operations. Procedure 8.7 in [1] recursively computes the last remainder whose degree exceeds $\frac{1}{2}$ deg r_{-1}. In the notation of [5], the output is r_{i-1}, r_i, V_{i-1}, U_{i-1}, and U_i, where r_{i-1} is the last remainder of degree greater than t. Thus deg $r_i \leq t$, and an additional iteration of Euclid's algorithm may be needed to make deg $r_k \leq t - 1$. If $2t$ is a power of 4, then the algorithm reduces the calculation of $\gcd(p_1,p_2)$ to two new greatest common divisor calculations with polynomials of degree at most $\frac{1}{2}$ deg p_1. The steps of this algorithm, as well as the additional iteration, require that a q_i and an r_i satisfying

$$r_{i-2} = q_i r_{i-1} + r_i$$

can be computed efficiently. This calculation has complexity $0(\deg r_{i-2})$ if deg q_i is upperbounded by a constant. However, if we need to divide polynomials of very different degrees, we first compute $r_{i-1}^{-1} = \lfloor x^{\deg r_{i-2}}/r_{i-1}(x) \rfloor$ by procedure 8.2 of [1], and then determine q_i as the leading coefficient of $r_{i-2}r_{i-1}$. The procedure for computing the approximate reciprocal of a polynomial of degree $k - 1$ involves the computation of a polynomial of degree $(k/2) - 1$ and multiplication of polynomials of degree k. Thus, when fast multiplications of the appropriate lengths are used, the procedure will recursively determine the reciprocal in $0(k \log k)$ operations. We conclude that the complexity of the decoding algorithm is $0(q \log^2 q)$ in terms of $GF(q)$ multiplications and additions.

In this correspondence we have indicated only the order of magnitude of the complexities, but inspection of the algorithms in [1] shows that the constant factors are small. Thus the use of the fast algorithm for evaluating polynomials is advantageous for codes of medium rate and length $2^8 + 1 = 257$. The modified algorithm for solving the key equation should be used when t^2 is large compared to $n \log^2 n$. For medium rate codes, this is the case for $n = 2^{16} + 1 = 65537$. The algorithm actually applies also to Goppa (and thus BCH) codes. However, the asymptotic results are most interesting for codes that remain good for large n.

References

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

[2] V. B. Afanasyev, "Time saving Reed–Solomon coding and error detection," in *Third Int. Symp. on Information Theory*, Tallin, 1973.

[3] R. C. Agarwal and C. S. Burrus, "Fast convolution using Fermat number transforms with applications to digital filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-22, pp. 87–97, Apr. 1974.

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

[5] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "A method for solving key equations for decoding Goppa codes," *Inform. Contr.*, vol. 27, pp. 87–99, Jan. 1975.

An Erasures-and-Errors Decoding Algorithm for Goppa Codes

YASUO SUGIYAMA, MASAO KASAHARA, MEMBER, IEEE, SHIGEICHI HIRASAWA, AND TOSHIHIKO NAMEKAWA, MEMBER, IEEE

***Abstract*—An erasures-and-errors decoding algorithm for Goppa codes is presented. Given the Goppa polynomial and the modified syndrome polynomial, a modified key equation is solved using Euclid's algorithm to determine the error locator polynomial and the errata evaluator polynomial.**

Manuscript received August 20, 1974; revised July 12, 1975.

Y. Sugiyama and S. Hirasawa are with the Communication Equipment Works, Mitsubishi Electric Corporation, Amagasaki, Japan.

M. Kasahara and T. Namekawa are with the Faculty of Engineering, Osaka University, Suita, Japan.

I. Introduction

It is well known that the receiver with an erasure option can improve the probability of decoding error. It is desirable to find as simple a decoding algorithm as possible for correcting erasures as well as errors. Such an algorithm enables one to carry out generalized minimum distance decoding [1] for further improving the probability of decoding error. Forney first formulated an erasures-and-errors decoding algorithm for Bose–Chaudhuri–Hocquenghem (BCH) codes [2]. Berlekamp formulated an elegant erasures-and-errors decoding algorithm for BCH codes based on his previous errors-only decoding algorithm [3].

