On the minimum distance of ternary cyclic codes

Citation for published version (APA):

DOI:
10.1109/18.212272

Document status and date:
Published: 01/01/1993

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 04. Oct. 2020
On the Minimum Distance of Ternary Cyclic Codes

Marijn van Eupen and Jacobus H. van Lint

Abstract—There are many ways to find lower bounds for the minimum distance of a cyclic code, based on investigation of the defining set. Some new theorems are derived. These and earlier techniques are applied to find lower bounds for the minimum distance of ternary cyclic codes. Furthermore, the exact minimum distance of ternary cyclic codes of length less than 40 is computed numerically. A table is given containing all ternary cyclic codes of length less than 40 and having a minimum distance exceeding the BCH bound. It seems that almost all lower bounds are equal to the minimum distance. Especially shifting, which is also done by computer, seems to be very powerful. For length equal to the minimum distance, mostly shifting, which is applied to find lower bounds for the minimum distance of ternary cyclic codes, seems to be very powerful. For length equal to the minimum distance, especially shifting, which is applied to find lower bounds for the minimum distance of ternary cyclic codes, seems to be very powerful.

Index Terms—Ternary cyclic code, minimum distance, shifting, selforthogonal, contraction.

I. INTRODUCTION

A CYCLIC CODE C of length n over the alphabet GF(q) (gcd(n, q) = 1) can be characterized as an ideal in the ring GF(q)[x]/(x^n - 1) with generator g(x) (say), which is a divisor of x^n - 1. A codeword of C will be written either as c(x) E GF(q)[x]/(x^n - 1) or as the vector c of length n having a as ith entry the coefficient of x^i in c(x). If a is a primitive rth root of unity in some extension field of GF(q), then all zeros of x^n - 1 can be written as a^j (0 <= j <= n - 1). If g(x) is not constant, then g(x) has some zero a^j since g(x) divides x^n - 1. But because g(x) is a polynomial over GF(q), it also has a^{k+i}, a^{2k+i}, ... as its zeros (we will say: m_{j,i}(x) divides g(x), where m_{j,i}(x) is the minimal polynomial of a^{j,i}, i.e., the monic polynomial that only has zeros a^{j,i}, a^{2j,i}, a^{3j,i}, ...).

So we can characterize g(x) (and also C) by the set G := \{m_{i,j}(x) \mid G(x) \mid (x^n - 1)\} (here we use that gcd(n, q) = 1, and so every zero of x^n - 1 (and so every zero of g(x)) has multiplicity one). Mostly we will characterize C by its defining set R := \{j \mid m_{i,j}(x) \mid G(x) \} (and sometimes by R, which is the set of integers modulo n that are not in R). We define the check polynomial h_1(x) as the reciprocal polynomial of h(x) := (x^n - 1)/g(x) (h_1(x) is the generator polynomial of the dual code). Of course every c(x) E C has zero a^j, j E R, and this is the same as saying: c has inner product zero with the vector (1, a^j, a^{2j}, ... a^{(n-1)j}). So a parity check matrix of C is

M(\{j_1, j_2, ..., j_r\}) :=

\begin{pmatrix}
1 & \alpha^{j_1} & \alpha^{2j_1} & \cdots & \alpha^{(n-1)j_1} \\
1 & \alpha^{j_2} & \alpha^{2j_2} & \cdots & \alpha^{(n-1)j_2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \alpha^{j_r} & \alpha^{2j_r} & \cdots & \alpha^{(n-1)j_r}
\end{pmatrix}

where R = \{j_1, j_2, ..., j_r\} is the defining set of C (we do not require that a parity check matrix has independent rows). For purposes that will become clear soon, we will define, for I \subset \{0, 1, ..., n - 1\}, M(R)_I as the submatrix of M(R) consisting of the ith columns of M(R), where i E I. One can see immediately that if R contains a consecutive set of length k, then M(R)_i contains a Vandermonde s x s submatrix and so rank(M(R)_i) = s, whenever \mid I \mid = s. So if we define for every codeword c E C its support \mid I \mid as the set of positions where e is nonzero, then a codeword c with support \mid I \mid <= s (i.e., the weight of c = \mid \{i \mid c_i \neq 0\} \mid is at most s) cannot occur since M(R)_I has full rank. So the minimum distance d = d(C) := \min_{c E C} \mid c \mid (c) is at least s + 1. We call this bound the BCH bound (cf. [3], [4]), and we will write d_{BCH} = s + 1, if s is the largest integer having the property that there is a code equivalent to C with a defining set containing a consecutive subset of size s (notice that R depends on the choice of a; we call two codes equivalent if their defining sets are the same up to multiplication with some integer coprime to n). The first generalization of the BCH bound was given by Hartmann and Tzeng [5]. They prove that if the consecutive sets I = \{j + a, j + a + 1, ..., j + a + b - 1\} (0 <= j <= s) are contained in R, and if (a, n) <= \delta, then the minimum distance d of the code with defining set R is at least \delta + s. Roos [6] generalized this by proving that if the statement is true for sufficiently many (say s') values of j, then the minimum distance is at least \delta + s' + 1. A last generalization can be found in [1] and is called the AB-method. It says that if A, B C \{0, 1, ..., n - 1\} are such that A + B := \{a + b \mod n \mid a \in A, b \in B\} is a subset of R, then the code with defining set R has minimum distance at least \delta, if rank(M(A)_j) + rank(M(B)_j) > \mid I \mid for every subset I of \{0, 1, ..., n - 1\} for which \mid I \mid <= \delta

If C_0 is a subcode of C, then we will write C_0 <= C. If R_0 is the defining set of C_0, then we must have R_0 <= R_0. Moreover we have: d(C) <= d(C_0). If C_0 <= C and C_0 != C, then we write C_0 < C (and we say: C_0 is a proper subcode of C). A cyclic code C is called minimal if C != \{0\} (0 denotes the zero word) and if \{0\} <= C <= C implies that C_0 = \{0\}. The generator of a minimal code is (x^n - 1)/m_j(x) for some j.

In Section II, we first give some theorems, that give good lower bounds for the minimum distance. The first one is called shifting and turned out to be very powerful. We used

Manuscript received July 8, 1991. This work was presented in part at Coding and Information Theory Conference, Essen, Germany, December 15–17, 1990.

M. van Eupen is with Eindhoven University of Technology, Eindhoven, The Netherlands, Hq 987.

J. H. van Lint is with Eindhoven University of Technology, Eindhoven, The Netherlands, Bg 335. He is also with Philips Research Laboratories, P.O. Box 80000, 5600 JA Eindhoven, The Netherlands.

IEEE Log Number 9203861.
a computer to compute the shifting bounds and for length less than 40 this bound was for approximately 95% of the codes equal to the minimum distance. Where the shifting bound did not equal the minimum distance we were often (but unfortunately not always) able to find a bound that did equal the minimum distance. Of course for many codes it is possible to find a lower bound more quickly by hand using other theorems than shifting (including the AB-method). But to avoid a long list of tedious examples, we only used other theorems when it was necessary (i.e., where the shifting bound did not equal the minimum distance). The results can be found in Table I.

In Section III, we treat the lengths 40 ≤ n ≤ 50. Because shifting is in the worst case exponential in n, the computer could in some cases not find the shifting bound in reasonable time. Fortunately other theorems were powerful enough to find almost all minimum distances. The results can be found in Table II and Table III.

II. LOWER BOUNDS

The technique of shifting was introduced in 1986 by Van Lint and Wilson [1]. It gives very good lower bounds for the minimum distance of cyclic codes. We now give a definition of the shifting bound d_{SHIFT}(C) of a cyclic code C, that is slightly different from the definition in [1], but easier to implement on a computer.

