided into two types. The type 1 codewords have the form

$$T_1 = \begin{bmatrix} a_1^{(1)} & a_2^{(1)} & \cdots & a_{\frac{M}{2}}^{(1)} & a_{\frac{M}{2}+1}^{(1)} & \cdots & a_M^{(1)} \\ \alpha_1 a_1^{(1)} & \alpha_2 a_2^{(1)} & \cdots & \alpha_{\frac{M}{2}} a_{\frac{M}{2}}^{(1)} & \alpha_1 a_{\frac{M}{2}+1} & \cdots & \alpha_{\frac{M}{2}} a_M^{(1)} \end{bmatrix}$$

where $[a_1^{(1)}, a_2^{(1)}, \ldots, a_M^{(1)}] \in \mathcal{M}_1$. The type 2 codewords have the form

$$T_2 = \begin{bmatrix} a_1^{(1)} & a_2^{(1)} & \cdots & a_{\frac{M}{2}}^{(1)} \\ \alpha_1 a_1^{(1)} & \alpha_2 a_2^{(1)} & \cdots & \alpha_{\frac{M}{2}} a_{\frac{M}{2}}^{(1)} \\ a_{\frac{M}{2}+1}^{(1)} & a_{\frac{M}{2}+2}^{(1)} & \cdots & a_M^{(1)} \\ \alpha_1 a_{\frac{M}{2}+1}^{(1)} & \alpha_2 a_{\frac{M}{2}+2}^{(1)} & \cdots & \alpha_{\frac{M}{2}} a_M^{(1)} \end{bmatrix} + \begin{bmatrix} 0 & 0 & \cdots & 0 \\ a_1^{(2)} & a_2^{(2)} & \cdots & a_{\frac{M}{2}}^{(2)} \\ \alpha_1 a_1^{(2)} & \alpha_2 a_2^{(2)} & \cdots & \alpha_{\frac{M}{2}} a_{\frac{M}{2}}^{(2)} \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

where $[a_1^{(1)}, a_2^{(1)}, \ldots, a_M^{(1)}] \in \mathcal{M}_1$ and $[a_1^{(2)}, a_2^{(2)}, \ldots, a_{\frac{M}{2}}^{(2)}] \in \mathcal{M}_2$. Notice that, except for arrangement, type 1 codewords are a special case of type 2 codewords and correspond to $[a_1^{(2)}, a_2^{(2)}, \ldots, a_{\frac{M}{2}}^{(2)}] = \underline{0}$. Both types of codewords involve 2M symbols from $GF(2^m)$.

Each type 1 codeword is a $2 \times M$ matrix that has at least $(1-R^{(1)})M$ non-zero columns. The first $\frac{M}{2}$ columns form a collection of vectors each of which is generated by one of the elements $\alpha_1, \alpha_2, \ldots, \alpha_{M/2}$. The last $\frac{M}{2}$ columns form a similar collection of vectors. By Proposition 9 all columns in the first half of T_1 are distinct as are the columns in the last half of T_1.

The codewords of type 2 are $4 \times \frac{M}{2}$ matrices. Each column of T_2 is generated by one of the elements $\alpha_1, \alpha_2, \ldots, \alpha_{M/2}$ according to Proposition 7, with

$$\underline{a} = \begin{bmatrix} a_i^{(1)} \\ \alpha_i a_i^{(1)} \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} a_{\frac{M}{2}+i}^{(1)} \\ \alpha_i a_{\frac{M}{2}+i}^{(1)} \end{bmatrix}, \quad \text{and} \quad \underline{c} = \begin{bmatrix} a_i^{(2)} \\ \alpha_i a_i^{(2)} \end{bmatrix}$$

for $1 \leq i \leq \frac{M}{2}$. Since $[a_1^{(2)}, a_2^{(2)}, \ldots, a_{M/2}^{(2)}] \in \mathcal{H}_2$ at least $(1\text{-}R^{(2)})\frac{M}{2}$ of the columns of

$$\begin{bmatrix} 0 & 0 & \ldots & 0 \\ a_1^{(2)} & a_2^{(2)} & \ldots & a_{\frac{M}{2}}^{(2)} \\ \alpha_1 a_1^{(2)} & \alpha_2 a_2^{(2)} & \ldots & \alpha_{\frac{M}{2}} a_{\frac{M}{2}}^{(2)} \\ 0 & 0 & \ldots & 0 \end{bmatrix}$$

are non-zero. By Proposition 8 then, at least $(1\text{-}R^{(2)})\frac{M}{2}$ of the columns of T_2 are non-zero. Furthermore, as all of the α_i are distinct, Proposition 9 assures us that among the non-zero columns of T_2, each column has no more than two duplicates.

At this stage, we deviate from the construction of Sugiyama, et al. We create the binary code $\mathcal{X}^{(2)}$ by encoding each $GF(2^m)$ symbol found in either a type 1 or a type 2 codeword by an $[n_0 s, k_0 s]$ interleaved code. To do this, all we need is that $m = k_0 s$, and we can guarantee this by the proper selection of s. We may now use the Weldon Counting Lemma to compute a lower-bound on the minimum distance in $\mathcal{X}^{(2)}$.

Since the code $\mathcal{K}^{(2)}$ is a linear code, it suffices to lower-bound the minimum weight of $\mathcal{K}^{(2)}$. First examine the type 1 codewords. There are at least $(1-R^{(1)})M$ non-zero columns, and each non-zero columns possesses at most one duplicate (notably a column in the first half of T_1 may be repeated in the second half). At worst we have two copies of each of the $\lambda_1 = \frac{1}{2}(1-R^{(1)})M$ distinct elements in a type 1 codeword. We employ the Weldon Counting Lemma with $\gamma = \frac{1}{2}(1-R^{(1)})$ and $r = 2$ to conclude that the total binary weight of the $\frac{1}{2}(1-R^{(1)})M$ columns that are guaranteed to be distinct is at least $\frac{1}{2}(1-R^{(1)})s2\delta^{(1)}2^m(1 + o(1))$ as $s \to \infty$. Thus the total binary weight of the $(1-R^{(1)})M$ non-zero (not necessarily distinct) columns is at least $2(1-R^{(1)})s\delta^{(1)}2^m(1 + o(1))$.

For a codeword of type 2 we have at least $(1-R^{(2)})\frac{M}{2}$ non-zero columns each of which has at most two duplicates. In the worst case there are exactly two duplications of each of $\lambda_2 = \frac{1}{3}(1-R^{(2)})\frac{M}{2}$ distinct non-zero columns. We can now apply the Weldon Counting Lemma to the $\frac{1}{3}(1-R^{(2)})\frac{M}{2}$ distinct non-zero columns (we have $\gamma = \frac{1}{3}(1-R^{(2)})\cdot\frac{1}{2}$ and $r = 4$) to conclude that the total binary weight of the distinct non-zero columns is at least $\frac{1}{3}(1-R^{(2)})\frac{1}{2}\, s4\delta^{(3)}M(1 + o(1))$. Hence the total binary weight of the $(1-R^{(2)})\frac{M}{2}$ (not necessarily distinct) columns is at least $(1-R^{(2)})s\delta^{(3)}2M(1 + o(1))$. Therefore the minimum distance $d^{(2)}$ in the code $\mathcal{K}^{(2)}$ is at least

$$\min\{(1-R^{(1)})s\delta^{(1)}2M, (1-R^{(2)})s\delta^{(3)}2M\}(1 + o(1)) .$$

The code $\mathcal{K}^{(2)}$ is a concatenated code. Its length is $(n_0 s)(2M)$ and its information content is $(k_0 s)(R^{(1)} + \frac{1}{2} R^{(2)})M$. The rate $r^{(2)}$ of $\mathcal{K}^{(2)}$ is clearly $r_0(\frac{1}{2} R^{(1)} + \frac{1}{4} R^{(2)})$, where $r_0 = \frac{k_0}{n_0}$; and the distance-to-length ratio satisfies

$$\frac{d^{(2)}}{2n_0 sM} \geq \min\{(1-R^{(1)})\,\frac{\delta^{(1)}}{n_0}\ ,\ (1-R^{(2)})\,\frac{\delta^{(3)}}{n_0}\}(1 + o(1))\ .$$