Goppa has discovered a wide class of linear codes which includes BCH codes and Srivastava codes as subclasses [4]–[6]. Goppa has described an errors-only decoding algorithm for Goppa codes similar to Peterson's algorithm for BCH codes. Retter [7] has shown that Goppa codes can be decoded with a BCH decoder as given by Berlekamp. Patterson has formulated an errors-only decoding algorithm for Goppa codes in a manner similar to Berlekamp's algorithm for BCH codes [8]. We have formulated an errors-only decoding algorithm for Goppa codes using Euclid's algorithm [9]. In this correspondence, we present an erasures-and-errors decoding algorithm that includes our errors-only algorithm as a special case.

In Section II, Goppa codes and Euclid's algorithm are briefly described. In Section III, the erasures-and-errors decoding algorithm for Goppa codes is described together with its correction capability. In Section IV, a method for solving the key equation for the erasures-and-errors decoding is described.

II. Preliminaries

The q-ary Goppa code of length n with Goppa polynomial $g(z)$ is defined as the set of vectors $(a_1,a_2,a_3,\cdots,a_n)$ that satisfy

$$\sum_{i=1}^{n} \frac{a_i}{z - \alpha_i} \equiv 0 \bmod g(z),$$

where q is a prime power, $g(z)$ is a polynomial over $GF(q^m)$ of degree $2t$, α_i is an element of $GF(q^m)$ excluding roots of $g(z)$, and m and t are positive integers. All α_i are distinct. The minimum distance of the code is at least $(2t + 1)$.

In the following we summarize the properties of Euclid's algorithm as described in [9]. Euclid's algorithm computes the greatest common divisor of two polynomials $r_{-1}(z)$ and $r_0(z)$ where we assume deg r_{-1} > deg r_0. Given polynomials $r_{i-2}(z)$ and $r_{i-1}(z)$ such that deg r_{i-2} > deg r_{i-1}, it determines quotient polynomials $q_i(z)$ and remainder polynomials $r_i(z)$ which satisfy the following relations

$$r_{i-2}(z) = q_i(z)r_{i-1}(z) + r_i(z)$$

and

$$\deg r_{i-1} > \deg r_i \tag{1}$$

iteratively until it finds $r_i(z) = 0$. We define the following polynomials

$$U_i(z) = q_i(z)U_{i-1}(z) + U_{i-2}(z)$$

and

$$V_i(z) = q_i(z)V_{i-1}(z) + V_{i-2}(z)$$

where $U_0(z) = 1$, $U_{-1}(z) = 0$, $V_0(z) = 0$, and $V_{-1}(z) = 1$. Then we have the following relations

$$r_i(z) = (-1)^i\{-V_i(z)r_{-1}(z) + U_i(z)r_0(z)\}, \tag{2}$$

and

$$r_i(z) \equiv (-1)^i U_i(z) r_0(z) \bmod r_{-1}(z), \tag{3}$$

$$\deg U_i = \deg r_{-1} - \deg r_{i-1}, \tag{4}$$

$$U_i(z)V_{i-1}(z) - U_{i-1}(z)V_i(z) = (-1)^i. \tag{5}$$

III. Erasures-and-Errors Decoding

We assume that the demodulator can inform the decoder of erasures in addition to the usual elements from $GF(q)$ as its outputs. If the ith output of the demodulator is an element from $GF(q)$, the decoder assigns it to the ith component r_i of the received word. If the ith output is an erasure, the decoder assigns a predetermined value to r_i. When the ith output is not an erasure, the value $e_i = r_i - a_i$, $e_i \neq 0$, is the error value of the error that occured on the channel. When the ith output is an erasure, the value $e_i = r_i - a_i$ is the difference between the predetermined value and the transmitted value a_i and is called the erasure value. Without loss of generality, we adopt the element 0 in $GF(q)$ as the predetermined value. In this case the erasure value e_i is simply the additive inverse of the transmitted value a_i.

