Coding against cyclic shifts*

Tjalling Tjalkens

Eindhoven University of Technology
Department of EE
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

t.j.tjalkens@tue.nl

Abstract

In several practical settings we wish to communicate an identifier based on flashing lights. The communication channels here use small alphabets, are error prone, unsynchronized, and require short code words. Nevertheless the number of identifiers is relatively large. We model this situation as a cyclic shift channel where the shift length equals the code wordlength. We study the behaviour of the capacity of this channel and define a class of self synchronized error correcting codes.

1 Introduction

In a Brain Computer Interface (BCI) experiment subjects are asked to look at one of several different flashing lights. EEG signals are measured to detect the response. The lights can flash in various frequencies or relative phases, and these different patterns can represent the different commands that the subject thus issues.

The simplest communication model that abstracts this process is a Discrete Memoryless (Time-invariant) Channel (DMC).The channel input X is chosen from the A different frequencies/phases from the light that the subject is looking at and the output values Y are the detected/estimated frequencies/phases. The A different signal values are denoted by the alphabet $\mathcal{A} = \{1, 2, \ldots, A\}$.

This DMC models the *confusion matrix* $P(Y|X)$ that arises from the BCI experiments. The capacity for this well studied channel is given by

$$C_{\text{DMC}} = \max_{P(X)} I(X;Y). \tag{1}$$

The maximum in (1) is realized by memoryless probabilities of the form $P(X^N) = \prod_{i=1}^{N} P(X_i)$.

In a BCI setting we assume that every light transmits a fixed sequence of input signals (frequency/phase) repeatedly.

The test subject is looking at the light in an unsynchronized manner. We model this by adding a random shift unit to the channel. This cyclic shift unit works on the full word length N. In Figure 1 π_s denotes a cyclic shift that depends on the value of a *shift* value s. More details on this notation will follow shortly, see definition 1.

As we shall show later, there is an alternative and identical view of the CSC as depicted in Figure 2.

Note that we can replace the test subject by a device with a light sensor. By pointing this device at a lights that flash in various patterns, one can identify different devices or commands.

*This work is in part supported by ENIAC Joint Undertaking, grant 270707-2, EnLight.

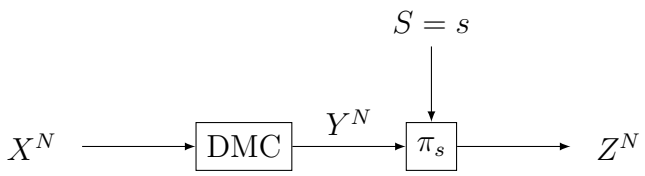

Figure 1: The Cyclic Shift Channel

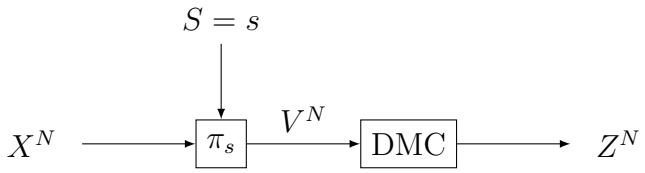

Figure 2: An alternative view to the CSC

2 The capacity of the CSC

In a practical setting the word length N of the CSC is severely constrained by the fact that subjects cannot and will not look at a single light for a long time. The capacity of a channel is an asymptotic result where the word lengths can be arbitrarily large and good codes often have long word lengths. Capacity considerations are still useful in order to compare the performance capabilities of different settings and the capacity achieving probability is still a guideline in designing good codes.

We define the capacity of the CSC from Figure 1 as

$$C_{\text{CSC}} = \lim_{N\to\infty} \sup_{P(X^N)} \frac{1}{N} I(X^N; Z^N). \tag{2}$$

It turns out, not unexpectedly, that the capacity of the CSC equals the capacity of its constituent DMC as given in (1). We state and prove that in the following theorem, but first we define a formal notation for a cyclic shift.

