A Class of Cyclic Codes that can Approach the Shannon Bound

Minyue Fu, Weizhou Su and Jun Ning
Department of Electrical and Computer Engineering
University of Newcastle, NSW 2308 Australia

June 2000

Abstract. In this paper, we propose a class of cyclic codes and show that this class of error correcting codes can approach the Shannon bound when applied to a symmetric binary channel.

1 Introduction

The purpose of this paper is to show that there exists a class of cyclic codes that can approach the Shannon bound when applied to a symmetric binary channel. We believe this class of cyclic codes has not been studied before and that the coding properties of these codes may be of theoretical interest.

2 Cyclic Codes

In this section, we introduce the class of cyclic codes to be considered in this paper.

Given any integer $r > 1$, let $GF(r)$ denote the Galois field $\{0, 1, \cdots, r-1\}$ with the standard mod $r$ addition and multiplication. A polynomial is said to be in $GF(r)$ if every coefficient is in $GF(r)$.

The cyclic codes we consider in this paper are of the following form:

$$c(x) = m(x)T(x) \mod x^q - 1$$

(2.1)

where $m(x) = [m_1(x) \ m_2(x) \ \cdots \ m_k(x)]$ is a polynomial vector representing a message and $c(x) = [c_1(x) \ c_2(x) \ \cdots \ c_n(x)]$ is a polynomial vector representing the codeword of $m(x)$. $T(x)$ is a $n$ by $k$ polynomial matrix which maps $m(x)$ to $c(x)$, all polynomials are in $GF(r)$, and $n, k, q, r$ are positive integers. This code is obviously a linear code, mapping $GF(r)^k$ to $GF(r)^nq$. Such a code will be denoted by $C[n, k, q, r]$, or $C$ for short.

The code (2.1) is indeed a cyclic code because $xc(x) = xm(x)T(x) \mod x^q-1$ is generated by $xm(x)$ mod $x^q-1$. So $xc(x) \in C$ if $c(x) \in C$.

The cyclic code (2.1) is called systematic if

$$T(x) = [I \ G(x)]$$

(2.2)

i.e.,

$$c(x) = [m(x) \ m(x)G(x)]$$

(2.3)

The coding ratio for a systematic cyclic code is $k/n$, regardless of $q$ and $r$.

3 Main Result

Given a symmetric binary channel with error probability $p \in (0, 1/2)$, the channel capacity is given by $C(p) = 1 + p \log p + (1 - p) \log (1 - p)$. Given a received $n$-bit word, the minimum distance decoding method refers to
the method which decodes the received word to be the message (or one of the messages) whose code is closest to the received signal in the sense that the least number of bits get altered.

A class of codes is said to be able to approach the Shannon bound if for any error probability \( p \in (0, 1/2) \), bound \( \epsilon > 0 \) on capacity loss and bound \( \delta > 0 \) on decoding error probability, there exists a code in this class such that its code rate \( k/n \geq C(p) - \epsilon \) and the probability of decoding error using the minimum distance decoding method is no larger than \( \delta \). The well-known Shannon’s channel capacity theorem [1] proves that the class of all (nonlinear) codes can approach the Shannon bound. It is also known [2] that linear codes can approach the Shannon bound as well.

The purpose of this section is to study the asymptotic performance of the class of systematic cyclic codes in (2.2).

To implement the code on a binary channel, we need a binary to base \( r \) converter. This is only needed if \( r \neq 2 \) and is constructed as follows: Define

\[
q = [(q - 1) \log_2 r]; \quad \hat{q} = [q \log_2 r] + 1
\]

This implies that

\[
2^\hat{q} \leq r^q - 1 \leq 2^{\hat{q} + 1}; \quad 2^q \geq r^\hat{q} \geq 2^{q - 1}
\]

We can map \( GF(2)^\hat{q} \) into \( GF(r)^\hat{q} \) using the standard base-2 to base-\( r \) conversion. For each \( \hat{m} \in GF(2)^\hat{q} \), the corresponding result \( m_i \in GF(r)^\hat{q} \) has zero for the most significant symbol. We will see how this property is important. Each codeword component \( c_i \in GF(r)^\hat{q} \) is mapped back to \( \hat{c}_i \in GF(2)^\hat{q} \) using the standard base-\( r \) to base-2 conversion.