Let $\mathfrak{E}$ be the set of integers i such that the ith output has an error. Let ε be the set of integers i such that the ith output is an erasure. Then the syndrome polynomial $S(z)$, the error locator polynomial $\sigma_e(z)$, the error evaluator polynomial $\eta_e(z)$, the erasure locator polynomial $\sigma_\varepsilon(z)$, and the erasure evaluator polynomial $\eta_\varepsilon(z)$ are given by

$$S(z) = -\sum_{i=1}^{n} r_i \frac{g(z) - g(\alpha_i)}{z - \alpha_i} [g(\alpha_i)]^{-1},$$

$$\sigma_e(z) = \prod_{i \in \mathfrak{E}} (z - \alpha_i),$$

$$\eta_e(z) = \sum_{i \in \mathfrak{E}} e_i \prod_{\substack{j \in \mathfrak{E} \\ j \neq i}} (z - \alpha_j),$$

$$\sigma_\varepsilon(z) = \prod_{i \in \varepsilon} (z - \alpha_i),$$

and

$$\eta_\varepsilon(z) = \sum_{i \in \varepsilon} e_i \prod_{\substack{j \in \varepsilon \\ j \neq i}} (z - \alpha_j).$$

We have the following important relation

$$S(z) \equiv \frac{\eta_e(z)}{\sigma_e(z)} + \frac{\eta_\varepsilon(z)}{\sigma_\varepsilon(z)} \bmod g(z). \tag{6}$$

In (6), $\sigma_e(z)$ and $\eta_e(z)$ are relatively prime. However, $\sigma_\varepsilon(z)$ and $\eta_\varepsilon(z)$ are not necessarily relatively prime since some erasure values e_i may be zeros. We notice that the decoder knows not only $S(z)$ but also $\sigma_\varepsilon(z)$.

We define n_e to be the number of errors which is equivalent to the number of elements of the set $\mathfrak{E}$ and is equal to $\deg \sigma_e$. When $n_e = 0$, $\sigma_e(z)$ and $\eta_e(z)$ are defined as one and zero, respectively. Similarly, n_ε is defined to be the number of erasures.

Theorem 1: If $2n_e + n_\varepsilon < 2t + 1$, then there exists a unique set of three polynomials $\{\sigma_e(z),\eta_e(z),\eta_\varepsilon(z)\}$ that satisfies the congruence (6) for given $S(z)$ and $\sigma_\varepsilon(z)$.

Proof: We assume that the two sets $\{\sigma_e^{(1)}(z),\eta_e^{(1)}(z), \eta_\varepsilon^{(1)}(z)\}$ and $\{\sigma_e^{(2)}(z),\eta_e^{(2)}(z),\eta_\varepsilon^{(2)}(z)\}$ satisfy the congruence (6) for the same $S(z)$ and $\sigma_\varepsilon(z)$. Suppose that $2n_e^{(1)} + n_\varepsilon < 2t + 1$ and $2n_e^{(2)} + n_\varepsilon < 2t + 1$, then from (6) we can derive the relation

$$(\eta_e^{(1)}(z)\sigma_e^{(2)}(z) - \eta_e^{(2)}(z)\sigma_e^{(1)}(z))\sigma_\varepsilon(z) + (\eta_\varepsilon^{(1)}(z) - \eta_\varepsilon^{(2)}(z))\sigma_e^{(1)}(z)\sigma_e^{(2)}(z) \equiv 0 \bmod g(z). \tag{7}$$