**Definition 1**: Suppose C is a cyclic code of length n with defining set R. We define the shifting bound d_{SHIFT}(C) inductively.

a) If C is a minimal cyclic code and r_{1}, r_{2}, ..., r_{w} is the longest sequence of different integers (mod n) such that
   1) \{r_{1}, r_{2}, ..., r_{w}\} \subseteq R,
   2) There is a sequence a_{1}, a_{2}, ..., a_{w} such that for all i: \{a_{i} + r_{1}, ..., a_{i} + r_{i-1}\} \subseteq R and a_{i} + r_{i} \notin R (all additions taken modulo n), then d_{SHIFT}(C) = \omega + 1.

b) If C is not a minimal code and

\[d_{SHIFT}(C) := \min_{C_{0} < C} d_{SHIFT}(C_{0}),\]

and r_{1}, r_{2}, ..., r_{w} is the longest sequence of different integers with \omega \leq d_{SHIFT}(C_{0}) - 1 satisfying conditions 1) and 2) in a), then d_{SHIFT}(C) = \omega + 1.

**Theorem 1 (Shifting)**: Suppose C is a cyclic code with minimum distance d. Then, d_{SHIFT}(C) ≤ d.

**Proof**: By induction w.r.t. the number of irreducible factors in the check polynomial.

1) Let C be a minimal code (i.e., there is one irreducible factor in the check polynomial) and r_{1}, r_{2}, ..., r_{w}(\omega > 0) a sequence satisfying conditions 1) and 2) in Definition a). Suppose c(x) ∈ C is of weight \omega and has support I. Then, we have for all 1 ≤ i ≤ \omega:

\[\text{rank}(M_{(r_{1}, r_{2}, ..., r_{i}))} = \text{rank}(M_{(a_{1} + r_{1}, a_{2} + r_{2}, ..., a_{i} + r_{i}))} + 1 \]

\[= \text{rank}(M_{(a_{1} + r_{1}, ..., a_{i} + r_{i-1}))} + 1 \]

The first equality follows from the fact that the rank of a matrix stays invariant under multiplication with \text{diag}(1, a^{\omega}, a^{2\omega}, ..., a^{(n-1)\omega}). To prove the second equality, we recall that

\[
M(a_{i} + r_{1}, ..., a_{i} + r_{i}) =
\begin{pmatrix}
1 & a^{a_{i} + r_{1}} & a^{2(a_{i} + r_{1})} & \cdots & a^{(n-1)(a_{i} + r_{1})} \\
1 & a^{a_{i} + r_{2}} & a^{2(a_{i} + r_{2})} & \cdots & a^{(n-1)(a_{i} + r_{2})} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & a^{a_{i} + r_{i}} & a^{2(a_{i} + r_{i})} & \cdots & a^{(n-1)(a_{i} + r_{i})}
\end{pmatrix}
\]

and see that c (the vector of length n corresponding to c(x)) is orthogonal to the first i - 1 rows of this matrix, but is not orthogonal to the last row (since otherwise c(x) would have \alpha^{\omega+r_{i}} as a zero, but \alpha_{i} + r_{i} \notin R and C is a minimal code, so this would imply that c(x) ≡ 0). So the last row is linearly independent from the other rows and the equality follows (notice that we may restrict ourselves to the columns corresponding to support I). The third equality is as obvious as the first one. So rank(M_{(r_{1}, r_{2}, ..., r_{i})}) = rank(M_{(r_{1}, r_{2}, ..., r_{i-1})}) + 1 = \cdots = rank(M_{(r_{i})}) + \omega - 1 = \omega. But this means that M_{(r_{1}, r_{2}, ..., r_{i})} has full rank and so c(x) cannot be a codeword. This implies that d_{SHIFT}(C) ≤ d.

2) Suppose C is not a minimal code. By induction, we know that d_{SHIFT}(C) ≤ d(C) := \min_{C_{0} < C} d(C_{0}). Suppose r_{1}, r_{2}, ..., r_{w} satisfy conditions 1) and 2) and \omega ≤ d_{SHIFT}(C_{0}) - 1 ≤ d(C) - 1. Again let c(x) be a codeword of weight \omega(> 0). We see that if \alpha^{a_{i}+r_{i}} is a zero of c(x), then c(x) is in a proper subcode of C, since a_{i} + r_{i} \notin R. This contradicts the fact that 0 < \omega ≤ d(C) - 1. So again the last row of M_{(a_{i} + r_{1}, a_{i} + r_{2}, ..., a_{i} + r_{i})} is linearly independent of the others and in the same way as in 1, we have rank(M_{(r_{1}, r_{2}, ..., r_{i})}) = \omega and c(x) cannot be a codeword.

In some cases shifting is not powerful enough and we have to improve the lower bound, using some other theorems. The first theorem is very useful and is also a consequence of a theorem of McEliece [7].

**Theorem 2**: Suppose C is a ternary cyclic code of length n with defining set R. Suppose that if i \notin R, then n - i \in R. Then, C is selforthogonal (or equivalently: wt(c(x)) ≡ 0 mod 3 for all c(x) ∈ C).

**Proof**: Suppose c(x), c'(x) ∈ C and \alpha is a primitive nth root of unity. If c(\alpha^{i}) \neq 0, then i \notin R so by assumption n - i \in R and c'(\alpha^{-n}) = 0. This is true for all i, so c(x)c'(x^{-1}) \equiv 0 mod (x^{n} - 1). Looking at the coefficient of x^{n-1} in c(x)c'(x^{-1}), we get \langle c(x), c'(x^{-1}) \rangle \equiv 0 mod 3 (\langle c(x), c'(x) \rangle denotes the inner product of c(x) and c'(x) in GF(3)). So C is selforthogonal (and because \langle c(x), c'(x) \rangle \equiv \langle c(x), c(x) \rangle + \langle c'(x), c'(x) \rangle + 2(c(x), c'(x), c(x) + c'(x)), this is equivalent to saying that all weights in C are divisible by 3).
because weights are divisible by 3, we have $d = 9$.

Example 22.3: $n = 22, R = \{0, 2, 6, 7, 8, 10, 11, 13, 17, 19, 20, 21\}$. Here, $d_{SHIFT} = 7$ and by Theorem 2 we have $d \geq 9$.

Example 23.1: $n = 23, R = \{0, 5, 7, 10, 11, 14, 15, 17, 19, 20, 22\}$. Here, $d_{SHIFT} = 7$ and by Theorem 2 we have $d \geq 9$.

Example 26.1: $n = 26, R = \{1, 2, 3, 6, 9, 18\}$. Here, $d_{SHIFT} = 13$ and by Theorem 2 we have $d \geq 15$.

Example 26.2: $n = 26, R = \{1, 2, 3, 4, 5, 6, 9, 10, 12, 15, 18\}$. Here, $d_{SHIFT} = 8$ and by Theorem 2 we have $d \geq 9$.

Example 26.3: $n = 26, R = \{1, 2, 3, 4, 6, 7, 9, 10, 11, 12, 18\}$. Here, $d_{SHIFT} = 8$ and by Theorem 2 we have $d \geq 9$.

Theorem 3 (Orthogonal Subcode Representation): Suppose $C$ and $C_0$ are ternary cyclic codes of length $n$ generated by $g(x)$ and $g_1(x)$ respectively, where $t > s$. Suppose that if $\gamma$ is a zero of $g(x)/q(x)$, then $\gamma^{-1}$ is a zero of $g_1(x)$. Then every $c(x) \in C$ can be written as $c(x) = c_0(x) + c_1(x)$, where $c_0(x) \in C_0$ and $c_1(x)$ is an element of the code $C_1$, that is generated by $g_1(x) = \prod_{j=1}^{s} m_j(x)$.