Definition 1. *The sequence obtained from x^N after k cyclic shifts is denoted by $\pi_k x^N$.*

Also, the inverse shift is denoted by π_k^{-1}, so

$$x^N = \pi_k^{-1}\pi_k x^N = \pi_k \pi_k^{-1} x^N.$$

Theorem 1. *Let X^N be the input of a CSC of word length N and Z^N be the corresponding output. Let X^N also be the input of the constituent DMC channel based on the confusion matrix and let Y^N be its output. This implies that Z^N is a cyclic shift of Y^N over a random number of positions. The random variable S denotes this shift and we say that if $S = s$ then $Z^N = \pi_s Y^N$. We have*

$$C_{CSC} = \lim_{N\to\infty} \sup_{P(X^N)} \frac{1}{N} I(X^N; Z^N) = \lim_{N\to\infty} \sup_{P(X^N)} \frac{1}{N} I(X^N; Y^N) = C_{DMC}. \tag{3}$$

The proof of this theorem is given in appendix A.

So the capacity of the CSC equals the capacity of the DMC it contains. Also, even though the channel is not memoryless, the capacity achieving input distribution $P(X^N)$ is memoryless and is the same distribution as for the containing DMC. The *penalty term* $I(X^N; S|Z^N) \leq H(S)$, fits well with the results of the error correcting codes that we will discuss later in this paper.

3 A characterization of the CSC

Using the alternative CSC representation from Figure 2 we can derive a seemingly strange result, namely that the channel outputs are independent of the shift variable and of each other. First we will show that the two views are actually identical. For this we only have to show that the conditional output probabilities, given the input, are the same in both views. For Figure 1 we find for all z^N and all x^N

$$P(Z^N = z^N | X^N = x^N) = \sum_{s \in \mathcal{S}} P(S = s) \prod_{i=1}^{N} P_{\mathrm{DMC}}(z_{\pi_s^{-1} i} | x_i).$$

For Figure 2 we find for all z^N and all x^N, because the DMC is memoryless,

$$P(Z^N = z^N | X^N = x^N) = \sum_{s \in \mathcal{S}} P(S = s) \prod_{i=1}^{N} P_{\mathrm{DMC}}(z_i | x_{\pi_s i}) = \sum_{s \in \mathcal{S}} P(S = s) \prod_{i=1}^{N} P_{\mathrm{DMC}}(z_{\pi_s^{-1} i} | x_i).$$

3.1 Independence of the Shift variable and the output symbols

In appendix B we show that the channel output Z^N is independent of the shift S. This holds under the condition that the underlying channel is memoryless and that the input distribution $P(X^N)$ is memoryless. The channel is memoryless and a memoryless input distribution can achieve capacity so these conditions are satisfied.

We can also show that the Z_i's are independent random variables under the same condition as above. The simple proof is left out due to space limitations.

3.2 Summary of the characterization

The capacity of the CSC equals the capacity of the DMC and is achieved by the same memoryless input distribution $P(X^N)$. The difference in mutual information of the CSC and the DMC is upper bounded by $H(S)$ which is again upper bounded by $\log_A N$.

Although the CSC is not memoryless, the capacity achieving input distribution is memoryless and under that condition also the channel output distribution is memoryless.

4 Coding for the CSC

Due to the asynchronous behavior of the problem setting, any received word will be a shifted version of a distorted code word. So we will have to consider the word synchronization problem.

In stead of associating a message with a single word and prepending a word start indicator sequence (a synchronization sequence), we consider the set of all cyclic shifts of a word as a single code word. The rationale is that we consider short words where reserving a part of the word for synchronization purposes is an absolute waste of transmission rate.

An example will clarify this idea. Assume that the symbol alphabet size $A = 2$.

Let the sequence length $N = 4$. There are $2^N = 16$ possible words but but we can find the following code words or sets, also called *cyclic classes*.

Word 1:	$\{0000\}$	Word 4:	$\{0101, 1010\}$
Word 2:	$\{0001, 0010, 0100, 1000\}$	Word 5:	$\{0111, 1110, 1101, 1011\}$
Word 3:	$\{0011, 0110, 1100, 1001\}$	Word 6:	$\{1111\}$

We wish to count the number of cyclic classes in the set of all sequences of length N given an alphabet of size A. A class is created from a sequence x^N and all it cyclic shifts.

Observation 1. *The number of sequences in a class is not fixed, some classes contain 1 element while other contain more than one element.*

Definition 2. *The* cyclic index *of a sequence x^N is defined as the minimal number of cyclic shifts that bring x^N back to itself.*

We see that the cyclic index of 0001 is 4. On the other hand 0000 has a cyclic index of 1 and 0101 has cyclic index 2.