If we hold the quantities r_0, $R^{(1)}$, and $R^{(2)}$ constant and let $s \to \infty$ (which forces the overall length to go to infinity) we see that asymptotically $\mathcal{K}^{(2)}$ satisfies

$$\Delta^{(2)} = \liminf_{s \to \infty} \frac{d^{(2)}}{2n_0 sM} \geq \min\{(1-R^{(1)})\,\frac{\delta^{(1)}}{n_0}\ ,\ (1-R^{(2)})\,\frac{\delta^{(3)}}{n_0}\}$$

$$r^{(2)} = \frac{1}{2}\, r_0(R^{(1)} + \frac{1}{2}\, R^{(2)})\ .$$

We seek to maximize the right-hand side of the inequality involving $\Delta^{(2)}$. Since the equation for $r^{(2)}$ is a linear constraint of the variables $R^{(1)}$ and $R^{(2)}$ we can maximize the expression

$$\min\{(1-R^{(1)})\,\frac{\delta^{(1)}}{n_0}\ ,\ (1-R^{(2)})\,\frac{\delta^{(3)}}{n_0}\}$$

by choosing $R^{(1)}$ and $R^{(2)}$ so that

$$(1-R^{(1)})\,\frac{\delta^{(1)}}{n_0} = (1-R^{(2)})\,\frac{\delta^{(3)}}{n_0}\ .$$

Elementary manipulations then reveal that $\mathcal{K}^{(2)}$ satisfies (asymptotically)

$$\Delta^{(2)} \geq \frac{1}{n_0}\,(\frac{1}{2\delta^{(1)}} + \frac{1}{4\delta^{(3)}})^{-1}(\frac{1}{2} - \frac{r^{(2)}}{r_0})\ ,$$

for $0 \leq r^{(2)} \leq \frac{3}{4}$.

If we choose $n_0 = k_0 = 1$, then we can identify $\mathcal{K}^{(2)}$ as the SKHN code $C^{(2)}$. In this case $w_0(x) = 1 + x$, and we know that $\delta^{(1)} = H^{-1}(\frac{1}{2})$ and

$\delta^{(3)} = H^{-1}(\frac{1}{4})$. Substituting these values into our equation yields the same lower-bound as found in the original SKHN construction. However, we can do better. If we choose, for example, the dual of a [63,12,24] code as the base code we produce $\delta^{(1)} = 9.05$ and $\delta^{(3)} = 4.48$. This creates a code $\mathcal{K}^{(2)}$ that lies above the Zyablov bound for rates between .205 and .537. This particular code also lies above the SKHN bound for rates between .061 and .441.

The Codes $\mathcal{K}^{(\ell)}$ for $\ell \geq 1$. In this section we present our construction for the codes $\mathcal{K}^{(\ell)}$. These codes generalize not only the codes of Sugiyama, et al., but also the codes of Weldon and Justesen. We adopt the notation $M = 2^m$, $L = 2^\ell$ and $J = 2^j$ throughout. We proceed initially in the same manner as in the SKHN construction.

Definition. The (j,ℓ)-th Justesen Code, $1 \leq j \leq \ell \leq m$, is a concatenated code consisting of an outer MDS code over $GF(2^m)$ of length $\mu = \frac{2M}{J}$ and rate $R^{(j)}$ and inner codes chosen from the class of randomly shifted codes. Specifically, each codeword of the (j,ℓ)-th Justesen Code has the form

$$\begin{bmatrix} a_1^{(j)} & \cdots & a_i^{(j)} & \cdots & a_\mu^{(j)} \\ \alpha_1 a_1^{(j)} & \cdots & \alpha_{i^*} a_i^{(j)} & \cdots & \alpha_{\mu^*} a_\mu^{(j)} \end{bmatrix}$$

where $[a_1^{(j)}, a_2^{(j)}, \ldots, a_\mu^{(j)}]$ is an element of an MDS code over $GF(2^m)$ of length $\mu = \frac{2M}{J}$ and rate $R^{(j)}$, the elements $\alpha_1, \alpha_2, \ldots, \alpha_{\frac{2M}{L}}$ are distinct members of $GF(2^m)$, and i^* is the least positive residue of i modulo $\frac{2M}{L}$.

In the construction of the code $\mathcal{K}^{(\ell)}$ we start with the $(1,\ell)$-th Justesen Code and recursively perform an operation we call half-stacking.

This procedure transforms an $r \times 2s$ matrix into a $2r \times s$ matrix by cutting the matrix in half along its rows and moving the entire right half of the matrix below the left half. Figure 4 illustrates this procedure more clearly than words ever can.

We define a sequence of codes $A_\ell^{(1)}, A_\ell^{(2)}, \ldots, A_\ell^{(\ell)}$ by taking $A_\ell^{(1)}$ to be the $(1,\ell)$-th Justesen Code and proceeding inductively. Let us assume that for some $j \geq 2$ the code $A_\ell^{(j-1)}$ has been defined. The code $A_\ell^{(j)}$ is produced by half-stacking each element of $A_\ell^{(j-1)}$ and adding to these rearranged matrices a $J \times \frac{2M}{J}$ matrix of the form

$$\begin{bmatrix} \underline{0} \\ \vdots \\ \underline{0} \\ \underline{x} \quad \underline{y} \\ \underline{0} \\ \vdots \\ \underline{0} \end{bmatrix}$$

where (i) $\left[\frac{\underline{x}}{\underline{y}}\right]$ is an element of the (j,ℓ)-th Justesen Code, and (ii) $\underline{x}$ is in row $\frac{J}{2}$. Henceforth, we denote the matrix $\left[\frac{\underline{x}}{\underline{y}}\right]$ by $\left[\frac{\underline{x}}{\underline{y}}\right]_j$ to emphasize its membership in the (j,ℓ)-th Justesen Code.

We turn our attention to the code $A_\ell^{(\ell)}$. The form of the codewords of $A_\ell^{(\ell)}$ is that of $L \times \frac{2M}{L}$ matrices. The following proposition makes this construction work.

Proposition 10. If each element $\underline{c}$ in $A_\ell^{(\ell)}$ is an $L \times \frac{2M}{L}$ matrix, then for $1 \leq i \leq \frac{2M}{L}$ the i-th column c_i of $\underline{c}$ is generated by α_i.