Therefore, the complete encoding procedure is as follows: Given any binary message \( \hat{m} \in GF(2)^\hat{k} \), split it into components \( m_i \in GF(2)^\hat{q}, i = 1, \ldots, k \). Convert each \( m_i \) into \( m_i \in GF(r)^\hat{q} \). Construct the codeword components \( c_i \in GF(r)^\hat{q} \) as in (2.3) and convert it into \( \hat{c}_i \in GF(2)^\hat{q} \). The coding ratio of this code is given by \( R = k\hat{q}/n\hat{q} \), and it is easy to verify that

\[
\frac{k}{n} \geq R \geq \frac{k}{n} \frac{(q - 1) \log_2 r - 1}{q \log_2 r + 1} \geq \frac{k}{n} \frac{q - 2}{q + 1}
\]

Therefore, \( k/n \) remains to be the coding ratio of the code if \( q \) is sufficiently large.

**Theorem 1.** Given a symmetric binary channel with error probability \( p \in (0, 1/2) \). Given any bound \( \epsilon > 0 \) on channel capacity loss, choose \( k \) and \( n \) such that the coding ratio \( k/n \) is such that

\[
C(p) - \epsilon \leq k/n \leq C(p) - \frac{\epsilon}{2}
\]

Take \( q \) be a prime number and let \( r \) be its smallest root (which always exists). Apply the code (2.3) with the base-2 to base-\( r \) converter described above. Then, the decoding error probability based on the minimum distance decoding criterion approaches to zero as \( q \to \infty \).

The proof of Theorem 1 requires a key result described below (see Appendix A for proof).

**Lemma 1.** Let \( q \) be any odd prime number and \( r \) be its smallest root. Then, \( x - 1 \) and \( x^{q - 1} + x^{q - 2} + \cdots + x + 1 \) are the only two irreducible factors of \( x^q - 1 \) in \( GF(r) \).

**Proof of Theorem 1.** Let the code be constructed as in the theorem. Given any bound \( \delta > 0 \) on decoding error probability. Take \( \mu > 0 \) to be such that \( C(p) - \epsilon/4 = C(p + \mu) \). Each binary codeword \( \hat{c} \) has \( \hat{q}n \) bits. By the weak law of large numbers, the probability that the number of transmission errors for \( \hat{c} \) to exceed \( (p + \mu)\hat{q}n \) is less than \( \delta \) when \( n\hat{q} \) is sufficiently large. So we can choose \( q \) sufficiently large to have this property.

The number of possible non-zero codewords \( \hat{c} \) to have weight less than \( (p + \mu)\hat{q}n \) is bounded by ([3], p. 115)

\[
\sum_{i=1}^{(p+\mu)\hat{q}n} \binom{i}{\hat{q}n} \leq 2^{H(p+\mu)\hat{q}n}
\]

Using (3.2), the above is in turn bounded by

\[
2^{H(p+\mu)\hat{q}n} \leq 2^{H(p+\mu)n\hat{q}n - (C(p) - \epsilon/4)\hat{q}n} \leq r^{\hat{q}n - (C(q) - \epsilon/2)\hat{q}n} p_1(q) \leq r^{(n-k)\hat{q}n} p_1(q)
\]

2
where \( \rho_1(q) = 2^H(p+\mu)n \) for an even \( n \).

For each \( c \) with weight \( \leq (p + \mu)n \), there is at most one \( c(x) \) corresponding to it (as sometimes there may not be a codeword corresponding to it). For each of these \( c(x) \), the message \( m(x) \) is unique because the code is systematic. Therefore, each component \( m_i(x) \) is unique. Because the most significant symbol in \( m_i(x) \) is zero, Lemma 1 implies that \( m_i(x) \) either does not contain any factor (apart from a constant) of \( x^q - 1 \) or \( m_i(x) \) contains \( x - 1 \) as a factor. Therefore, \( m(x) \) either contains \( x - 1 \) as a factor or does not contain any factor of \( x^q - 1 \) (apart from a constant). Rewriting (2.3) as

\[
c_{k+i}(x) = m(x)g_i(x) \mod x^q - 1, \quad \forall i = 1, \ldots, n - k
\]