Moreover, $wt(c(x)) \equiv wt(c_0(x)) + wt(c_1(x)) \mod 3$.

Proof: It is easy to see that $gcd(g_0(x), g_1(x)) = g_1(x)$. So $g(x) = a(x)g_0(x) + b(x)g_1(x)$ for certain polynomials $a(x), b(x) \in GF(3)[x]/(x^n - 1)$. So also $c(x) = c_0(x) + c_1(x)$ for certain $c_0(x) \in C_0$ and $c_1(x) \in C_1$. Furthermore $g_0(x)g_1(x)$ is a zero of $g_0(x)$ if and only if $g_1(x)$ is a zero of $g_0(x)$.

Example 22.4: $n = 22, R = \{5, 7, 10, 11, 14, 15, 17, 19, 20, 21\}$. Here, $d_{SHIFT} = 6$. We wish to apply Theorem 3 to prove that $d \geq 7$. Let the code $C$ with defining set $R$ be generated by $g(x)$. Let $C_0$ be generated by $g_0(x)/(x^n - 1)$, then the condition in Theorem 3 is satisfied. $C_0$ is the code of Example 22.2 and is selforthogonal and has $d = 9$. Furthermore, $C_1$ is generated by $(x^n - 1)/(x^3 - 1)$ and $c_0(x) \in C_1$ can only have a weight divisible by 3 if $c_1(x) \equiv 0$. Now suppose $c(x) \in C$ has weight 6. Write $c(x) = c_0(x) + c_1(x)$. Since $c_1(x) \equiv 0$, then $c_0(x) \in C_0$ and $c_1(x) \in C_1$. Then by the congruence relation $wt(c(x)) \equiv wt(c_0(x)) \mod 3$, and from the observations previously made we have $c_1(x) \equiv 0$. So $c(x) \in C_1$, which is a contradiction, since $C_1$ has no codewords of weight 6. So $d \geq 7$.

Example 22.7: $n = 26, R = \{2, 6, 7, 8, 10, 11, 13, 17, 18, 19, 21\}$. Here, $d_{SHIFT} = 6$. Suppose the code $C_0$ is selforthogonal. Then the conditions in Corollary 1 are satisfied (see Example 23.1) and $d_{SHIFT} = 6$. Suppose the code $C_0$ is selforthogonal. Then the conditions in Corollary 1 are satisfied (see Example 23.1) and $d_{SHIFT} = 6$. Suppose the code $C_0$ is selforthogonal. Then the conditions in Corollary 1 are satisfied (see Example 23.1) and $d_{SHIFT} = 6$.
we can use knowledge of the cyclic codes of length \( n/2 \) to find better lower bounds.

**Definition 2:** Suppose \( C \) is a ternary cyclic code of length \( m \). Then the code of length \( 2m \) defined by:

\[
C^2 := \{ c(x^2) + xc'(x^2) | c(x), c'(x) \in C \}
\]

is called the square of \( C \).

It is easy to see that \( C^2 \) is a ternary cyclic code again, which has the same minimum distance as \( C \). Moreover the following holds.

**Theorem 4:** A ternary cyclic code \( D \) of length \( n = 2m \) is a square, if and only if \( R = m + R \), where \( R \) is the defining set of \( D \) and \( m + R := \{ m + r \mod n | r \in R \} \).

**Proof:** Suppose \( D = C^2 \), where \( C \) is a cyclic code of length \( m \). Suppose \( a \) is a primitive \( n \)th root of unity. Then obviously \( a^m = -1 \) and so \( a^{m+i} = -a^i \) for all \( i \). Let \( i \in R \). Because \( D \) is a cyclic code of \( C \), we know that \( a^{(2i)} - a^{(x^2)} = 0 \), for all \( c(x) \), \( c'(x) \in C \). But then also \( c(a^{2i}) - c'(a^{(x^2)}) = 0 \), for all \( c(x) \), \( c'(x) \in C \), since \( c'(x) \in C \) implies \( -c'(x) \in C \). So we also have \( c((-a)^x^2)^j + (-a)^i c((-a)^x^2)^j = 0 \), for all \( c(x) \), \( c'(x) \in C \), and so \( -a^i = a^{m+i} \) is a zero of all elements of \( D \). This means that \( m = n \) in \( R \). Again \( a \) is a primitive \( n \)th root of unity. We shall prove that if \( R = m + R \), then \( D = C^2 \), where \( C \) has defining set \( R' = R \mod m \). Take \( d(x) \in D \) and write \( d(x) + d(-x) = a(x^2) \), \( d(x) + d(x) = x^2 a(x^2) \), where \( a(x) \), \( b(x) \in GF(3)[x]/(x^m - 1) \). We have \( d((-a)^i) = d(-a^i) = 0 \) for all \( i \in R \). So also \( a^{(x^2)} = b(a^{(x^2)}) = 0 \) for all \( i \in R \). But this means that \( a^{(x^2)} = b(a^{(x^2)}) = 0 \) for all \( i \in R \), so \( a(x), b(x) \in C \) and so \( d(x) = 2((d(x) + d(-x)) + (d(x) - d(-x))) = 2((a(x^2) + x^2 a(x^2)) \in C^2 \). On the other hand, if \( c(a^{(x^2)}) = c'(a^{(x^2)}) = 0 \) for all \( i \in R \) (i.e., \( c(x), c'(x) \in C \), then \( c(a^{(x^2)}) + a^{(x^2)} c'(a^{(x^2)}) = 0 \) for all \( i \in R \). But then \( c(x^2) + c'(x^2) \) has zeros \( a^i \) and \( -a^i = a^{m+i} \) for all \( i \in R \). So \( c(x)^2 + c'(x)^2 \in D \). \( \square \)

Notice that the code of Example 22.3 is the square of the code of Example 11.1

**Example 22.8:** \( n = 22 \), \( R = \{ 2, 6, 7, 8, 10, 13, 17, 18, 19, 21 \} \). By Theorem 4, this code is the square of the code of Example 11.2. So \( d \geq 5 \). This also explains why the code has no words of weight 7 or 21.

In the explanation of Table III, the following theorem will be quite useful. We dedicate it to Paris, since it was proved (at night) in a hotel room in that city.

**Theorem 5 ("Paris by Night"):** Suppose \( C \) is a ternary cyclic code of length \( n = 2m \) and defining set \( R \). Let \( C_L \) be the cyclic code of length \( n \) with defining set \( R \cup (m + R) \) (the so called lower square of \( C \)) and \( C_U \) the cyclic code of length \( n \) with defining set \( R \cap (m + R) \) (the upper square of \( C \)). Call \( d_L \) and \( d_U \) the minimum distances of \( C_L \) and \( C_U \), respectively. Then, for the minimum distance \( d \) of \( C \), we have \( d \geq \min \{ d_L, 2d_U \} \).

**Proof:** Notice that \( C_L \) and \( C_U \) are ternary cyclic codes. Moreover, using Theorem 4, it is easy to see that \( C_L \) and \( C_U \) are squares. Of course \( C_L \leq C \leq C_U \). Suppose \( c(x) \in C \) and \( c(x) \notin C_L \). Then \( c(x) \) is nonzero both on the even positions and on the odd positions, since otherwise \( c(x) = 0 \) would imply \( c(a^{-i}) = c(a^{m+i}) = 0 \), and \( c(x) \) would be an element of \( C_U \). But \( c(x) \) is also in the square \( C_U \). So \( \text{wt}(c(x)) \geq 2d_U \).