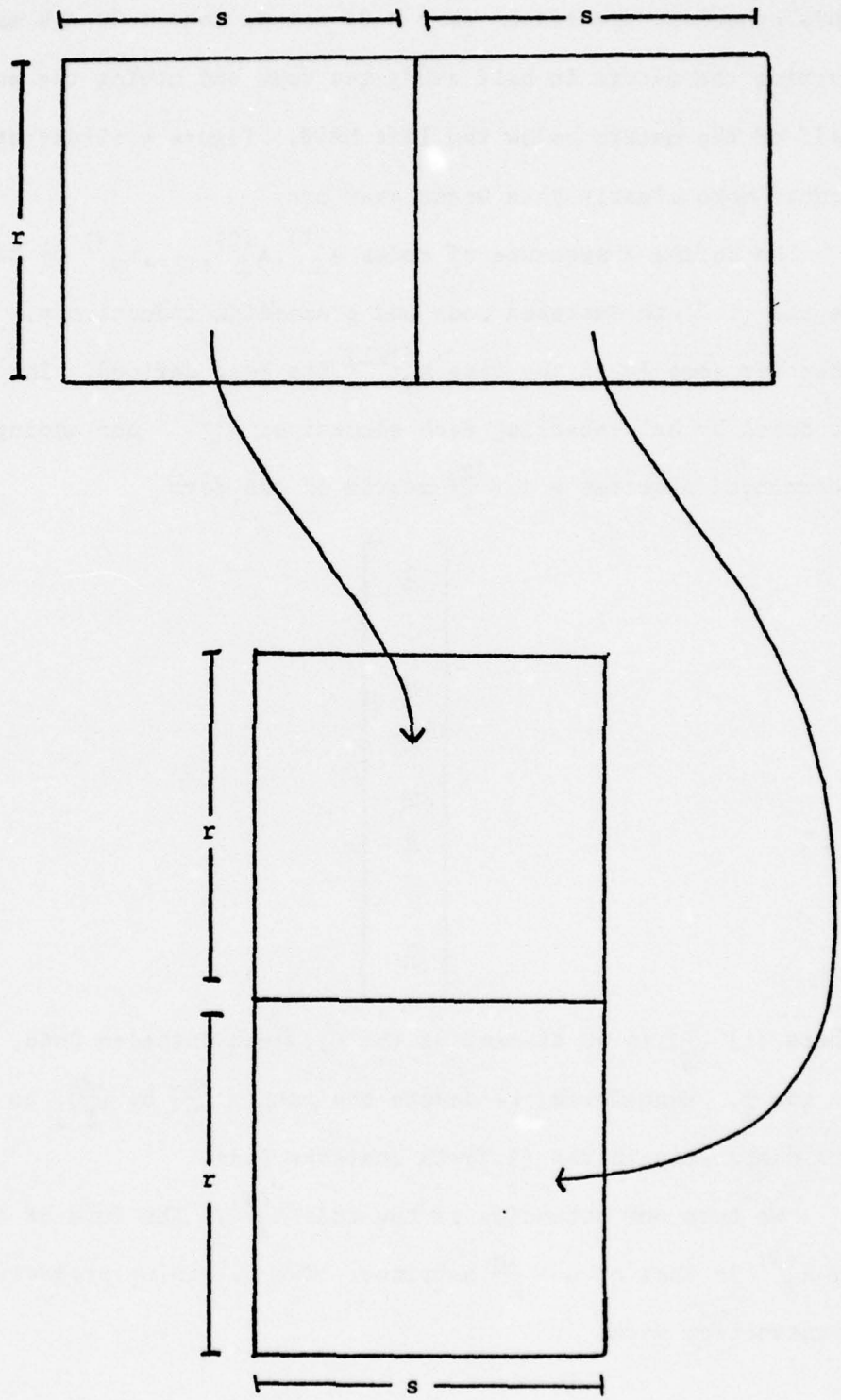


Figure 4. The half-stacking of an r×2s matrix to produce an 2r×s matrix.

Proof. We show that the i-th column of each element of $A_\ell^{(j)}$ is generated by α_{i*}. This is certainly true if $j = 1$ since $A_\ell^{(1)}$ is the $(1, \ell)$-th Justesen Code. Assume that for some $j \geq 2$ the k-th column of $A_\ell^{(j-1)}$ is generated by α_{k*} for $1 \leq k \leq \frac{4M}{J}$. If $1 \leq i \leq \frac{2M}{J}$ then the i-th column of an element from $A_\ell^{(j)}$ has the form

$$\begin{bmatrix} y_{1,i} \\ y_{2,i} \\ \vdots \\ y_{\frac{J}{2},i} \\ y_{1,i+\frac{2M}{J}} \\ y_{2,i+\frac{2M}{J}} \\ \vdots \\ y_{\frac{J}{2},i+\frac{2M}{J}} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ a_i^{(j)} \\ \alpha_{i*}a_i^{(j)} \\ 0 \\ \vdots \\ 0 \end{bmatrix} . \tag{6}$$

By induction the vectors $\begin{bmatrix} y_{1,i} \\ \vdots \\ y_{\frac{J}{2},i} \end{bmatrix}$ and $\begin{bmatrix} y_{1,i+\frac{2M}{J}} \\ \vdots \\ y_{\frac{J}{2},i+\frac{2M}{J}} \end{bmatrix}$ are both generated by α_{i*}, as is the vector $\begin{bmatrix} a_i^{(j)} \\ \alpha_{i*}a_i^{(j)} \end{bmatrix}$. By Proposition 7 the vector in Eq. (6) is also generated by α_{i*}. Thus the i-th column of $A_\ell^{(j)}$ is

generated by α_{i*} for $j = 1,2,\ldots,\ell$. Hence the i-th column of $A_\ell^{(\ell)}$ is generated by $\alpha_{i*} = \alpha_i$. □

To ease the burden of counting the minimum weight in the code $\mathcal{X}^{(\ell)}$ (still to be defined) we divide the elements of $A_\ell^{(\ell)}$ into ℓ classes.

Definition. We say that the codeword $\underline{c} \in A_\ell^{(\ell)}$ is of type j if $\left[\frac{x}{y}\right]_j \neq 0$ but $\left[\frac{x}{y}\right]_{j+1} = \left[\frac{x}{y}\right]_{j+2} = \ldots = \left[\frac{x}{y}\right]_\ell = 0$.

If $\underline{c}$ is of type j then it is the result of repeated half-stackings of some element of $A_\ell^{(j)}$. We are now ready to define $\mathcal{X}^{(\ell)}$.

Definition. Let C be any binary $[n_0,k_0,d_0]$ code. The code $\mathcal{X}^{(\ell)}(C)$ (or just $\mathcal{X}^{(\ell)}$ when the exact nature of the code C is not important) is a concatenated code having $A_\ell^{(\ell)}$ as outer code and an s-degree interleaved code based on C as inner code.

Notice that the inner code has length $n_0 s$ and information content $k_0 s$. Furthermore, the outer code is one defined on the symbol alphabet $GF(2^m)$ so that concatenation may be done provided $m = k_0 s$.

We first note that $\mathcal{X}^{(\ell)}$ is linear and that its length is $2n_0 sM$. The information content $\mathcal{X}^{(\ell)}$ is clearly

$$m(R^{(1)}M + R^{(2)} \cdot \frac{M}{2} + R^{(3)} \cdot \frac{M}{4} + \ldots + R^{(\ell)} \cdot \frac{M}{2^{\ell-1}})$$

and hence the rate $r^{(\ell)}$ of $\mathcal{X}^{(\ell)}$ is given by

$$r^{(\ell)} = r_0 \sum_{j=1}^{\ell} 2^{-j} R^{(j)} ,$$

where $r_0 = \frac{k_0}{n_0}$. To compute the minimum weight of $\mathcal{X}^{(\ell)}$ we again use the Weldon Counting Lemma, applying it separately for each of the ℓ types of

codewords in $A_\ell^{(\ell)}$. (From here on we will say that a member $\underline{x}$ of $\mathcal{X}^{(\ell)}$ is of <u>type j</u> if a type j outer codeword is concatenated with the inner codes to produce $\underline{x}$.) This gives us the following proposition.

<u>Proposition 11</u>. If $\underline{w}$ is an element of $\mathcal{X}^{(\ell)}$ of type j, $1 \leq j \leq \ell$, then

$$wt(\underline{w}) \geq 2(1-R^{(j)})s\delta^{(J-1)}2^m(1 + o(1)) .$$

<u>Proof</u>. Let $\underline{c}$ be a type j element of $A_\ell^{(\ell)}$. The word $\underline{c}$ is the result of repeated half-stackings of some element of $A_\ell^{(j)}$. Hence we may view $\underline{c}$ as a $J \times \frac{2M}{J}$ matrix of the form

$$A + \begin{bmatrix} \underline{0} \\ \vdots \\ \underline{0} \\ \underline{x} \\ \underline{y} \\ \underline{0} \\ \vdots \\ \underline{0} \end{bmatrix}$$

where $\left[\frac{\underline{x}}{\underline{y}}\right]_j$ is a non-zero element of the (j,ℓ)-th Justesen Code. The i-th column of $\underline{c}$ is generated by α_{i*} from Proposition 10, and therefore by Proposition 8 at least $(1-R^{(j)})\frac{2M}{J}$ of the columns of $\underline{c}$ are non-zero.