where \( g_i(x) \) is the \( i \)th column of \( G(x) \). For a given \( c_{k+i}(x) \), \( g_i(x) \) has \( r^q(k-1) \) solutions if \( m(x) \) does not contain a factor of \( x^q - 1 \). If \( m(x) \) contains \( x - 1 \) as a factor, then \( g_i(x) \) either has no solution (when \( c_{k+i}(x) \) does not contain the factor \( x - 1 \)) or has at most \( rr^q(k-1) \) solutions (if \( c_{k+i}(x) \) also contains \( x - 1 \)) because in the latter case for each solution \( g_i(x) \), \( g_i(x) + a(1 + x + \cdots + x^{r-1}) \) is also a solution for any \( a \in GF(r) \). In any case, there are at most \( r^q(k-1)+1 \) solutions of \( g_i(x) \) for each \( c_{k+i}(x) \). Therefore, there are at most \( r^q(n-k)\rho_1(q) \) number of non-zero codewords with weight less than or equal to \((p + \mu)nq\), the maximum number of \( G(x) \) which can lead to a non-zero codeword with weight less than \( (p + \mu)nq \) is bounded by

\[
\rho_2(q) = r^q(n-k)\rho_1(q) = 2^H(p+\mu)n x(q)^{q(n-k)q}. \quad \rho_2(q) = 2^H(p+\mu)n x(q)^{q(n-k)q}.
\]

where \( \rho_2(q) = r^q(n-k)\rho_1(q) = 2^H(p+\mu)n x(q)^{q(n-k)q} \). Note in particular that \( \rho_2(q) \) is sufficiently large. Since there are \( r^q(n-k)\rho_1(q) \) number of possible \( G(x) \), there must exist some \( G(x) \) when \( q \) is sufficiently large such that no non-zero codeword has weight less than or equal to \((p + \mu)nq\). Therefore, the probability of decoding error will be less than \( \delta \). Since \( \delta \) is arbitrary, the probability of decoding error approaches zero as \( q \to \infty \).

### 4 Conclusion

In this paper, we have analyzed the asymptotic performance of the class of cyclic codes in (2.3). Namely, we have shown its capability to approach the Shannon bound by taking the polynomial degree \( q \) to be sufficiently large and taking \( n, k \) so that they approximate the channel capacity well.

Several concluding technical remarks are in order.

First, we note that the design philosophy behind such a cyclic code may be very different from the classical cyclic codes such the BCH codes where the focus is to use factors of \( x^q - 1 \). Here, \( q \) and \( r \) are chosen to avoid many factors of \( x^q - 1 \) from appearing, or at least this is the way we have managed to approach the Shannon bound. It remains a research problem as to how to best construct good codes of this class.

Secondly, we have not addressed the decoding problem for the codes in (2.3). This remains to be an interesting problem. One possible approach is to use an iterative decoding algorithm. More precisely, we can separate \( m(x)G(x) \) into two parts by splitting \( G(x) \) into \( [G_1(x) \ G_2(x)] \). This gives two systematic subcodes \( c_1(x) = [m(x) \ m(x)G_1(x)] \) and \( c_2(x) = [m(x) \ m(x)G_2(x)] \). (Obviously, we may split the code \( c(x) \) into many subcodes if we want.) Then, iterative decoding algorithms such as those used in decoding of turbo codes may be applied; see [7].

Finally, we note on the use of the \( GF(r) \) field in construction of our cyclic codes. This causes some inconvenience in the implementation, as a binary to base-\( r \) converter is required. The need for the \( GF(r) \) field is purely technical, and can be avoided if arbitrarily large prime numbers can be found with 2 as its root. We conjecture that this is the case based on simulation, but no mathematical proof is available. So we leave this as an open number theory problem.
A Proof of Lemma 1

A.1 Idempotent Polynomials

A polynomial \( f(x) \) in \( GF(r) \) is called irreducible if it is not the product of two polynomials in \( GF(r) \) of degree at least one. An irreducible factor of \( x^n - 1 \) (in \( GF(r) \)) is an irreducible polynomial of degree at least one that divides \( x^n - 1 \). It is known that the irreducible factors of \( x^n - 1 \) are unique for each \( n \). There is a close relationship between the factors of \( x^n - 1 \) and the so-called idempotent polynomials.

A polynomial \( I(x) \) in \( GF(r) \) is called an idempotent polynomial (with respect to \( n \)) if \( I(x) = I'(x) \mod x^n - 1 \).