**Corollary 2:** If \( d_L \leq 2d_U \), then \( d = d_U \).

**Proof:** From Theorem 5 we have: \( d \geq d_L \). But \( C_L \leq C \), so we also have: \( d \leq d_L \).

**Example 22.9:** \( n = 22 \), \( R = \{ 2, 4, 6, 7, 8, 10, 12, 13, 14, 16, 17, 18, 19, 20, 21 \} \). \( C_U \) is the code of Example 22.8 and so \( d_U \geq 5 \). \( C_L \) has defining set \( \{ 0, 11 \} \) and so \( d_L = 11 \). Using Theorem 5, we get: \( d \geq 10 \).

**Theorem 6:** Suppose \( C \) is a ternary cyclic code of length \( n = 2m \) with defining set \( R \). Let \( C_E \) be the cyclic code of length \( n \) with defining set \( R \cup \{ f \mid j \equiv 1 \mod 2 \} \) and \( C_O \) the cyclic code of length \( n \) with defining set \( R \cup \{ f \mid j \equiv 0 \mod 2 \} \). Then every element of \( C \) can be written as \( (a + b, -a + b) \), where \( (a, -a) \in C_E \), and \( (b, b) \in C_O \) (\( a \) and \( b \) are of length \( m \)), and

\[
\begin{align*}
\text{wt}((a+b, -a+b)) &= 3(m - l_{b0}) - \text{wt}(a) - \text{wt}(b), \quad \text{where} \quad l_{b0} := |\{i|a_i = b_i = 0\}|, \\
\text{wt}((a+b, -a+b)) &\geq 2 \max \{ \text{wt}(a), \text{wt}(b) \} - \min \{ \text{wt}(a), \text{wt}(b) \}, \\
\text{wt}((a+b, -a+b)) \leq \min \{ \text{d}_{E}, \text{d}_{O}, \text{d}(E/2, d_{O}/2) \}, \quad \text{where} \quad \text{d}_{E} \text{ and } \text{d}_{O} \text{ are the minimum distances of } C_E \text{ and } C_O, \text{ respectively, and } (a + b, -a + b) \neq 0.
\end{align*}
\]

**Proof:** By Theorem 3, we can write every codeword of \( C \) as the sum of a codeword of \( C_E \) and a codeword of \( C_O \). But every codeword of \( C_E \) is of the form \( (a, -a) \) and every codeword of \( C_O \) is of the form \( (b, b) \) (since \( x^m - 1 \) and \( x^m + 1 \) are divisors of their respective generators). Now, define for given \( a \) and \( b \) in \( GF(3)^m \) \( l_{rs} := |\{i|a_i = r \text{ and } b_i = s\}| \). Then, we have

\[
\begin{align*}
\text{wt}(a) + \text{wt}(b) + \text{wt}(a + b) + \text{wt}(-a + b) &= \sum_{r \neq 0} \sum_{s \neq 0} l_{rs} + \sum_{r} \sum_{s \neq 0} l_{rs} + \sum_{r} \sum_{s \neq -r} l_{rs} \\
&+ \sum_{r \neq 0} \sum_{s \neq 0} l_{rs} \\
&= \sum_{r \neq 0} \left( \sum_{s} l_{rs} + \sum_{s \neq 0} l_{rs} + \sum_{s \neq -r} l_{rs} \right) + 3 \sum_{r \neq 0} l_{0s} \\
&= \sum_{r \neq 0} \left( \sum_{s} l_{rs} + 2 \sum_{s} l_{rs} \right) + 3 \sum_{s \neq 0} l_{0s} \\
&= 3 \left( \sum_{r \neq 0} \sum_{s} l_{rs} + \sum_{s \neq 0} l_{0s} \right) \\
&= 3 \sum_{r, s} (l_{rs} - l_{00}) \\
&= 3(m - l_{00}).
\end{align*}
\]
And so
\[ wt((a + b, -a + b)) = wt(a + b) + wt(-a + b) = 3(m - l_0) \]
\[ - wt(a) \geq 3 \max\{wt(a), wt(b)\} \]
\[ - wt(a) - wt(b) \geq \max\{wt(a), wt(b)\}, \]
\[ - wt(a) - wt(b) = 2 \max\{wt(a), wt(b)\} \]
\[ - \min\{wt(a), wt(b)\} \geq \max\{wt(a), wt(b)\}. \]

So a) and b) have been proved. To prove c), we observe that there are three possibilities:
1) \( wt(a) = 0 \) then \( wt((a + b, -a + b)) = d_0 \).
2) \( wt(b) = 0 \) then \( wt((a + b, -a + b)) = d_0 \).
3) \( wt(a) \neq 0, wt(b) \neq 0 \) then \( max\{wt(a), wt(b)\} \geq max(d_0, d_0/2) \).

Example 22.10: \( n = 22, R = \{0, 1, 3, 5, 9, 15\} \). \( C_R \) is equivalent the code of Example 22.1 and so \( d_R \geq 12 \). \( C_0 \) has defining set \( \{0\} \) and so \( d_0 = 22 \). Applying Theorem 6c) we get \( d \geq 11 \). But a codeword of weight 11 cannot occur by Corollary 1 a). So \( d \geq 12 \).

Theorem 5 and Theorem 6 will be used more optimally in the next section. We shall now discuss the relationship between Theorem 6 and contraction. The method of contraction was introduced in [1]. For the parameters that we shall treat in this paper, we could actually do without the method. However, one special case is so often quite useful that we mention it here.

Lemma 1: Let \( C \) be a cyclic code of length \( 2m \) with defining set \( R \), containing as even integers the set \( 2R \) (where \( R' \in \{0, 1, \ldots, m - 1\} \)). Define the contraction \( C' \) as
\[ \{c_0 + c_m, c_1 + c_{m+1}, \ldots, c_{m-1} + c_{2m-1} \mid c \in C\} \]
and write \( c_i := c_i + c_{m+i}, (0 \leq i < m) \). Then \( C' \) is a cyclic code of length \( m \) with defining set \( R' \).

Proof: Let \( \alpha \) be a primitive \((2m)\)th root of unit and \( \beta = \alpha^2 \) a primitive \( m \)th root of unit. Then
\[ \sum_{i=0}^{m-1} c_i (\beta^j)^{i} = \sum_{i=0}^{m-1} (c_i + c_{m+i}) (\alpha^{2j})^{i} = \sum_{i=0}^{m-1} c_i (\alpha^{2j})^{i} + \sum_{i=m}^{2m-1} c_i (\alpha^{2j})^{i} = 0, \]
if \( j \in R' \) (and hence \( 2j \in R \)).

We use this as follows. If \( c' \in C' \) and \( c' \neq 0 \), then \( wt(c') \geq wt(c) \). If \( c' = 0 \), then \( wt(c) \) is even and furthermore 0 belongs to the defining set of \( C' \) (in other words \( c \) is in the even-like subcode of \( C \)).

Example: \( n = 26, R = \{2, 4, 6, 8, 10, 12, 13, 14, 16, 18, 20, 22, 24\} \). We see that \( R' = \{0\} \) (see Lemma 1). So if \( c \) is a codeword and \( c' \) its contraction, then either \( wt(c') = 13 \) (and so \( wt(e) \geq 13 \) or \( wt(e') = 0 \). So, if \( wt(e) < 13 \), then \( wt(e) \) is even. This is also obvious if we use Theorem 6 and observe that \( C_0 \) has defining set \( \{0\} \).

Suppose \( C' \) is a ternary cyclic code of length \( n = 2m \) and \( c := (a + b, -a + b) \), where \( (a, -a) \in C_{2m} \) and \( (b, b) \in C_0 \), is a code of \( C' \). Then obviously \( c' = 2b \) (see Lemma 1) and so \( wt(c') = wt(b) \). So actually \( d_0/2 \) equals the minimum distance of \( C' \). One could also define \( C''_0 = \{(c_0 - c_m - c_1 + c_{m+1}, \ldots, \pm c_{m-1} \pm c_{2m-1} \mid c \in C \} \) and derive something similar for \( d' \). But unfortunately \( C''_0 \) need not be cyclic.