Section the columns of $\underline{c}$ into L/J blocks corresponding to the first $\frac{2M}{L}$ columns, the second $\frac{2M}{L}$ columns, and so forth. Each of the non-zero columns in the i-th block is generated by one of the elements $\alpha_1, \alpha_2, \ldots, \alpha_{2M/L}$. Since these elements are distinct and since the length of each column of $\underline{c}$ is J, we know that there can be at most (J-1)

duplications of any column in the i-th block. At worst the i-th block contains only σ_i/J distinct non-zero columns, where σ_i is the total number of non-zero columns in the i-th block. We have no way of estimating the size of σ_i, but we are guaranteed that not all of the σ_i are small since

$$\sum_{i=1}^{L/J} \sigma_i \geq (1-R^{(j)}) \frac{2M}{J} \quad .$$

To form a generic element $\underline{w}$ of $\mathcal{X}^{(\ell)}$ of type j it suffices to encode the symbols forming $\underline{c}$ by the chosen interleaved code. This does not disturb the zero-nature or the distinctness of any of the columns of $\underline{c}$. That is, the number of non-zero columns and the pattern of repeated columns are unchanged by the encoding. We will apply the Weldon Counting Lemma to the σ_i/J non-zero binary columns in the i-th block that we know are distinct, keeping in mind that the weight of the i-th block is J times more than the estimate given for the distinct columns. We take $\lambda_i = \sigma_i/J = \gamma_i 2^m$ with $0 \leq \gamma_i < 1$ and $r = J$ in the application of the lemma to bound the weight of the i-th block below by

$$J\{\gamma_i Js\delta^{(J-1)}2^m(1 + o(1))\} \ .$$

Thus the total weight of $\underline{w}$ is bounded below by

$$\sum_{i=1}^{L/J} \sigma_i Js\delta^{(J-1)}(1 + o(1)) \ .$$

Using the inequality

$$\sum_{i=1}^{L/J} \sigma_i \geq (1-R^{(j)}) \frac{2M}{J} \ ,$$

we can achieve a lower-bound of

$$2(1-R^{(j)})s\delta^{(J-1)}M(1 + o(1))$$

for the elements of $\mathcal{X}^{(\ell)}$ of type j. □

This last proposition allows us to lower-bound the minimum distance $d^{(\ell)}$ by the minimum of the ℓ numbers

$$2(1-R^{(j)})s\delta^{(J-1)}M(1 + o(1)) , \quad j = 1,2,\ldots,\ell .$$

If we hold the rate $r^{(\ell)}$ of $\mathcal{X}^{(\ell)}$ constant by fixing the values of r_0, $R^{(1)},R^{(2)},\ldots,R^{(\ell)}$ and allow $s \to \infty$, we find that the asymptotic distance-to-length ratio $\Delta^{(\ell)}$ is lower-bounded by

$$\min_{1 \le j \le \ell} \{(1-R^{(j)}) \frac{\delta^{(J-1)}}{n_0}\} .$$

The equation

$$r^{(\ell)} = r_0 \sum_{j=1}^{\ell} 2^{-j} R^{(j)}$$

introduces a linear restriction among the variables $R^{(1)},R^{(2)},\ldots,R^{(\ell)}$. Thus we can maximize the above lower-bound by choosing $R^{(1)},R^{(2)},\ldots,R^{(\ell)}$ so that for $j = 1,2,\ldots,\ell$,

$$(1-R^{(j)}) \frac{\delta^{(J-1)}}{n_0} = (1-R^{(1)}) \frac{\delta^{(1)}}{n_0} .$$

If we incorporate this restriction, the following asymptotic inequality results:

$$\Delta^{(\ell)} \ge \frac{1}{n_0} \Big(\sum_{\substack{j=1 \\ J=2^j}}^{\ell} \frac{1}{2^j \delta^{(J-1)}} \Big)^{-1} (1-2^{-\ell} - \frac{r^{(\ell)}}{r_0}) .$$

This bound agrees with the results of Sugiyama, et al. when we take the base code to be the trivial [1,1,1] code with weight enumerator $w_0(x) = 1 + x$. (Recall that we computed $\delta^{(J-1)} = H^{-1}(\frac{1}{J})$ in this case.) If in addition we take $\ell = 1$ we get Justesen's bound for his "unpunctured" codes:

$$\Delta^{(1)} \geq H^{-1}(\tfrac{1}{2})(1-2r^{(1)}) \ .$$

Moreover, if $\ell = 1$ and we use various base codes in our construction, we get

$$\Delta^{(1)} \geq \frac{\delta^{(1)}}{n_0} \left(1 - \frac{2r^{(1)}}{r_0}\right) ,$$

which is the same result achieved by Weldon in [7]. Hence, the codes $\chi^{(\ell)}$ generalize not only the SKHN construction but also the constructions of Weldon and Justesen as well.

Many base codes were tested numerically, and some produce codes whose curves lie above both the Zyablov bound and the SKHN bound for certain rates. The cases in which the codes $\chi^{(\ell)}$ are superior to the SKHN bound and the Zyablov bound occur by choosing base codes with large rates.

The merit of the codes $\chi^{(\ell)}$ is decided on the basis of two criteria; namely, the places where the codes $\chi^{(\ell)}$, $\ell = 1,2,3,\ldots$ meet either the Zyablov bound or the SKHN bound.

Two particular base codes are noteworthy. First of all, the dual of a [63,12,24] BCH code produces an intersection point of r = .205 with the Zyablov bound. Specifically, $\chi^{(2)}$ lies above both the Zyablov bound and the SKHN bound for rates between .205 and .441. Secondly, the dual of a

[91,12,36] BCH code produces codes $\mathcal{K}^{(2)}$ and $\mathcal{K}^{(3)}$ that lie above both the Zyablov and SKHN bounds for rates between .234 and .592. Figures 5 and 6 show the behavior of the codes $\mathcal{K}^{(\ell)}$ using both these base codes.

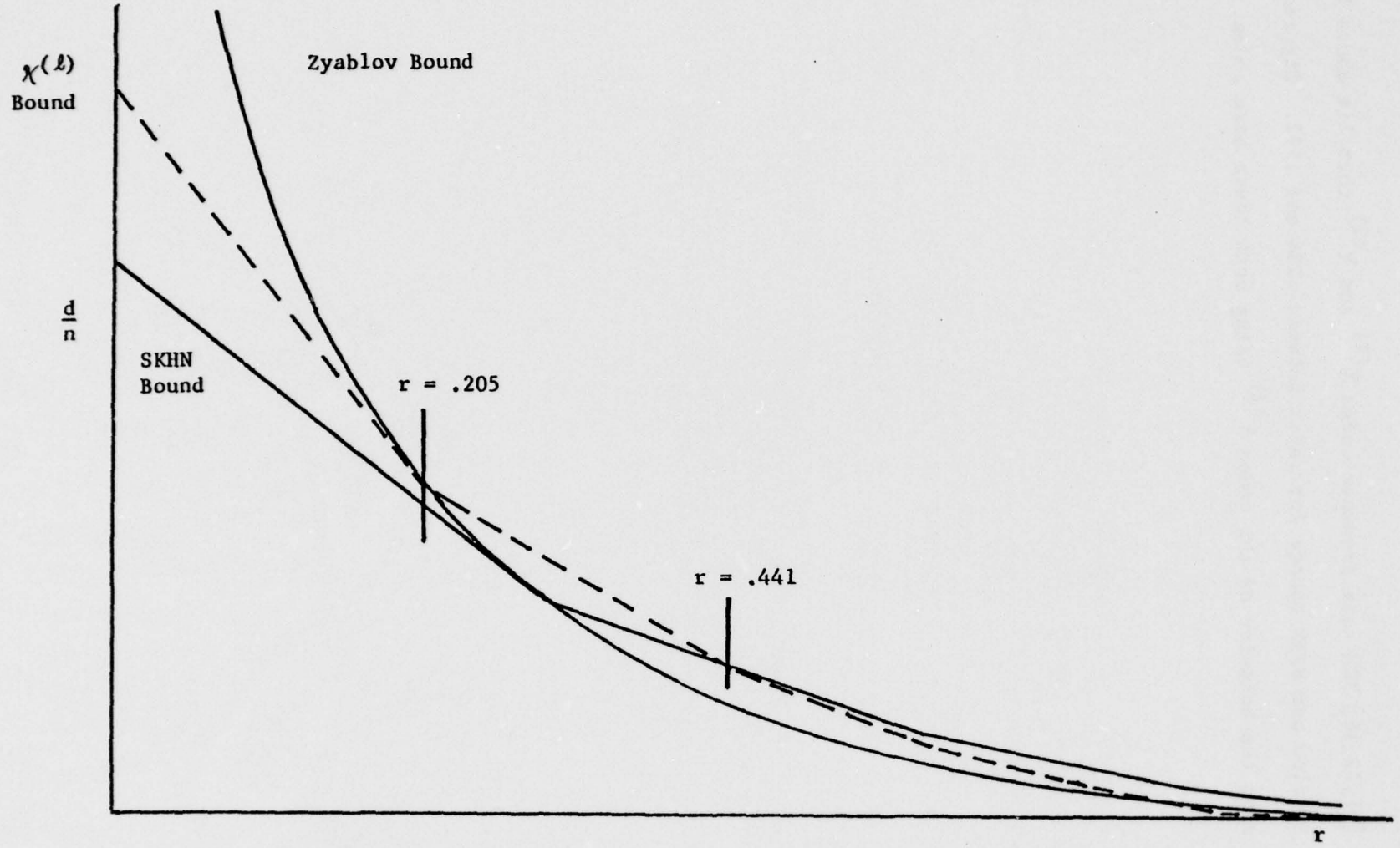


Figure 5. The SKHN bound and the bound for the code $\gamma^{(\ell)}$ are shown for $1 \leq \ell \leq 8$ and $0 < r < 1$. The code $\gamma^{(\ell)}$ is based on the dual of a [63,12,24] BCH code.