Idempotent polynomials are closely related to cyclic codes and irreducible factors. Denote \( Z_n = \{0, 1, \cdots, n-1\} \) and define

\[
C_i = \{ j = r^k \cdot i \mod n \mid k = 0, 1, \cdots, s \}, i = 0, 1, \cdots, n - 1
\]

(A.1)

where \( s \) is the order of \( r \), i.e., \( r \) is the minimum integer with \( r^s \mod n = 1 \). It is easy to check that \( \{C_i\} \) forms a partition of \( Z_n \) in the following sense: \( \cup_i C_i = Z_n \) and that \( C_i \) and \( C_{i'} \) are either identical or disjoint for different \( i \) and \( i' \). Also define

\[
c_i(x) = \sum_{j \in C_i} x^j
\]

Then, the key properties of idempotent polynomials are summarised below.

Lemma 2. A polynomial \( I(x) \) is an idempotent iff

\[
I(x) = \sum_{i \in K} a_i c_i(x), \quad a_i \in GF(r)
\]

(A.2)

where \( K \) is a set of indices corresponding to different \( c_i(x) \). In particular, the number of non-zero idempotent polynomials is equal to \( r^{[K]} - 1 \), where \( |K| \) denotes the number of elements in \( K \).

Proof: It is easy to see that every \( c_i(x) \) is an idempotent because

\[
c_i'(x) = \left( \sum_{j \in C_i} x^j \right)^r = \sum_{j \in C_i} x^{rj} = \sum_{j' \in C_i} x^{j'} \mod x^n - 1
\]

Also, \( a_i c_i(x) \) is an idempotent for every \( a_i \in GF(r) \) because \( a_i^r = a_i \mod r \). It is also trivial that the sum of idempotents is also an idempotent. So every polynomial in the form of (A.2) is an idempotent. To show that every idempotent must be in the form of (A.2), we assume that \( I(x) \) contains a term \( ax^j \) with \( j \in C_i \) for some \( i \) and \( a \in GF(r) \). It follows that \( I^{r^k}(x) \) contains all \( a^{r^k} x^{j'} = ax^{j'} \) with \( j' = r^k j \mod n, k = 2, 3, \cdots \), i.e., \( I^{r^k}(x) \) contains \( ax^{j'} \) for all \( j' \in C_i \). Since \( I^{r^k}(x) = I(x) \), we conclude that \( I(x) \) contains all \( ax^{j'}, j' \in C_i \) Hence, (A.2) covers all idempotents.

Lemma 3. Suppose \( g.c.d.(n, k) = 1 \). Then, \( g.c.d.(x^n - 1, x^k - 1) = x - 1 \) in \( GF(r) \) for all \( r > 1 \).

Proof. Without loss of generality, assume \( n > k \). Denote

\[
f_m(x) = x^{m-1} + \cdots + x + 1
\]

Obviously, \( g.c.d.(x^n - 1, x^k - 1) = x - 1 \) iff \( g.c.d.(f_m(x), f_k(x)) = 1 \). To show the latter, we apply the Euclidean algorithm. Express \( n = kq + j \) for some \( j < k \). Then \( j \neq 0 \) and \( g.c.d.(j, k) = 1 \). Consequently, \( f_n(x) = f_j(x) \mod f_k(x) \). So \( g.c.d.(f_n(x), f_k(x)) = 1 \) iff \( g.c.d.(f_k(x), f_j(x)) = 1 \). Reassign \( (n, k) = (k, j) \) and repeat the above argument until \( f_j(x) \) has degree 0. In this case, \( g.c.d.(f_k(x), f_j(x)) = 1 \). Hence, the result holds.