Since the degree of the polynomial on the left side of (7) is less than $2t$, the left side of the congruence (7) must be zero. Since $\sigma_\varepsilon(z)$ is relatively prime to $\sigma_e^{(1)}(z)$ and $\sigma_e^{(2)}(z)$, $\sigma_\varepsilon(z)$ must divide $(\eta_\varepsilon^{(1)}(z) - \eta_\varepsilon^{(2)}(z))$. However, as $\deg \sigma_\varepsilon > \deg (\eta_\varepsilon^{(1)} - \eta_\varepsilon^{(2)})$, we have the relations $\eta_\varepsilon^{(1)}(z) = \eta_\varepsilon^{(2)}(z)$ and $\eta_e^{(1)}(z) \cdot \sigma_e^{(2)}(z) = \eta_e^{(2)}(z)\sigma_e^{(1)}(z)$. The latter relation implies that $\eta_e^{(1)}(z) = \eta_e^{(2)}(z)$ and $\sigma_e^{(1)}(z) = \sigma_e^{(2)}(z)$. Therefore, there exists a unique set of polynomials $\{\sigma_e(z),\eta_e(z),\eta_\varepsilon(z)\}$. Q.E.D.

Theorem 1 implies that we can correctly decode the received word with n_e errors and n_ε erasures by solving the congruence (6), provided that $2n_e + n_\varepsilon < 2t + 1$.

In the following discussions, we assume $1 \leq 2n_e + n_\varepsilon < 2t + 1$. The congruence (6) can be rewritten as follows:

$$S(z) \equiv \frac{\eta(z)}{\sigma_e(z)\sigma_\varepsilon(z)} \bmod g(z), \tag{8}$$

where $\eta(z) = \eta_e(z)\sigma_\varepsilon(z) + \eta_\varepsilon(z)\sigma_e(z)$. Defining $\sigma(z) = \sigma_e(z)\sigma_\varepsilon(z)$, we call $\sigma(z)$ the errata locator polynomial and $\eta(z)$ the errata evaluator polynomial. We call the congruence (8) the key equation for erasures-and-errors decoding of Goppa codes.

As described in Section IV, we can obtain $\sigma_e(z)$ and $\eta(z)$, for given $S(z)$ and $\sigma_\varepsilon(z)$ by solving the key equation (8). Once $\sigma_e(z)$ becomes known, the error locations i, $i \in \mathfrak{E}$, are found by determining the roots α_i, $i \in \mathfrak{E}$, of $\sigma_e(z)$ by the Chien search. When the error locations become known, the error values e_i, $i \in \mathfrak{E}$, are given by $\eta(\alpha_i)/\sigma'(\alpha_i)$, where $\sigma'(z)$ is the formal derivative of the errata locator polynomial $\sigma(z)$. The erasure values e_i, $i \in \varepsilon$, are also given by $\eta(\alpha_i)/\sigma'(\alpha_i)$.

IV. Solving Key Equation

In this section, we describe a method for solving the key equation for erasures-and-errors decoding of Goppa codes. We again assume that $1 \leq 2n_e + n_\varepsilon < 2t + 1$. The key equation (8) can be rewritten as follows,

$$\sigma_e(z)\sigma_\varepsilon(z)S(z) \equiv \eta(z) \bmod g(z), \tag{9}$$

since $\sigma_e(z)$ is relatively prime to $g(z)$ and so is $\sigma_\varepsilon(z)$. Since the decoder knows $S(z)$ and $\sigma_\varepsilon(z)$, the following polynomial becomes known to the decoder,

$$S_\varepsilon(z) \equiv \sigma_\varepsilon(z)S(z) \bmod g(z), \tag{10}$$

where $\deg S_\varepsilon$ is less than $2t$. We call $S_\varepsilon(z)$ the modified syndrome polynomial. Substituting (10) in (9), we obtain

$$\sigma_e(z)S_\varepsilon(z) \equiv \eta(z) \bmod g(z). \tag{11}$$

The congruence (11) exhibits the following four properties.

Property 1: $\deg \sigma_e = n_e \leq t - n_\varepsilon/2$.

Property 2: $\deg \eta \leq n_e + n_\varepsilon - 1 \leq t - 1 + n_\varepsilon/2$.

Property 3: The polynomial $\sigma_e(z)$ is monic.

Property 4: The polynomials $\sigma_e(z)$ and $\eta(z)$ are relatively prime.

From these properties, we have the following lemma.

Lemma 1: If $2n_e + n_\varepsilon < 2t + 1$, then there exists a unique pair of polynomials $\{\sigma_e(z),\eta(z)\}$ satisfying the congruence (11) for a given $S_\varepsilon(z)$.