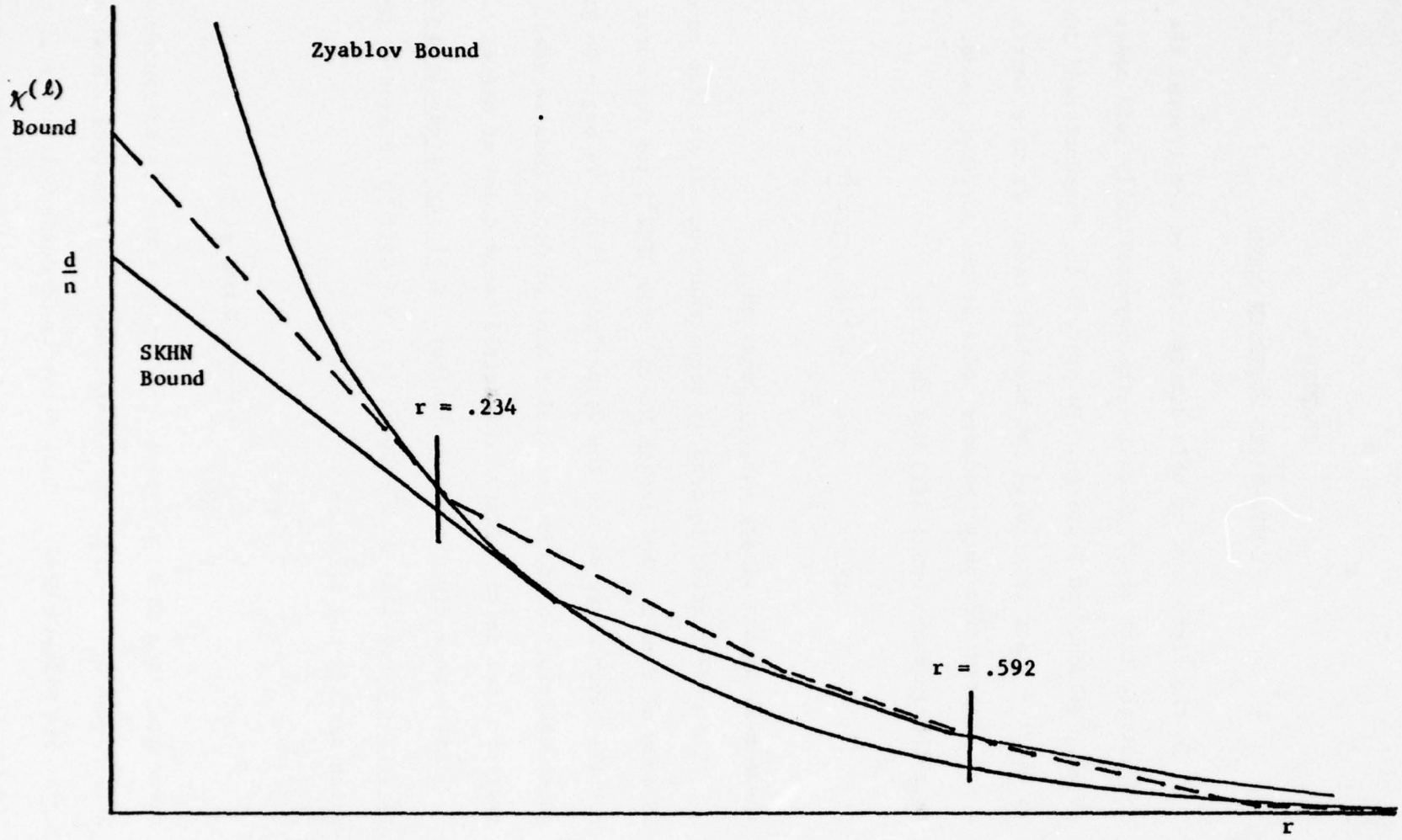


Figure 6. The SKHN bound and the bound for the code $\chi^{(\ell)}$ are shown for $1 \leq \ell \leq 8$ and $0 < r < 1$. The code $\chi^{(\ell)}$ is based on the dual of a [91,12,36] BCH code.

CHAPTER 4

GOPPA-BASED JUSTESEN CODES

In the first part of this dissertation we constructed the codes $\mathcal{K}^{(\ell)}$ and noticed that they generalize the asymptotically good codes of Sugiyama, et al., Weldon, and Justesen. To produce the "unpunctured" Justesen codes we set $\ell = 1$ and take $\{0,1\}$ for the base code. In this section we present a new construction that produces codes at the Justesen bound. We recall that the Justesen bound $J(R)$ has the form

$$J(R) = \max_{1/2 \leq r \leq 1} \{H^{-1}(1-r)(1 - \frac{R}{r})\} ,$$

where $H(\cdot)$ is the binary entropy function.

The construction is done by concatenation. We utilize an extended RS code of rate R_{RS} and length $N = 2^k$ over $GF(2^k)$ for the outer code, and for the inner codes we use the Goppa Codes [10]. To begin we give a quick description of the particular kind of Goppa Code we need. Our interest lies in the binary irreducible Goppa Codes of degree t.

Let m be a fixed positive integer. A <u>binary irreducible Goppa Code of degree t</u> consists of 2^m-tuples $(c_\gamma: \gamma \in GF(2^m))$ indexed by $GF(2^m)$ which satisfy the relation

$$\sum_{\gamma \in GF(2^m)} \frac{c_\gamma}{z + \gamma} \equiv 0 \bmod g(z) ,$$

where $g(z)$ is a monic polynomial of degree t that is irreducible over $GF(2^m)$, and $c_\gamma \in \{0,1\}$. Our construction works because almost all Goppa Codes are Gilbert Codes. This property of Goppa Codes is well known (see, for example, MacWilliams and Sloane [11]). To establish the asymptotic

nature of our codes, we need a good estimate of the number of Goppa Codes that lie on the Gilbert bound. To produce this estimate, we will compute the number of Goppa Codes that are not Gilbert bound codes. This result may be found in Berlekamp [19] and is repeated for completeness.

Let us recall a few facts about Goppa Codes and irreducible polynomials that can be found in Berlekamp [12]. First of all, if we denote the number of irreducible polynomials of degree t over $GF(2^m)$ by I_t it is easy to establish that $I_t = \frac{2^{mt}}{t}(1 + o(1))$ as $t \to \infty$. This estimate is required in the determination of our code's minimum weight. We see also that I_t counts the total number of different binary irreducible Goppa Codes of degree t. It is not clear that distinct polynomials produce distinct Goppa Codes, but this is not crucial to the construction.

We also know that a binary irreducible Goppa Code of degree t has redundancy no larger than mt, and that any non-zero 2^m-tuple $\underline{x}$ can occur in at most $\lfloor \frac{wt(\underline{x})}{t} \rfloor$ different irreducible binary Goppa Codes of degree t.

Let us enumerate the Goppa Codes in question as $G_1, G_2, \ldots, G_{I_t}$. Furthermore, let us assume that if the minimum weight in the code G_i is d_i, then $d_i \leq d_j$ provide $1 \leq i \leq j \leq I_t$. We choose $t = \lfloor (1-r)2^m/m \rfloor$, where r is a fixed real number between zero and one. With this restriction it is easy to show that for $1 \leq i \leq I_t$ all of the codes G_i have rate at least r, and that as $m \to \infty$ the number I_t (the total number of Goppa Codes under discussion) satisfies

$$\log_2 I_t = (1-r)2^m(1 + o(1)) .$$

This last equation is a consequence of $I_t = \frac{2^{mt}}{t}(1 + o(1))$, $t \to \infty$.