We shall not go into detail about this. The reader can verify easily that in the ternary case Theorem 6 is always at least as powerful as contraction to length \( n/2 \). A ternary code of length \( n = 2m \) is called double if \( \{jj \equiv 1 \text{ mod } 2 \} \subset R \), and it is easy to see that the minimum distance of a double code is twice the minimum distance of its contraction.

Special Cases

Example 26.6: \( n = 26, R = \{1, 2, 3, 6, 8, 9, 17, 18, 20, 23, 24, 25\} \). \( d_{\text{SHIFT}} = 6 \). We wish to prove that \( d \geq 7 \). Suppose \( c(x) \) is a codeword of weight 6. Obviously we have
\[ (c(x) + c(-x))^2 - (c(x) - c(-x))^2 = 2(c(x)c(-x)). \]

Notice that \( c(x)c(-x) \) is in the code with defining set \( \{0, 13\} \). Moreover, \( gcd(c(x), x^2 - 1) = gcd(c(-x), x^2 - 1) = 1 \) for any \( c \), since the codes corresponding to \( R \cup \{1\} \) and \( R \cup \{-1\} \) have \( d \geq 8 \), and so \( c(x)c(-x) \) is equal to \( (x^{10} - 1)/(x^2 - 1) \), up to cyclic shift or multiplication by 2. We wish to prove that this is not possible for any \( w_0, w_1 \), where \( w_0 = \text{wt}(c(x) + c(-x)) \) and \( w_1 = \text{wt}(c(x) - c(-x)) = 6 - w_0 \). First, we have to make some preparations. We apply Theorem 3 to get \( c(x) = c_0(x) + c_1(x), \) where \( c_0(x) \) is in the code with defining set \( R \cup \{0, 13\} \) and \( c_1(x) \) is in the code with defining set \( \{0, 13\} \).

We do this because looking at the zeros of \( c_0(x) \) and \( c_1(x) \) we have \( c_0(x)q_0(x^-1) \equiv 0 \) and \( c_0(x)c_1(x^{-1}) \equiv 0 \), and so \( (c_0(x), c_0(-x)) = (c_0(x), c_1(-x)) = (c_0(x), c_1(x)) = 0 \).
So \( w_0 = (c(x) + c(-x), c(x) - c(-x)) = 2(c(x), c(-x)) = 2(c_0(x), c_1(-x)) + 2(c_1(x), c_1(x)) = 2(c_0(x), c_1(x)) + (c_1(x), c_1(x)) = 2(c_1(x), c_1(x)). \) But, again by Theorem 3, we can write \( c_1(x) = a_1l(x) + a_2l(-x) \), where \( l(x) = x^{25} + x^{24} + \ldots + x + 1 \equiv (x^{29} - 1)/(x - 1) a_1, a_2 \in GF(3) \).
Then, \( (l(x), l(-x)) = 0 \), we have that \( (c_1(x), c_1(-x)) = a_1a_2 \) and moreover \( c_1(1) = \{c_1(x), l((x)) = 2a_1 \) and \( c_1(-1) = \{c_1(x), l((-x)) = 2a_2 \).
So \( w_0 \equiv 2(c_1(x), c_1(x)) = 2(1)(c_1(-1)) \). We know that \( c_1(1) \not= 0 \) and \( c_1(-1) \not= 0 \). So we must be in one of the following cases.

1) \( c_1(1) + c_1(-1) = 0 \) (i.e., \( 1 \) is a zero of \( c(x) + c(-x) \)).
This means that \( c_1(1)(-1) = 2 \) or equivalently that \( w_0 \equiv 1 \text{ mod } 3 \) and \( w_1 \equiv 2 \text{ mod } 3 \).

2) \( c_1(1) - c_1(-1) = 0 \) (i.e., \( 1 \) is a zero of \( c(x) - c(-x) \)).
This means that \( c_1(1)(-1) = 1 \) or equivalently that \( w_0 \equiv 2 \text{ mod } 3 \) and \( w_1 \equiv 1 \text{ mod } 3 \).

So certainly \( w_0 \not= 0 \text{ mod } 3 \). If \( w_0 = 1 \), then \( 1 \) is not a zero of \( c(x) + c(-x) \) and so we are in case 2, which is a contradiction, since \( w_0 \not= 2 \text{ mod } 3 \).
So essentially there is just one case left, and that is the case where \( w_0 = 2 \) and both
nonzero coefficients in \( c(x) + c(-x) \) have the same sign (if they have different sign, then 1 is a zero of \( c(x) + c(-x) \) and \( w_0 \equiv 1 \) mod 3). In that case \( (c(x) + (c(-x))^2 \) has weight at most 3, and
\[
\text{wt}\left((c(x) - c(-x))^2\right) \leq 4 + \left(\frac{4}{2}\right) = 10.
\]

Looking at (1), we see that there cannot be any overlap between \( (c(x) + c(-x))^2 \) and \( (c(x) - c(-x))^2 \), since the weight of their sum must be 13. But also by (1), we must then have that all nonzero coefficients in \( (c(x) + c(-x))^2 \) have the same sign. This is a contradiction, since the nonzero coefficients in \( c(x) + c(-x) \) have the same sign. So \( c(x) \) cannot have weight 6 and so \( d \geq 7 \).

Example 26.7: \( n = 26, R = \{0, 13, 14, 16, 17, 22, 23, 25\} \). Here, \( d_{\text{SHIFT}} = 5 \) and the Roos bound equals 6 (which is rather remarkable). To prove this, observe that the consecutive sets \( \{13 + 3j, 14 + 3j\} \) are contained in \( R \) if \( j \in \{0, 1, 3, 4\} \).

By Roos [6], this means that \( d \geq 3 + 4 - 1 = 6 \).

Explanation of Table 1

By computer we calculated the minimum distance of all ternary cyclic codes with length \( < 40 \). In many cases this minimum distance is equal to the BCH bound. In Table 1 we give the minimum distance of all ternary cyclic codes of length \( < 40 \), and minimum distance not equal to the BCH bound. So the minimum distances of all other ternary cyclic codes of length \( < 40 \) can easily be found by computing the BCH bound.

Also by computer we calculated the \( d_{\text{SHIFT}} \) (see Definition 1) of the codes listed in Table 1. We see from Table 1 that in many cases the minimum distance equals the \( d_{\text{SHIFT}} \). In the other cases, some minimum distances are equal to bounds given by other theorems in this section (or belong to the special cases). In these cases we refer to an example in this section. Unfortunately some minimum distances are left, that could not be derived theoretically by us. These are indicated by a question mark. By the shifting certificate we mean the sequence \( r_1, r_2, \ldots, r_{d_{\text{SHIFT}} - 1} \), which has been put in the order of Definition 1.

Of course we did not mention codes that were equivalent to a code that was already in the list. We saw that we could call two codes equivalent, if their defining sets were the same up to multiplication with an integer coprime to \( n \). But if \( n = 2m \), we can also call two codes equivalent, if their defining sets are the same up to a shift over \( m \) (this corresponds to a substitution of \( x \) by \(-x\) in the codewords \( c(x) \), and so the minimum distance stays invariant). So we call two codes equivalent if their defining sets can be obtained from each other by some combination of multiplying with an integer coprime to \( n \) and shifting over \( m \) (where \( n = 2m \)).

To find the defining sets of the codes in Table 1 and to find equivalent codes, we refer to the Appendix.