This lemma follows from Properties 1–4 in a manner similar to the proof of Theorem 1, therefore, the details are omitted.

Lemma 1 states that for solving the key equation (8) it is sufficient to solve the congruence (11). We call the congruence (11) the *modified key equation.*

We will describe the properties of the modified syndrome polynomial $S_\varepsilon(z)$, which the decoder knows, in the following Lemmas 2, 3, and 4.

Lemma 2: The number of errors n_e is equal to zero if and only if the degree of the modified syndrome polynomial $S_\varepsilon(z)$ is less than the number of erasures n_ε.

Proof: When $n_e = 0$, it is clear that $S_\varepsilon(z) = \eta_\varepsilon(z)$, which implies that $\deg S_\varepsilon < n_\varepsilon$. Conversely, when $\deg S_\varepsilon < n_\varepsilon$, the degrees of the polynomials on both sides of the congruence (11) are less than $2t$, which implies that

$$\sigma_e(z)S_\varepsilon(z) = \eta(z). \tag{12}$$

Since the polynomials $\sigma_e(z)$ and $\eta(z)$ are relatively prime, (12) implies that $\eta_e(z) = 0$ and $\sigma_e(z) = 1$, i.e., $n_e = 0$. Q.E.D.

Lemma 3: The number of errors n_e is not equal to zero if and only if the degree of the modified syndrome polynomial $S_\varepsilon(z)$ is greater than or equal to $t + n_\varepsilon/2$.

Proof: When $n_e \neq 0$, we assume $\deg S_\varepsilon < t + n_\varepsilon/2$. Then (11) reduces to (12), which is contradictory to the relation $n_e \neq 0$. Conversely, when $\deg S_\varepsilon \geq t + n_\varepsilon/2$, we assume $n_e = 0$. Then $S_\varepsilon(z) = \eta_\varepsilon(z)$, which is contradictory to the relation $\deg S_\varepsilon \geq t + n_\varepsilon/2 \geq n_\varepsilon > \deg \eta_\varepsilon$. Q.E.D.

Lemmas 2 and 3 are not contradictory since the relation $n_\varepsilon \leq t + n_\varepsilon/2$ holds based on the assumption $1 \leq 2n_e + n_\varepsilon < 2t + 1$. We can decide whether n_e equals zero or not from $\deg S_\varepsilon$. If $n_e = 0$, the solution of the modified key equation can be directly obtained, i.e., $\sigma_e(z) = 1$ and $\eta(z) = S_\varepsilon(z)$. Therefore, we assume $n_e \neq 0$ in the following discussions.

Lemma 4: Let $p(z)$ be the greatest common divisor of the Goppa polynomial $g(z)$ and the modified syndrome polynomial $S_\varepsilon(z)$. Then $p(z)$ divides $\eta(z)$ and $\deg p \leq \deg \eta \leq t - 1 + n_\varepsilon/2$.

From the congruence (11), we can easily derive the relation, therefore, the details of the proof are omitted.

In the following we discuss solving the modified key equation (11). We summarize the problem as follows.

Problem: Given the Goppa polynomial $g(z)$ of degree $2t$ and the modified syndrome polynomial $S_\varepsilon(z)$ of degree $< 2t$, find a pair of relatively prime polynomials $\sigma_s(z)$ and $\eta_s(z)$ of degrees at most $t - n_\varepsilon/2$ and at most $t - 1 + n_\varepsilon/2$ which satisfy

$$\sigma_s(z)S_\varepsilon(z) \equiv \eta_s(z) \bmod g(z), \tag{13}$$

where $\sigma_s(z)$ is monic.

Lemma 5: Find a pair of polynomials $\{\sigma_0(z),\eta_0(z)\}$ such that they satisfy the relations $\sigma_0(z)S_\varepsilon(z) \equiv \eta_0(z) \bmod g(z)$, $\deg \sigma_0 \leq t - n_\varepsilon/2$, and $\deg \eta_0 \leq t - 1 + n_\varepsilon/2$ but are not necessarily relatively prime and $\sigma_0(z)$ is not necessarily monic, then

$$\sigma_0(z) = \mu(z)\sigma_e(z)$$

and

$$\eta_0(z) = \mu(z)\eta(z), \tag{14}$$

for some $\mu(z)$.