To show that the concatenation we have in mind is asymptotically good, we need the following lemma.

Lemma. For every $0 < \varepsilon < 1\text{-}r$ and each integer $m \geq 2$ there exists a code G_M of length $n = 2^m$ such that as $n \to \infty$

$$\frac{d_M}{n} \geq (H^{-1}(1-r) - \varepsilon)(1 + o(1)) .$$

Furthermore, the number of Goppa Codes that do not satisfy this last inequality is $o(I_t)$ as $n \to \infty$.

Proof. Let $0 < \varepsilon < 1\text{-}r$ and $m \geq 2$ be given. Choose a number D so that:

(i) $D \leq \frac{1}{2} n$,

(ii) $\lfloor \frac{n}{t} \rfloor 2^{nH(D/n)} = 2^{-\varepsilon n} I_t$.

Table 1 shows the values of D and D/n for various values of r, ε, and m.

It is possible to make such a selection for D for the following reasons. The function $G(x) = \lfloor \frac{n}{t} \rfloor 2^{nH(x/n)}$ is an increasing function of x on the interval $[1, \frac{n}{2}]$ that assumes the value $\frac{e}{1-r}\, n \log_2 n(1 + o(1))$ at $x = 1$ and the value $\frac{1}{1-r}\, 2^n \log_2 n(1 + o(1))$ at $x = \frac{n}{2}$. Moreover, as $n \to \infty$, the right-hand side of (ii) is asymptotic to $2^{(1-r-\varepsilon+o(1))n}$. Thus, for large n, the right-hand side of (ii) falls between the extreme values of G(x) on the interval $[1, \frac{n}{2}]$. The existence of D is then guaranteed by the intermediate value theorem.

Specifically, it is easy to show that $G(\frac{n}{2})$ is greater than $2^{-\varepsilon n} I_t$ for all $n > 4$ and that $G(1)$ is less than $2^{-\varepsilon n} I_t$ for all n that satisfies the inequality $F(n) < 1 - r - \varepsilon$, where $F(n) = \frac{2 \log_2 n}{n} + H(\frac{1}{n})$. Straightforward calculations reveal that the function F(x), as a function of the real variable x, is strictly decreasing for $x \geq e$. Table 2 shows the least integral value n for which $F(n) < 1\text{-}r$ for various values of r and

Table 1

Values of D and D/n for Various Values of r, ϵ, and m

r = .25, ϵ = .1			r = .25, ϵ = .01			r = .25, ϵ = .0001		
m	D/n	D	m	D/n	D	m	D/n	D
5	.369	2.266	5	.458	3.090	5	.468	3.190
10	.632	1.628×10^{2}	10	.722	2.048×10^{2}	10	.732	2.100×10^{2}
25	.650	5.591×10^{6}	25	.740	7.020×10^{6}	25	.749	7.179×10^{6}
r = .5, ϵ = .1			r = .5, ϵ = .01			r = .5, ϵ = .0001		
m	D/n	D	m	D/n	D	m	D/n	D
5	.034	1.094	5	.054	1.742	5	.057	1.821
10	.076	7.784×10^{1}	10	.103	1.053×10^{2}	10	.106	1.086×10^{2}
25	.079	2.663×10^{6}	25	.107	3.581×10^{6}	25	.110	3.691×10^{6}
r = .75, ϵ = .1			r = .75, ϵ = .01			r = .75, ϵ = .0001		
m	D/n	D	m	D/n	D	m	D/n	D
7	.008	0.964	7	.022	2.852	7	.024	3.088
10	.019	1.913×10^{1}	10	.036	3.694×10^{1}	10	.038	3.912×10^{1}
25	.022	7.227×10^{5}	25	.040	1.325×10^{6}	25	.042	1.399×10^{6}

Table 2

The Smallest Integer n Satisfying $F(n) < 1-r$, where

$$F(x) = \frac{2 \log_2 x}{x} + H(\frac{1}{x})$$

r	n	r	n
0.00	13	.55	39
.05	14	.60	45
.10	15	.65	54
.15	16	.70	65
.20	18	.75	83
.25	19	.80	109
.30	21	.85	156
.35	23	.90	255
.40	26	.95	580
.45	30	1.00	∞
.50	34		

thus gives some indication as to where the above asymptotic inequalities are true. (Note that the upper inequality is always true for $n \geq 4$.)

Next consider all of the length n binary vectors $\underline{x}$ of weight less than or equal to D. We wish to bound from above the number of codes in the sequence $G_1, G_2, \ldots, G_{I_t}$ that contain a vector $\underline{x}$ of weight not greater than D. There are $\binom{n}{wt(\underline{x})}$ binary vectors of weight $wt(\underline{x})$, and each of these is in no more than $\lfloor \frac{wt(\underline{x})}{t} \rfloor$ of the codes in question. By observing that $\lfloor \frac{wt(\underline{x})}{t} \rfloor \leq \lfloor \frac{n}{t} \rfloor$, we may use the quantity $\sum_{1 \leq j \leq D} \lfloor \frac{n}{t} \rfloor \binom{n}{j}$ as an upper bound on the number of codes possessing a vector of weight D or less. Since our list of codes is ordered by minimum weight, only codes at the start of the list possess low weight codewords. Let us take

$$M = \sum_{1 \leq j \leq D} \lfloor \frac{n}{t} \rfloor \binom{n}{j} .$$

From the inequality

$$\sum_{1 \leq j \leq D} \binom{n}{j} \leq 2^{nH(D/n)}$$

and the choice of D we can conclude that

$$1 \leq \lfloor \frac{n}{t} \rfloor \binom{n}{1} \leq M \leq \lfloor \frac{n}{t} \rfloor 2^{nH(D/n)} = 2^{-\epsilon n} I_t < I_t .$$

Thus the code G_M exists and by our choice of M we must have $d_M \geq D$ (recall that d_i is the minimum weight of G_i). The ordering of the codes $G_1, G_2, \ldots, G_{I_t}$ and some easy calculations reveal that as $n \to \infty$

$$\frac{d_i}{n} \geq \frac{D}{n} = (H^{-1}(1-r) - \epsilon)(1 + o(1))$$

for all $i \geq M$. This establishes the first part of the lemma.

To prove the second and more important part of the lemma, we note that the number of non-Gilbert codes among $G_1, G_2, \ldots, G_{I_t}$ is no more than M. We may estimate M in light of the inequality

$$0 < \frac{M}{I_t} = \frac{\lfloor \frac{n}{t} \rfloor 2^{nH(D/n)}}{I_t} = 2^{-\epsilon n} .$$

Hence $M = o(I_t)$ as $n \to \infty$, and therefore the number of non-Gilbert codes among $G_1, G_2, \ldots, G_{I_t}$ is also $o(I_t)$. □

We are now ready to construct our codes and compute their asymptotic distance-to-length ratio.

Definition. The code J_G is a concatenated code whose outer code is an RS code of length $N = 2^k$ over the alphabet $GF(2^k)$ of rate R_{RS} and whose inner codes are the Goppa Codes $G_1, G_2, \ldots, G_{I_t}$ of length n and rate r discussed earlier.

The actual mechanics of the construction proceed as follows. Each outer RS codeword has the form $[\alpha_1, \alpha_2, \ldots, \alpha_N]$, where each α_i is representable as a k-bit binary column vector. If we insist that $k = rn$, then each α_i can be encoded by one of the codes G_{i*}, where we select

$$i* \equiv i \mod I_t$$

$$1 \leq i* \leq I_t .$$

Each codeword then has the form $[A_1\alpha_1, A_2\alpha_2, \ldots, A_{i*}\alpha_i, \ldots, A_{N*}\alpha_N]$, where A_{i*} is the generator matrix of the code G_{i*}. Note that the actual rate of G_{i*} is larger than r, so that A_{i*} may have more than k columns. Thus, to do the indicated encodings it may be necessary to pad each k-bit column vector α_i with an appropriate number of zeros.