III. LONGER CODES

In this section, ternary cyclic codes will be studied with \( 40 \leq n \leq 50 \). Computing exact minimum distances for these codes is very time consuming, so we only computed lower bounds. Of course, first of all we used shifting, since this method seems to be very powerful as we saw in Section II. But, for some codes, shifting also took too much time and so we had to compute a lower bound using one of the other theorems in Section II. Of course, we also tried to improve the \( d_{\text{SHIFT}} \) using one of the other theorems.

\( n = 40 \)

There are too many ternary cyclic codes of length 40 to give a list of all of them. So for each dimension \( k \) we computed the best of the lower bounds for the minimum distance of all codes of dimension \( k \), and only listed the corresponding codes (Table II). For some codes in Table II we were able to find an upper bound for the minimum distance (just by computing some codewords), that equaled the lower bound for its minimum distance. This is indicated by a boldface entry in the column "\( d_{\text{max}} \)." In the last column of Table II, an explanation is given for the lower bound. If this lower bound is not the BCH bound and is not explained in some example, then the bound equals the \( d_{\text{SHIFT}} \) and the sequence \( r_1, r_2, \ldots, r_{d_{\text{SHIFT}} - 1} \) is given.

Example 40.1: \( n = 40, R = \{1, 3, 9, 27\} \). We will use Theorem 5 to prove that \( d \geq 24 \). Notice that \( C_I \) has as defining set all integers modulo 40, and so we may take \( d_L \) infinite. \( C_I \) is the square of the code of length 20 with defining set \( \{0, 2, 4, 5, 6, 8, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19\} \), which has minimum distance at least 12 by the BCH bound. So by Theorem 5 we have \( d \geq 24 \).

Example 40.2: \( n = 40, R = \{0, 1, 3, 9, 27\} \). Again we will use "Paris by night" to prove that \( d \geq 22 \). Also again \( d_L \) is infinite. \( C_I \) is the square of the code of length 20 with defining set \( \{0, 1, 3, 7, 9\} \), which has minimum distance at least 11 by the BCH bound. So by Theorem 5, we have \( d \geq 22 \).

\( 40 < n \leq 50 \)

In Table III, we give a complete list of all ternary cyclic codes of length \( 40 < n \leq 50 \), having a lower bound for the minimum distance more than the BCH bound. Again we also computed upper bounds and if the lower bound equals the upper bound, we indicate this with a boldface entry in the column "\( d \geq \)." In the last column one can find an explanation for the lower bound for \( d \). If shifting gives the best bound, then a shifting certificate is given. Otherwise we either refer to a theorem ("Paris" means Theorem 5) or to an example. Sometimes we also give an explanation for the upper bound, by giving a subcode with a known minimum distance contained in our code (e.g., "\( \geq 13 \)"). The "\( \leq \)"-sign denotes "a code equivalent to." If the code is double or a square, then the minimum distance can be calculated from Table I and there is nothing else to explain. Notice that not for all codes the shifting bound is given in Table III. Here the computer spent too much time to compute the shifting bound. Because we need lower bounds for the minimum distance of all proper subcodes of a code to compute the shifting bound, we have put other lower bounds in the computer where shifting bound was not known. Notice that this means that the entries in the column
We shall use Theorem 5 to prove that $d$ need not equal the shift of Definition 1 (but it is

Example 44.1: $n = 44$, $R = \{2, 6, 10, 11, 18, 30, 33\}$. We shall use Theorem 5 to prove that $d \geq 22$. $C_d$ has defining set $\{11, 33\}$ and so $d_L = 22$. $C_U$ is the square of the code of length 22 with defining set $\{0, 1, 3, 4, 5, 7, 9, 12, 13, 14, 15, 16, 17, 19, 20, 21\}$, which has minimum distance 12, since it is equivalent to the code of Example 22.10. So also $d_U = 12$. By Theorem 5, we have $d \geq 22$. By Corollary 2, we have $d = 22$.

Example 44.2: $n = 44$, $R = \{0, 2, 6, 10, 11, 18, 30, 33\}$. We wish to apply Theorem 6 to prove that 21 is the best lower bound for the minimum distance that we can find (notice that "Paris by Night" gives $d \geq 20$). Suppose $c = (a + b, -a + b)$ is a nonzero element of the code of weight less than 21, where $(a, -a) \in C_E$ and $(b, b) \in C_D$. If $c \in C_E$ or $c \in C_D$, then
wt(c) ≥ 22 (C₉ has defining set [11, 33] and C₀ is the code number 7 in Table III). If wt(a) = 22, then by Theorem 6b) we also have wt(c) ≥ 22. So we can assume that wt(a) = 11. Now we can say immediately that wt(c) = 2wt(a) + 2wt(b) = 1 + 2wt(b) ≠ 2 mod 3 (since b is an element of the code of Example 22.10 and so wt(b) ≠ 2 mod 3) and so wt(c) ≠ 20 so d ≥ 21. We shall show that we cannot improve this bound by using Theorem 6. Let b(x) be the polynomial corresponding to b. Notice that b(x) ± b(−x) is an element of the code of Example 22.8 (minimum distance 5) and cannot be the zero word (since otherwise wt(c) ≥ 22 + 10 = 32 or c ∈ C₉, which has minimum distance 22). Since a is either zero on the even or zero on the odd positions, we have l₀ ≤ 11 − 5 = 6. So 22 − l₀ ≥ 22 − 6 = 16 and, by Theorem 6a), we have

Table I (Part 2)

<table>
<thead>
<tr>
<th>n</th>
<th>d</th>
<th>w₀</th>
<th>w₁</th>
<th>w₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>13</td>
<td>6</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>22</td>
<td>14</td>
<td>7</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>22</td>
<td>15</td>
<td>6</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>22</td>
<td>16</td>
<td>7</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>22</td>
<td>17</td>
<td>6</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>22</td>
<td>18</td>
<td>7</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>22</td>
<td>19</td>
<td>6</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>22</td>
<td>20</td>
<td>7</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>22</td>
<td>21</td>
<td>6</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>22</td>
<td>22</td>
<td>7</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>6</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>22</td>
<td>24</td>
<td>7</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>22</td>
<td>25</td>
<td>6</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>22</td>
<td>26</td>
<td>7</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>22</td>
<td>27</td>
<td>6</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>22</td>
<td>28</td>
<td>7</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>22</td>
<td>29</td>
<td>6</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>22</td>
<td>30</td>
<td>7</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>22</td>
<td>31</td>
<td>6</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>22</td>
<td>32</td>
<td>7</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 39, NO. 2, MARCH 1993
Theorem 6a), we have \( wt(c) \) 
and so wt(c) by Theorem 6b).

Example 44.3: \( n = 44, R = \{2, 4, 6, 10, 11, 12, 16, 18, 20, 30, 33, 36\} \). We wish to apply Theorem 6 to prove that \( d \geq 16 \).

Suppose \( e = (a + b, -a - b) \) is a nonzero codeword of weight less than 16, where \( (a, -a) \in C_E \) (having defining set \( \{11, 33\} \)) and \( (b, b) \in CO \) (code number 14). As in Example 44.2, we may assume that \( wt(a) = 11 \). If \( b(x) \) is the polynomial corresponding to \( b \), then \( b(x) + b(-x) \) is an element of the code of length 22 with defining set \( \{0, 11\} \) and we may assume that \( b(x) - b(-x) \neq 0 \) (since otherwise \( wt(b) = 0 \) and so \( wt(c) = 22 \)). So \( l_{00} \leq 11 - 2 = 9 \) and \( 22 - l_{00} \geq 13 \). So by Theorem 6a), we have \( wt(e) \geq 3 + 13 - wt(a) - wt(b) = 28 - wt(b) \). But \( wt(b) \equiv 0 \mod 3 \) (by Theorem 2) and we must be in one of the following two cases.