Since the proof of Lemma 5 can be done easily, it is omitted.

It is clear from Lemma 5 that there exists a unique solution of the above problem. The solution $\sigma_s(z)$ and $\eta_s(z)$ of the problem are the error locator polynomial $\sigma_e(z)$ and the errata evaluator polynomial $\eta(z)$ which correspond to the given modified syndrome polynomial $S_\varepsilon(z)$.

Now we can solve the problem using Euclid's algorithm.

Theorem 2: Set $r_{-1}(z) = g(z)$ and $r_0(z) = S_\varepsilon(z)$. Start the divisions of Euclid's algorithm for computing the greatest common divisor of $r_{-1}(z)$ and $r_0(z)$. Stop the divisions when the degrees of the remainder polynomials $r_i(z)$, $i = 0,1,2,\cdots$, satisfy $\deg r_{k-1} \geq t + n_\varepsilon/2$ and $\deg r_k \leq t - 1 + n_\varepsilon/2$, for some k. Then the solution of the above problem is given by

$$\eta_s(z) = (-1)^k \delta r_k(z)$$
$$\sigma_s(z) = \delta U_k(z), \tag{15}$$

where δ is a nonzero constant which makes $\sigma_s(z)$ monic.

Proof: The proof can be done according to the following steps.

i) From (1), Lemma 3, and Lemma 4, we see that the number of iterations k is unique.

ii) From (3), it is clear that the polynomials given by (15) satisfy (13).

iii) It is clear that $\deg \eta_s \leq t - 1 + n_\varepsilon/2$ is satisfied.

iv) From (4), it is clear that $\deg \sigma_s \leq t - n_\varepsilon/2$ is satisfied.

v) From (2), (5), and Lemma 5, we can see that the polynomials $\sigma_s(z)$ and $\eta_s(z)$ given by (15) are relatively prime.

vi) The factor δ makes $\sigma_s(z)$ monic. Q.E.D.

Corollary: The number of iterations k satisfies the relation $k \leq n_e \leq t - n_\varepsilon/2$.

This corollary is almost obvious, therefore, the proof is omitted. The corollary implies that the maximum number of iterations decreases as the number of erasures increases.

V. Concluding Remarks

We have presented an algorithm for erasures-and-errors correction for Goppa codes based on Euclid's algorithm. The algorithm includes the errors-only decoding algorithm for Goppa codes presented earlier by the authors [9] as the special case where $n_\varepsilon = 0$.

Our algorithm can be modified to eliminate multiplicative inversion as well as the algorithms described in [9] and [10]. However, the computations of the error values and erasure values require the divisions of elements in $GF(q^m)$ for nonbinary codes.

We compare the complexity of the erasures-and-errors decoding algorithm with that of the errors-only decoding algorithm for Goppa codes presented by the authors. The erasures-and-errors decoding algorithm requires the additional computations of the erasure locator polynomial $\sigma_\varepsilon(z)$, the modified syndrome polynomial $S_\varepsilon(z)$, the errata locator polynomial $\sigma(z)$, and the erasure values e_i, $i \in \varepsilon$, as the number of erasures n_ε increases. However, we can see from the corollary that the maximum number of iterations required for solving the modified key equation decreases as the number of erasures increases. We can conclude that the erasures-and-errors decoding algorithm is only slightly more complicated than the errors-only decoding algorithm.

Acknowledgment

The authors would like to thank Prof. E. R. Berlekamp for his warm-hearted and timely suggestions to investigate the decoding problem for Goppa codes and Prof. T. Kasami for his helpful suggestions. The authors would like to thank the referees for their helpful suggestions and comments. Y. Sugiyama and S. Hirasawa are grateful to Dr. T. Kitsuregawa, Dr. J. Baba, Dr. S. Kobayashi, Mr. Y. Maeda, and Mr. K. Fujiwara of Mitsubishi Electric Corporation for their encouragement.