The length η of the code J_G is clearly nN, and the information content is kK. As usual, the rate $\mathcal{R}$ of J_G is given by rR_{RS}. Our lemma on the minimum distance of binary irreducible Goppa Codes tells us that the number of codes whose distance-to-length ratio is less than $(H^{-1}(1-r)-\varepsilon)(1 + o(1))$ is $o(I_t)$ provided $0 < \varepsilon < 1-r$. Therefore, the fraction of codes that are "bad" is $o(1)$ as $n \to \infty$. If we use every code in the list $G_1, G_2, \ldots, G_{I_t}$ at least once, we will almost always be using Gilbert-bound codes. Thus we must require $N \geq I_t$. We show next that the inequality $N \geq I_t$ holds asymptotically if and only if $r \geq \frac{1}{2}$. Assuming $N \geq I_t$ for the moment, we compute a lower bound on the minimum weight of the code J_G.

We start by partitioning the N positions of a typical outer codeword into $p = \lfloor \frac{N}{I_t} \rfloor$ sets $S_1, S_2, \ldots, S_p$, where the set S_1 consists of the first I_t positions, S_2 the next I_t positions, and so forth. Each position of S_i is encoded by exactly one of the codes in the list $G_1, G_2, \ldots, G_{I_t}$. Let $\underline{c} = [\alpha_1, \alpha_2, \ldots, \alpha_N]$ be a typical non-zero codeword in the RS outer code, and let D_i be the number of α_j with $j \in S_i$ that are non-zero. As is the case in this kind of argument, we have no control over the size of the number D_i. However, not all of the D_i can be small since $\sum_{i=1}^{p} D_i$ must be at least the minimum weight D of the RS outer code minus the last few positions not considered by the sets $S_1, S_2, \ldots, S_p$. Specifically, we have

$$\sum_{i=1}^{p} D_i \geq D - (N-pI_t) \geq (1-R_{RS})N - I_t .$$

The fraction of the D_i non-zero positions in S_i that are encoded by non-Gilbert codes is $o(1)$. Therefore, the total Hamming weight of the binary encodings in each S_i is at least

$$0 \cdot D_i o(1) + n(H^{-1}(1-r)-\varepsilon)(1 + o(1))D_i(1 - o(1)) \ .$$

Summing over all of the sets S_i produces a lower bound on the minimum weight d of the code J_G, viz.,

$$d \geq \sum_{i=1}^{p} nD_i(H^{-1}(1-r)-\varepsilon)(1 + o(1)) \ .$$

Incorporating our estimate for $\sum_{i=1}^{p} D_i$ into the last inequality and rearranging produces

$$d \geq (H^{-1}(1-r)-\varepsilon)(1-R_{RS} - \frac{I_t}{N})nN(1 + o(1)) \ .$$

The length of our code is $\eta = nN$, so that the asymptotic distance-to-length ratio Δ is bounded below by the quantity

$$(H^{-1}(1-r)-\varepsilon)(1 - R_{RS} - \lim_{\eta \to \infty} \frac{I_t}{N}) \ .$$

We conclude by showing that $\lim_{\eta \to \infty} \frac{I_t}{N} = 0$ provided $r \geq \frac{1}{2}$. There are two cases to consider corresponding to $r > \frac{1}{2}$ and $r = \frac{1}{2}$. Our estimate of $\log_2 I_t$ shows that $\log_2 \frac{I_t}{N} = n(1-2r)(1 + o(1))$, so that $\frac{I_t}{N} \to 0$ as $\eta \to \infty$ provided $r > \frac{1}{2}$. If $r = \frac{1}{2}$ we may use our initial estimate of I_t to show that $\frac{I_t}{N} = \frac{2 \log_2 n}{n}(1 + o(1))$, so that $\frac{I_t}{N} \to 0$ as $\eta \to \infty$. In either case $\lim_{\eta \to \infty} \frac{I_t}{N}$ is zero. Recalling that ε is arbitrary and that $\mathcal{R} = rR_{RS}$, we may state the following lemma.

Lemma. The code J_G of rate $\mathcal{R}$ satisfies the following asymptotic bound:

$$\Delta = \liminf_{n \to \infty} \frac{d}{n} \geq \max_{\frac{1}{2} \leq r \leq 1} \{H^{-1}(1-r)(1 - \frac{R}{r})\} .$$

Notice that this bound is precisely the Justesen bound $J(R)$.

The remainder of this chapter is devoted to a discussion of the complexity of decoding the code J_G. If we use techniques similar to those suggested by Justesen [4] and Sarwate [13], we may decode the Goppa-based Justesen Codes in $0(n^2)$ bit operations. We simply decode each received inner word by searching the particular inner Goppa Code in question to find that codeword nearest the received inner word, and then we perform an overall correction by decoding the resulting outer word, using the fast decoding schemes of Justesen [14] or Sarwate [13]. By this method, we can correct no more than $\frac{1}{2} J(R)n$ errors -- the number guaranteed by the minimum distance. However, there is a fast decoding algorithm for Goppa Codes [15, 16]. If we utilize this fast algorithm, we can reduce our overall decoding complexity to $0(n \log^3 n)$ bit operations but we must sacrifice the asymptotic nature of our codes. As we will see, we can make the code J_G correct $0(\frac{n}{\log\log n})$ errors in $0(n \log^3 n)$ bit operations. This is better than comparable BCH Codes of length n, which in practically the same amount of time correct only $0(\frac{n}{\log n})$ errors. We present here a sketch of this procedure.

We begin by assuming that $\underline{r} = (r_\gamma : \gamma \in GF(2^m))$ is the received vector. From this we compute the syndrome polynomial $S(z)$. This is the polynomial having degree t-1 or less that satisfies

$$S(z) = \sum_{\gamma \in GF(2^m)} \frac{r_\gamma}{z + \gamma} \mod g(z) .$$

(Recall that g(z) is the irreducible polynomial of degree t that generates the given Goppa Code.) It is quite easy to show that the polynomial S*(z) defined by

$$S^*(z) = \sum_{\gamma \in GF(2^m)} r_\gamma \frac{g(z) + g(\gamma)}{z + \gamma} \cdot \frac{1}{g(\gamma)}$$

is congruent to S(z) mod g(z). Let us say that the set $E \subset GF(2^m)$ is the set of error locations. The error locator polynomial $\sigma(z)$ is therefore

$$\sigma(z) = \prod_{\gamma \in E} (z + \gamma) .$$

We can write the key equation for decoding Goppa Codes by noticing that

$$S(z)\sigma(z) \equiv \sigma'(z) \mod g(z)$$

where $\sigma'(z)$ is the formal derivative of $\sigma(z)$. There is no need to bring the error evaluator polynomial $\eta(z)$ into the picture as our Goppa Codes are binary. Thus the problem is to find a polynomial $\sigma(z)$ of degree t satisfying the key equation, given the polynomials S(z) and g(z). As is well-known, the work involved consists of (1) computing the syndrome polynomial S(z), (2) computing the polynomial $\sigma(z)$ from the key equation, and (3) determining the roots of $\sigma(z)$. Furthermore, Sarwate [15] shows that it requires (1) $O(n \log n) + O(t \log t)$ arithmetic operations to compute S(z), (2) $O(t \log^2 t)$ arithmetic operations to compute $\sigma(z)$ using the Euclidean Algorithm as suggested by Sugiyama, et al. [16] and as implemented by Moenck [17] (or as found in Figure 8.7 of Aho, Hopcroft, and Ullman [18]), and (3) $O(n \log n)$ arithmetic operations to compute the roots of $\sigma(z)$ via Fast Fourier Transforms [18]. The bit complexity of each of these steps can be found by multiplying the respective

arithmetic complexity by $\log^2 n$. Since $t = \lfloor (1-r) \frac{n}{\log_2 n} \rfloor$, we can easily show that our Goppa Codes can be decoded in $O(n \log n)$ arithmetic operations or $O(n \log^3 n)$ bit operations respectively. Thus to decode all N inner codes requires $O(nN \log^3 n)$ bit operations. The outer code is an RS code, which requires $O(N \log^4 N)$ bit operations to decode. Thus the total number of bit operations required for decoding is $O(nN \log^3 n) + O(N \log^4 N)$. From $\eta = nN = n2^k$ and $k = rn$ we have (for $n \geq 1$)

$$rn \leq \log_2 n + rn = \log_2 \eta \leq 2n ,$$

so that the decoding complexity can be written as

$$O(\eta \log^3 \log \eta) + O(\frac{\eta}{\log \eta} \log^4 \frac{\eta}{\log \eta}) .$$

This clearly simplifies to $O(\eta \log^3 \eta)$ bit operations.