1) \( wt(b) \leq 12 \); \( wt(c) \geq 28 - 12 = 16 \).

2) \( wt(b) \geq 15 \); \( wt(c) \geq 2 \ast 15 - 11 = 19 \) by Theorem 6b).

So \( d \geq 16 \).

Example 44.4: \( n = 44, R = \{1, 2, 3, 4, 5, 6, 9, 10, 12, 15, 16, 18, 20, 23, 25, 27, 30, 31, 36, 37\} \). This code satisfies the condition of Theorem 2, and so \( d \equiv 0 \mod 3 \). \( d_{\text{shift}} = 10 \) and so \( d \geq 12 \). Moreover \( d = 12 \), since this code contains number 13 in Table 3.

Example 44.5: \( n = 44, R = \{0, 1, 2, 3, 4, 5, 6, 9, 10, 12, 15, 16, 18, 20, 23, 25, 27, 30, 31, 36, 37\} \). This code satisfies the conditions of Corollary 1 a) (see Example 44.4). Here \( d_{\text{shift}} = 10 \), so we have by Corollary 1 c) that \( d \geq 11 \).

Example 44.6: \( n = 44, R = \{0, 7, 8, 13, 14, 17, 19, 21, 22, 24, 26, 28, 29, 32, 34, 35, 38, 39, 40, 41, 42, 43\} \). We wish to apply Theorem 6 to prove that \( d \geq 10 \). Suppose \( e = (a + b, -a - b) \) is a nonzero codeword of weight less than 10, where \( (a, -a) \in C_E \) (number 19) and \( (b, b) \in CO \) (number 14). Notice that \( C_E \) is a square with minimum distance 10, i.e., if \( wt(a) < 20 \), then \( wt(a) \) (and also \( e \)) is either zero on the even or on the odd positions (and if it is zero both on the even and on the odd positions, then \( wt(e) \geq 18 \)). Let \( b(x) \) be the polynomial corresponding to \( b \), then \( b(x) + b(-x) \) has zeros 1 and -1 and so \( b(x) \) must have weight at least 2 on both the even and the odd positions if it is zero on either the even or the odd positions, then \( b = 0 \) and \( wt(e) \geq 10 \). We have three cases.

1) \( wt(a) \geq 9 \) or \( wt(b) \geq 9 \). Then, \( wt(e) \geq 10 \) by Theorem 6b).

2) \( wt(a) < 9 \) and \( wt(b) = 9 \). Then, \( wt(e) \geq 10 \) by Theorem 6b).

3) \( wt(a) = 9 \) and \( wt(b) = 9 \). Then by the observations just made \( l_{00} \leq 2 + (11 - 2) = 11 \) and so by Theorem 6a) we have: \( wt(e) \leq 3 \ast 11 - 9 = 9 \).

So \( d \geq 10 \), and since code nr.19 in Table III is contained in this code, we have \( d = 10 \).

Example 44.7: \( n = 44, R = \{1, 3, 5, 8, 9, 11, 15, 22, 23, 24, 25, 27, 28, 31, 32, 33, 37, 40\} \). We wish to apply Theorem 6 to prove that \( d \geq 8 \). Suppose \( c = (a + b, -a - b) \) is a nonzero codeword of weight < 8, where \( (a, -a) \in C_E \) (equivalent to number 13) and \( (b, b) \in CO \) (number 38). \( C_E \) is self-orthogonal and has minimum distance 12, so \( wt(c) \) can only be

### Table 11

<table>
<thead>
<tr>
<th>( k )</th>
<th>( d_{\text{shift}} )</th>
<th>( 2 \ast k )</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>44</td>
<td>88</td>
<td>BCH</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>44</td>
<td>BCH</td>
</tr>
<tr>
<td>3</td>
<td>26</td>
<td>52</td>
<td>BCH</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>48</td>
<td>Example 40.1</td>
</tr>
<tr>
<td>5</td>
<td>22</td>
<td>44</td>
<td>Example 40.2</td>
</tr>
<tr>
<td>6</td>
<td>24</td>
<td>48</td>
<td>BCH</td>
</tr>
<tr>
<td>7</td>
<td>22</td>
<td>44</td>
<td>BCH</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>40</td>
<td>BCH</td>
</tr>
<tr>
<td>9</td>
<td>18</td>
<td>36</td>
<td>BCH</td>
</tr>
<tr>
<td>10</td>
<td>18</td>
<td>36</td>
<td>BCH</td>
</tr>
<tr>
<td>11</td>
<td>17</td>
<td>34</td>
<td>BCH</td>
</tr>
<tr>
<td>12</td>
<td>16</td>
<td>32</td>
<td>BCH</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>28</td>
<td>BCH</td>
</tr>
<tr>
<td>14</td>
<td>13</td>
<td>26</td>
<td>BCH</td>
</tr>
<tr>
<td>15</td>
<td>12</td>
<td>24</td>
<td>BCH</td>
</tr>
<tr>
<td>16</td>
<td>11</td>
<td>22</td>
<td>BCH</td>
</tr>
<tr>
<td>17</td>
<td>10</td>
<td>20</td>
<td>BCH</td>
</tr>
<tr>
<td>18</td>
<td>9</td>
<td>18</td>
<td>BCH</td>
</tr>
<tr>
<td>19</td>
<td>8</td>
<td>16</td>
<td>BCH</td>
</tr>
<tr>
<td>20</td>
<td>7</td>
<td>14</td>
<td>BCH</td>
</tr>
<tr>
<td>21</td>
<td>6</td>
<td>12</td>
<td>BCH</td>
</tr>
<tr>
<td>22</td>
<td>5</td>
<td>10</td>
<td>BCH</td>
</tr>
<tr>
<td>23</td>
<td>4</td>
<td>8</td>
<td>BCH</td>
</tr>
<tr>
<td>24</td>
<td>3</td>
<td>6</td>
<td>BCH</td>
</tr>
<tr>
<td>25</td>
<td>2</td>
<td>4</td>
<td>BCH</td>
</tr>
<tr>
<td>26</td>
<td>1</td>
<td>2</td>
<td>BCH</td>
</tr>
</tbody>
</table>

### Explanation

- **BCH**: Binary cyclic code
- **Example**: Example from the text
Table III  
(Part 1)  