Lemma 4. \( x^n - 1 \) in \( GF(r) \) does not contain any factor of the form \( a^2(x) \), where \( a(x) \) is a polynomial in \( GF(r) \) with degree at least one.
Proof. Suppose $x^n - 1 = a^2(x)b(x)$ for some $a(x)$ and $b(x)$. Without loss of generality, assume $a(x)$ is monic, irreducible, and with degree at least one. The above implies
\[ a^r(x)b(x) = a(x^r)b(x) = (x^n - 1)a^{r-2}(x) \]
Decompose $a^{r-2}(x)$ into
\[ a^{r-2}(x) = \sum_{i=0}^{r-1} x^i a_i(x^r) \]
and write $n = kr + j$ with $0 < j < r$ and $\gcd(j, r) = 1$ (because $n$ is prime). Then,
\[ (x^n - 1)a^{r-2}(x) = \sum_{i=0}^{r-1} x^i(x^{kr+j} - 1)a_i(x^r) = \sum_{i=0}^{r-1} x^i(x^{kr}a_{-j+i}(x^r) - a_i(x^r)) \]
where the subscript in $a_{-j+i}(x^r)$ is mod $r$. Hence, $a(x)$ divides $x^ka_{-j+i}(x) - a_i(x)$ for all $i$. It follows that $a(x)$ divides $x^k(x^ka_{-2j+i}(x) - a_{-j+i}(x))$, $x^{2k}(x^ka_{-3j+i}(x) - a_{-2j+i}(x))$, etc. Consequently, $a(x)$ divides $x^k a_{-j+i}(x) - a_i(x) = (x^k - 1)a_i(x)$ for all $i$. Since $x^k - 1 = (x^k - 1)^r$, $a(x)$ either divides $x^k - 1$ or $a_i(x)$ for all $i$. From Lemma 3, we know that $\gcd(x^n - 1, x^k - 1) = x - 1$. If $a(x)$ divides $x^k - 1$ (and because $a(x)$ also divides $x^n - 1$, $a(x)$ must contain $x - 1$ and by its irreducibility, $a(x) = x - 1$. But this is not possible because $(x - 1)^2$ does not divide $x^n - 1$. Therefore, $a(x^r)$ must divide $a_i(x^r)$ for all $i$. Consequently, $a^r(x)$ divides $a^{r-2}(x)$, which is not possible. Therefore, the result in Lemma 4 holds.

The key relationship between idempotents and cyclic codes is given below.

Lemma 5. Given any cyclic code, let $g(x)$ be its minimal generator. Then, there must exists an idempotent $I(x)$ such that
\[ g(x) = \gcd(I(x), x^n - 1) \]  
(A.3)

Proof. Since $g(x)$ is a minimal generator, there exists a unique $h(x)$ such that $g(x)h(x) = x^n - 1$. From Lemma 4, $\gcd(h(x), g(x)) = 1$. By the Euclidean algorithm, there exists polynomials $t(x)$ and $s(x)$ such that
\[ 1 = t(x)g(x) + s(x)h(x) \]
Multiplying both sides by $t(x)g(x)$ gives
\[ t(x)g(x) = (t(x)g(x))^2 + t(x)s(x)(x^n - 1) = (t(x)g(x))^2 \pmod{x^n - 1} \]
It follows that
\[ t(x)g(x) = (t(x)g(x))^r \]
Hence, $t(x)g(x)$ is an idempotent. Note that $\gcd(t(x), h(x)) = 1$. Hence,
\[ g(x) = \gcd(t(x)g(x), x^n - 1) \]

A.2 Irreducible Factors of $x^n - 1$

Now we are ready to prove Lemma 1.

Since $n$ is a prime number and $r$ is its root, there are only two partitions of $\{0, 1, \cdots, n-1\}$, namely, $C_0 = \{0\}$ and $C_1 = \{1, 2, \cdots, n-1\}$. The only idempotents are of the form:
\[ I(x) = a_0 + a_1(x + \cdots + x^{n-1}), \quad a_0, a_i \in GF(r) \]
To find out all the factors of $x^n - 1$, we only need to consider all such $I(x)$ (see Lemma 5). For $a_1 = 0$, the factor is the trivial factor $1$. For $a_1 \neq 0$ and $a_0 = -1a_1 \mod r$, $x - 1$ is a factor. For $a_1 = a_0 \neq 0$, the factor is $1 + x + \cdots + x^{n-1}$. It can be shown by the Euclidean algorithm that no other factors exist in other cases of $a_0$ and $a_1$. Indeed, let $a_1 \neq 0$ and divide $a_1(x^{n-1} + \cdots + x + 1)$ by $a_1(x^{n-1} + \cdots + x) + a_0$ gives the reminder $a_1 - a_0 \neq 0 \mod r$. By the Euclidean algorithm, the only common factor $a_1(x^{n-1} + \cdots + x + 1)$ and $a_1(x^{n-1} + \cdots + x) + a_0$ is a constant, i.e., $1$. Therefore, the only non-trivial factors of $x^n - 1$ are $x - 1$ and $x^{n-1} + \cdots + x + 1$, which must be irreducible (otherwise there would be more non-trivial factors).
References