REFERENCES

[1] G. D. Forney, Jr., *Concatenated Codes*. Cambridge, Mass.: M.I.T. Press, 1966, pp. 35–62.
[2] ——, "On decoding BCH codes," *IEEE Trans. Inform. Theory*, vol. IT-11, pp. 549–557, Oct. 1965.
[3] E. R. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968, pp. 229–231.
[4] V. D. Goppa, "A new class of linear error correcting codes," *Probl. Peredach. Inform.*, vol. 6, pp. 24–30, Sept. 1970.
[5] ——, "Rational representation of codes and (L,g) codes," *Probl. Peredach. Inform.*, vol. 7, pp. 41–49, Sept. 1971.
[6] E. R. Berlekamp, "Goppa codes," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 590–592, Sept. 1973.
[7] C. T. Retter, "Decoding Goppa codes with a BCH decoder," *IEEE Trans. Inform. Theory* (Corresp.), vol. IT-21, p. 112, Jan. 1975.
[8] N. J. Patterson, "The algebraic decoding of Goppa codes," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 203–207, Mar. 1975.
[9] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "A method for solving key equation for decoding Goppa codes," *Inform. Contr.*, vol. 27, pp. 87–99, Jan. 1975.
[10] H. O. Burton, "Inversionless decoding of binary BCH codes," *IEEE Trans. Inform. Theory*, vol. IT-17, pp. 464–466, July 1971.

Binary Codes with Improved Minimum Weights

WILLIAM O. ALLTOP

Abstract—**A recent table of Helgert and Stinaff gives bounds for $d_{max}(n,k)$, the maximum minimum distance over all binary linear (n,k) error-correcting codes, $1 \le k \le n \le 127$. Twelve new codes are constructed which improve lower bounds in the table. Two methods are employed: the algebraic puncturing technique of Solomon and Stiffler and generation by combinatorial incidence matrices.**

I. INTRODUCTION

A binary linear (n,k) error-correcting code is a k-dimensional subspace C of n-dimensional vector space V_n over $GF(2)$. Each codeword in C is considered as an n-tuple of 0's and 1's, with the weight of the codeword being the number of 1's in the corresponding n-tuple. The distance between two codewords is the number of coordinates in which they differ. Equivalently, the distance between codewords x and y is the weight of the difference $x - y$. The minimum distance for a code C is the minimum over the distances between pairs of distinct codewords in C. Since the difference $x - y$ is in a linear code C for x and y in C, the minimum distance of a linear code is just the minimum weight over the nonzero members of the code.

Fixing n and k, let $d_{max}(n,k)$ denote the maximum over all binary linear (n,k) error-correcting codes C of the minimum distance d_{min} of C. For $1 \le k \le n \le 127$, the table of Helgert and Stinaff [1] gives a lower and an upper bound for $d_{max}(n,k)$. Here we shall construct twelve codes with minimum distance larger than the corresponding lower bound in [1]. Four codes are constructed by the algebraic puncturing technique of Solomon and Stiffler [3]. A modification of that technique is used to construct three more codes. Combinatorial incidence matrices are used to define five codes which improve the lower bounds in [1]. Other lower bounds can be improved by successive removal of single columns from some of the new codes.

II. ALGEBRAIC PUNCTURING

The algebraically punctured codes of Solomon and Stiffler [3] are obtained by deleting columns from the q-ary maximal length shift register codes of length $q^k - 1$, dimension k, and minimum weight $q^k - q^{k-1}$. We shall give a brief description, without proofs, of this method for $q = 2$. Let C be a binary $(2^k - 1, k)$ code of minimum weight 2^{k-1} whose rows consist of the 0-vector together with all $2^k - 1$ shifts of a maximal linearly recurring sequence of degree k over $GF(2)$. Suppose $c_1,c_2,\cdots,c_k$ are columns which form a basis of the column space of C. Suppose further that $s_1,s_2,\cdots,s_f$ are integers satisfying $s_1 + s_2 + \cdots + s_f \le k$. For $1 \le i \le f$, let $t_i = s_1 + s_2 + \cdots + s_i$, and let the subspaces C_i of column vectors be defined by