The problem with this fast decoding scheme is this. To be able to correct an overall number of errors equal to half of the guaranteed minimum distance, we must be able to decode each inner codeword completely. That is, if the minimum distance in a given inner code is d we must be able to correct $\frac{1}{2}$ d or fewer errors when they occur and to declare an erasure otherwise. Our fast decoding algorithm, however, guarantees only that t/2 or fewer errors can be corrected in each inner word. (The quantity t/2 can be improved to t by noticing that a binary irreducible Goppa Code generated by g(z) is identical to the Goppa Code generated by $g^2(z)$ -- but this is a very slight improvement.) In any case, the number of errors that the inner decoder corrects is $O(\frac{n}{\log n})$. The outer decoder can tolerate O(N) errors, and so the overall code (using the fast inner

decoder) can correct $0(\frac{nN}{\log n})$ errors. This is $0(\frac{\eta}{\log\log\eta})$ errors in terms of the overall length η of the code.

As we have seen, the code J_G is no worse than the Justesen Code in terms of error tolerance and decoding complexity. If we increase the decoding speed via the fast algorithm for decoding Goppa Codes, we lose the asymptotic goodness of the code. Let us then compare the quickly decoded version of J_G to BCH Codes or comparable length and rate. A BCH Code of length n and rate r can correct about $\frac{n}{\log_2 n} \log_e(\frac{1}{r})$ errors. The quick version of J_G can correct about $\frac{1}{2}(1-r_i)(1-r_o) \frac{n}{\log_2\log_2 n}$ errors where r_i and r_o are the rates of the inner and outer codes respectively, and can be optimized by choosing $r_i = r_o = r^2$. In this case the error tolerance of the fast version J_G is about $\frac{1}{2}(1 - \sqrt{r})^2 \frac{n}{\log_2\log_2 n}$. Table 3 gives the value n for which the quick Goppa-based Justesen Code corrects more errors than the comparable BCH Code. By applying the algorithm for decoding Goppa Codes to BCH Codes (a BCH Code is after all a Goppa Code), we can show that it requires about 37 $n \log_2^3 n$ bit operations to decode a BCH Code. By appealing to the same algorithm, we can show that it takes about 32 $n \log_2^3 n$ bit operations to decode the fast version of the code J_G. Hence there is very little difference in the decoding time for either of the two types of codes. The difference between the constants 32 and 37 becomes even more insignificant in light of the need to take n larger than 10^6 in order to have the quick version of J_G correct more errors than the corresponding BCH Code (see Table 3).

What we have shown is this. The codes J_G and the Justesen Codes are similar in that both codes have the same error tolerance and decoding complexity. It is possible to decrease the decoding time of J_G by using a

Table 3

The Value of n at Which the Goppa-based Justesen Codes Decoded via the Fast Algorithm Tolerates More Errors than the Comparable BCH Codes. The quantity r is the code's rate and n is its length. The number n is defined by the equation

$$\frac{\log_2 \log_2 n}{\log_2 n} = - \frac{(1 - \sqrt{r})^2}{2 \log_e r} .$$

When r assumes a value of .284668, the quantity $\log_2 n$ takes on its minimum value of 21.850.

r	$\log_2 n$	r	$\log_2 n$
.05	33.684	.55	28.550
.10	27.040	.60	31.897
.15	24.109	.65	36.502
.20	22.634	.70	43.012
.25	21.971	.75	52.640
.30	21.873	.80	67.893
.35	22.239	.85	94.882
.40	23.039	.90	152.90
.45	24.303	.95	~346
.50	26.093		

fast decoding scheme for the Goppa Codes. When this is done, the codes J_G lose their ability to correct an asymptotically good number of errors. In this case, however, the quick version of J_G is still better than the long primitive BCH Codes in the following way. The fast version of J_G and the long primitive BCH Codes both take practically the same amount of time to decode, but the quickened codes J_G correct more errors than the comparable BCH Codes ($0(\frac{n}{\text{loglog } n})$ as compared to $0(\frac{n}{\log n})$) at lengths in excess of 10^6. Thus we have a trade-off. We can use a length n Goppa-based Justesen Code to correct $0(n)$ errors if we are willing to spend $0(n^2)$ time units to do it. On the other hand, by using the fast decoding procedure for the codes J_G, we can successfully decode a received word in $0(n \log^3 n)$ time units but only if the number of errors in the received word is not greater than $0(\frac{n}{\text{loglog } n})$.

BIBLIOGRAPHY

[1] E. N. Gilbert, "A Comparison of Signalling Alphabets," Bell System Tech. J., 1952, Vol. 31, pp. 504-522.

[2] R. R. Varsharmov, "Estimate of the Number of Signals in Error Correcting Codes," Dokl. Akad. Nauk. SSSR, 1957, Vol. 117, No. 5, pp. 739-741.

[3] V. V. Zyablov, "On Estimation of Complexity of Construction of Binary Linear Concatenated Codes," Probl. Peredach, Inform., 1971. Vol. 7, pp. 5-13.

[4] J. Justesen, "A Class of Constructive Asymptotically Good Algebraic Codes," IEEE Transactions on Information Theory, 1972, Vol. IT-18, No. 5, pp. 652-656.

[5] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "A Modification of the Constructive Asymptotically Good Codes of Justesen for Low Rates," Inform. Contr., 1974, Vol. 25, pp. 341-350.

[6] E. J. Weldon, Jr., "Justesen's Construction -- the Low Rate Case," IEEE Transactions on Information Theory, 1973, Vol. IT-19, No. 5, pp. 711-713.

[7] E. J. Weldon, Jr., "Some Results on the Problem of Constructing Asymptotically Good Error-Correcting Codes," IEEE Transactions on Information Theory, 1975, Vol. IT-21, pp. 412-417.

[8] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "A New Class of Asymptotically Good Codes Beyond the Zyablov Bound," IEEE Transactions on Information Theory, 1978, Vol. IT-24, pp. 198-204.

[9] J. M. Wozencraft and I. M. Jacobs. Principles of Communication Engineering, New York: Wiley, 1965.

[10] V. D. Goppa, "A New Class of Linear Error-Correcting Codes," Probl. Peredach. Inform., 1970, Vol. 6, pp. 24-30.

[11] F. J. MacWilliams and N. J. A. Sloane, The Theory of Error-Correcting Codes, Amsterdam: North-Holland, 1978.

[12] E. R. Berlekamp, Algebraic Coding Theory, New York: McGraw-Hill, 1968.

[13] D. V. Sarwate, "On the Complexity of Decoding Goppa Codes," IEEE Transactions on Information Theory, 1977, Vol. IT-23, pp. 515-516.

[14] J. Justesen, "On the Complexity of Decoding Reed-Solomon Codes," IEEE Transactions on Information Theory, 1976, Vol. IT-22, pp. 237-238.

[15] D. V. Sarwate, "An Asymptotically Efficient Decoding Algorithm for Goppa Codes," Proceedings of the 1976 IEEE Canadian Communications and Power Conference, 1976, Oct., pp. 213-215.

[16] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "A Method for Solving Key Equation for Decoding Goppa Codes," Inform. Contr., 1975, Vol. 27, pp. 87-99.

[17] R. Moenck, "Fast Computation of GCD's," Proceedings of the Fifth Annual ACM Symposium on the Theory of Computing, 1973, pp. 142-151.

[18] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, The Design and Analysis of Computer Algorithms, Reading, Mass.: Addison-Wesley, 1974.

[19] E. R. Berlekamp, "Goppa Codes," IEEE Transactions on Information Theory, 1973, Vol. IT-19, No. 5, pp. 590-592.

VITA

Robert Joseph Kleinhenz was born on November 6, 1949 in Long Beach, California. In 1967 he graduated from St. Ignatius High School in San Francisco. From there, he went to the University of Santa Clara in Santa Clara, California, where he received his bachelors degree in mathematics in 1971. In 1973, he was awarded a masters degree in mathematics from the University of Illinois at Urbana-Champaign.