<table>
<thead>
<tr>
<th>n</th>
<th>m</th>
<th>e</th>
<th>(d_k)</th>
<th>(d_m)</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>0</td>
<td>20</td>
<td>10</td>
<td>(0.2,4,7,8)</td>
</tr>
<tr>
<td>2</td>
<td>41</td>
<td>9</td>
<td>18</td>
<td>13</td>
<td>(2,4,7,8)</td>
</tr>
<tr>
<td>3</td>
<td>41</td>
<td>11</td>
<td>12</td>
<td>7</td>
<td>(4,7,8)</td>
</tr>
<tr>
<td>4</td>
<td>41</td>
<td>24</td>
<td>8</td>
<td>5</td>
<td>(0.2,8)</td>
</tr>
<tr>
<td>5</td>
<td>41</td>
<td>25</td>
<td>8</td>
<td>5</td>
<td>(0.2,8)</td>
</tr>
<tr>
<td>6</td>
<td>44</td>
<td>5</td>
<td>24</td>
<td>16</td>
<td>(0.2,4,7,8,11,14,22)</td>
</tr>
<tr>
<td>7</td>
<td>44</td>
<td>5</td>
<td>24</td>
<td>16</td>
<td>(0.2,4,7,8,11,14,22)</td>
</tr>
<tr>
<td>8</td>
<td>44</td>
<td>5</td>
<td>24</td>
<td>16</td>
<td>(0.2,4,7,8,11,14,22)</td>
</tr>
<tr>
<td>9</td>
<td>44</td>
<td>5</td>
<td>24</td>
<td>16</td>
<td>(0.2,4,7,8,11,14,22)</td>
</tr>
<tr>
<td>10</td>
<td>44</td>
<td>5</td>
<td>24</td>
<td>16</td>
<td>(0.2,4,7,8,11,14,22)</td>
</tr>
<tr>
<td>11</td>
<td>44</td>
<td>5</td>
<td>24</td>
<td>16</td>
<td>(0.2,4,7,8,11,14,22)</td>
</tr>
<tr>
<td>12</td>
<td>44</td>
<td>5</td>
<td>24</td>
<td>16</td>
<td>(0.2,4,7,8,11,14,22)</td>
</tr>
<tr>
<td>13</td>
<td>44</td>
<td>5</td>
<td>24</td>
<td>16</td>
<td>(0.2,4,7,8,11,14,22)</td>
</tr>
<tr>
<td>14</td>
<td>44</td>
<td>5</td>
<td>24</td>
<td>16</td>
<td>(0.2,4,7,8,11,14,22)</td>
</tr>
<tr>
<td>15</td>
<td>44</td>
<td>5</td>
<td>24</td>
<td>16</td>
<td>(0.2,4,7,8,11,14,22)</td>
</tr>
<tr>
<td>16</td>
<td>44</td>
<td>5</td>
<td>24</td>
<td>16</td>
<td>(0.2,4,7,8,11,14,22)</td>
</tr>
<tr>
<td>17</td>
<td>44</td>
<td>5</td>
<td>24</td>
<td>16</td>
<td>(0.2,4,7,8,11,14,22)</td>
</tr>
<tr>
<td>18</td>
<td>44</td>
<td>5</td>
<td>24</td>
<td>16</td>
<td>(0.2,4,7,8,11,14,22)</td>
</tr>
<tr>
<td>19</td>
<td>44</td>
<td>5</td>
<td>24</td>
<td>16</td>
<td>(0.2,4,7,8,11,14,22)</td>
</tr>
<tr>
<td>20</td>
<td>44</td>
<td>5</td>
<td>24</td>
<td>16</td>
<td>(0.2,4,7,8,11,14,22)</td>
</tr>
</tbody>
</table>

smaller than 8 if \(wt(a) = 6\) (if \(wt(a) = 0\), then \(c \in C_o\), which has minimum distance 8). But if \(wt(a) = 6\), then \(a\) is zero on either the even or the odd positions. By Theorem 6b) \(wt(c)\) can only be smaller than 8 if \(wt(b) \leq 6\). So we have two cases.

1) \(m - l_00 = 6\). Then \(c\) is in the lower square \(C_l\) by the observations just made (\(c\) is zero on either the even or the odd positions). But \(d_2 = 9\), so this is impossible.

2) \(m - l_00 \geq 7\). Then, by Theorem 6a), we have: \(wt(c) \geq 21 - 6 - 6 = 9\). So \(d \geq 8\), and since code number 38 is contained in it, we have \(d = 8\).

Example 44.8: \(n = 44\), \(R = \{7,8,13,17,19,21,22,24,28,29,32,35,39,40,41,43\}\). We wish to apply Theorem 6 to prove that \(d \geq 6\). Suppose \(c = (a + b, -a + b)\) is a nonzero codeword of weight < 6, where \(a, -a \in C_E\) (number 19)
and \((b,b) \in C_T\) (number 38). The only way to get \(w_t(e) < 6\), is to take \(w_t(a) = 5\) and \(w_t(b) = 4\) or 5. But \(C_T\) is a square and so \(w_t(a) = 5\) implies that \(a\) is zero on either the even or the odd positions. We have two cases.

1) \(m - l_{00} = 5\). Then \(e\) is an element of the lower square, which has minimum distance 6,
2) $m - l_0 \geq 6$. Then, by Theorem 6a), we have $wt(c) \geq 18 - 5 - 5 = 8$.

So $d \geq 6$, and since the lower square has minimum distance 6, we have $d = 6$.

Example 44: $n = 44, R = \{1, 3, 5, 8, 9, 15, 22, 23, 24, 25, 27, 28, 31, 32, 37, 40\}$. We wish to apply Theorem 6 to prove that $d \geq 7$. Suppose $c = (a+b, -a+b)$ is a nonzero codeword of weight < 7, where $(a, -a) \in \mathcal{C}_E$ (equivalent to number 19) and $(b, b) \in \mathcal{C}_O$ (number 38). The only way to get $wt(c) < 7$ is by taking $wt(a) = 5$ and $wt(b) = 4$ or 5 or $wt(a) = 6$ and $wt(b) = 6$. But $\mathcal{C}_E$ is a square and so $wt(a) < 10$ implies that $a$ is zero on either the even or the odd positions. We have two cases.

1) $m - l_0 = wt(a)$. Then, $c$ is an element of the lower square, which has minimum distance 7.

2) $m - l_0 \geq wt(a) + 1$. Then, by Theorem 6a), we have

<table>
<thead>
<tr>
<th>$n$</th>
<th>minimal polynomials</th>
<th>zeros exponents</th>
<th>permutations</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>$x^5 + x^4 + 2x^2 + 2x + 1$</td>
<td>1,3,5,9,15</td>
<td>1 2 3 4 5 6</td>
</tr>
<tr>
<td>2</td>
<td>$x^5 + 2x^4 + 2x^3 + 2x^2 + 1$</td>
<td>2,6,8,10,18</td>
<td>1 5 4 3 2 6</td>
</tr>
<tr>
<td>3</td>
<td>$x^5 + x^4 + 2x^2 + 2x + 1$</td>
<td>4,12,14,16,20</td>
<td>6 4 5 2 3 1</td>
</tr>
<tr>
<td>27</td>
<td>$x^5 + 2x^4 + 2x^3 + 2x^2 + 1$</td>
<td>7,13,17,19,21</td>
<td>11</td>
</tr>
<tr>
<td>26</td>
<td>$x^5 + 2x^4 + 2x^3 + 2x^2 + 1$</td>
<td>1,2,3,4</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>$x^5 + 2x^4 + 2x^3 + 2x^2 + 1$</td>
<td>1,2,3,4</td>
<td>1 2 3 4</td>
</tr>
</tbody>
</table>

...
wt(c) ≥ 3(wt(a) + 1) - wt(a) - wt(b) = 2wt(a) - wt(b) + 3 ≥ 8.

So d ≥ 7, and since the lower square has minimum distance 7, we have d = 7.

**Example 46.1:** n = 46, R = \{1, 3, 9, 10, 13, 14, 20, 22, 23, 25, 27, 28, 29, 30, 31, 34, 35, 38, 39, 40, 41, 42, 44\}. \(d_{\text{SHEF}} = 8\). The conditions in Corollary 1 are satisfied. By Corollary 1 a) weights are congruent to 0 or 1 mod 3. So d ≥ 9.

**APPENDIX**

In Table IV, we give a complete list of all irreducible factors of \(x^n - 1\), n ≤ 50, over GF(3) (a list for n ≤ 100 can be found in [9]). We also give the exponents of the zeros of these polynomials for some chosen nth root of unity. As we saw, a cyclic code of length n can be represented by a set of irreducible factors of \(x^n - 1\). In the last column, we give the permutations on the minimal polynomials, that permute
codes into equivalent codes. If \( x^{n-1} + x^{n-2} + \cdots + x + 1 \) is an irreducible factor of \( x^n - 1 \), this will be denoted by "trivial."

**ACKNOWLEDGMENT**

The authors are very grateful to F. Bussemaker, Department of Mathematics, Eindhoven University of Technology, for writing the programs.

**REFERENCES**