$$\begin{aligned} C_1 &= \langle c_1,c_2,\cdots,c_{t_1}\rangle \\ C_2 &= \langle c_{t_1+1},c_{t_1+2},\cdots,c_{t_2}\rangle \\ &\vdots \\ C_f &= \langle c_{t_{f-1}+1},c_{t_{f-1}+2},\cdots,c_{t_f}\rangle. \end{aligned}$$

Now let $C_i^{\#}$ consist of the nonzero columns in C_i. Each row of $C_i^{\#}$ has length $2^{s_i} - 1$ and weight 0 or 2^{s_i-1}. Upon deleting the columns of $C_1^{\#} \cup C_2^{\#} \cup \cdots \cup C_f^{\#}$ from C, one obtains the algebraically punctured (n,k) code SS where

$$n = 2^k - 1 - \sum \{2^{s_i} - 1 : 1 \le i \le f\}$$

$$d_{min} = 2^{k-1} - \sum \{2^{s_i-1} : 1 \le i \le f\}.$$

As an example consider the (45,6) codes SS_1 of the first row in Table I. For this code $k = 6$, $s_1 = 4$, $s_2 = 2$. Therefore,

$$n = 2^6 - 1 - (2^4 - 1 + 2^2 - 1) = 45$$

$$d_{min} = 2^5 - (2^3 + 2) = 22.$$

TABLE I
ALGEBRAICALLY PUNCTURED CODES

Code	(s_j)	n	k	d_{min}	HS^-	HS^+
SS_1	(4,2)	45	6	22	21	22
SS_2	(5,2)	93	7	46	45	46
SS_3	(4,3)	105	7	52	51	52
SS_4	(4,2)	109	7	54	53	54
SS_5	(5,2,3)	87	7	42	40	42
SS_6	(5,3)	90	7	44	43	44
SS_7	(5,2,4)	81	7	38	37	40

The codes SS_j, $1 \le j \le 4$, of Table I result directly from the method of Solomon and Stiffler described above. To construct SS_5 one starts with the (93,7) code SS_2. Now select a three-dimensional column space C_3 from the original (127,7) code which meets the five-dimensional space C_1 in a one-dimensional subspace, and meets the two-dimensional space C_2 in the 0-vector. For example, suppose the columns $c_1,c_2,\cdots,c_7$ are linearly independent in the original (127,7) code. One can let $C_3 = \langle c_1, c_2 + c_6, c_3 + c_7\rangle$, with $C_1 = \langle c_1,c_2,c_3,c_4,c_5\rangle$ and $C_2 = \langle c_6,c_7\rangle$. Since the 0-vector is not a column in the code, $C_3^{\#}$ meets $C_1^{\#}$ in one column, and $C_3^{\#}$ is disjoint from $C_2^{\#}$. It follows that $C_3^{\#}$ contains six columns of the (93,7) code SS_2. Upon deleting these six columns, the (87,7) code SS_5 results, with $d_{min} = 46 - 4 = 42$. Similarly SS_6 results from deleting $C_1^{\#} \cup C_3^{\#}$ from the original (127,7) code. Since C_1 and C_3 intersect in two columns, one of which is the 0-column, $C_1^{\#} \cup C_3^{\#}$ contains 37 nonzero columns. Therefore, the resulting code SS_6 has length 90, and $d_{min} = 44$. The (81,7) code SS_7 can be constructed by puncturing SS_2. As in the construction of SS_5, let SS_2 be the result of removing $C_1^{\#} \cup C_2^{\#}$ from the original (127,7) code, where $C_1 = \langle c_1,\cdots,c_5\rangle$ and $C_2 = \langle c_6,c_7\rangle$. To

Manuscript received May 22, 1975.
The author is with the Naval Weapons Center, China Lake, CA 93555.