Observation 2. *All sequences in a class have the same cyclic index. The number of sequences in a class is equal to the cyclic index of a sequence in the class. (This is trivial).*

Definition 3. *The cyclic index of a class is given by the number of sequences in the class.*

Observation 3. *If d is the number of sequences in a class corresponding to a sequence of length N, then d divides n. Or equivalently, the sequence x^N consists of an integer number of repetitions of a sequence y^d, where y^d has a cyclic index of d, i.e. it is not in itself a repetition of shorter sequences.*

Definition 4. *The number of cyclic classes from sequences of length N over an alphabet of size A with cyclic index d is denoted by $W(N, d, A)$.*

The following table lists some example values of $W(N, A)$ for an alphabet of size $A = 3$ and several values of N.

N	$\log_3(W(N,3))/N$	N	$\log_3(W(N,3))/N$
1	1.	11	0.801591
2	0.815465	12	0.811651
3	0.727553	13	0.820408
4	0.723197	14	0.828447
5	0.71578	15	0.835671
$\log_3(W(5000,3))/5000 = 0.998449$			

We see that the size of the set of all classes is not essentially smaller than the set of all words.

Theorem 2. *For any fixed alphabet size A the size of the set of all cyclic classes, $W(N, A)$, is essentially the same as the size of the set of all sequences, in the sense that*

$$\lim_{N\to\infty} \frac{\log_A W(N, A)}{N} = 1. \tag{4}$$

Proof. A cyclic class contains d sequences for any positive integer d that divides N, as we discussed before. So a cyclic class contains at most N words. All possible words belong to exactly one cyclic class so the set of cyclic classes partitions the set of all words. Therefor

$$W(N, A) \geq \frac{A^N}{N}.$$

And obviously we cannot have more cyclic classes than words, so

$$W(N, A) \leq A^N.$$

So

$$\log_A \frac{A^N}{N} = N - \log_A N \leq \log_A W(N,A) \leq \log_A A^N = N.$$
$$\lim_{N\to\infty} \frac{N - \log_A N}{N} = 1 \leq \lim_{N\to\infty} \frac{\log_A W(N,A)}{N} \leq \lim_{N\to\infty} \frac{N}{N} = 1.$$

□

5 A t-error correcting code

We have seen that, due to the asynchrony, code words must at least be cyclic sets. The capacity considerations tell us that the loss of information transfer is described by $H(S) - H(S|X^N, Z^N)$ which is upperbounded by $\log_A N$. This fits well with the choice to use the cyclic classes as code word candidates.

5.1 A Hamming distance for cyclic classes

We wish to find error correcting codes for the CSC. For this reason we define a Hamming distance between the code words (cyclic classes) of possible codes.

Definition 5. *Let m_1 and m_2 be two cyclic classes over the set of sequences of length N over the alphabet $\mathcal{A}$. The* cyclic Hamming distance *$d_{cH}(m_1, m_2)$ is defined as the minimal Hamming distance between any pair of words from $m_1 \times m_2$,*

$$d_{cH}(m_1, m_2) = \min\{d_H(x_1^N, x_2^N) : x_1^N \in m_1, x_2^N \in m_2\}. \tag{5}$$

Here $d_H(x_1^N, x_2^N)$ is the ordinary Hamming distance.

Note that for any two cyclic classes m_1 and m_2, $d_{cH}(m_1, m_2) = d_{cH}(m_2, m_1)$.

Essential for the error correction capabilities of codes for the CSC is the *triangle inequality*. The triangle inequality holds for the ordinary Hamming distance and the next theorem states that the triangle inequality also holds for the cyclic Hamming distance.

Theorem 3. *Let m_1, m_2, and m_3 be three arbitrary cyclic classes of sequences of length N over an alphabet $\mathcal{A}$. The following inequality holds.*

$$d_{cH}(m_1, m_2) \leq d_{cH}(m_1, m_3) + d_{cH}(m_2, m_3). \tag{6}$$

Proof. Let $x_1^N \in m_1$ and $x_2^N \in m_2$ be two sequences with minimal distance, so

$$d_H(x_1^N, x_2^N) = d_{cH}(m_1, m_2). \tag{7}$$

Also let $\tilde{x}_2^N \in m_2$ and $\tilde{x}_3^N \in m_3$ be two sequences with minimal distance, $d_H(\tilde{x}_2^N, \tilde{x}_3^N) = d_{cH}(m_2, m_3)$. There exist a cyclic shift π_j such that $x_2^N = \pi_j \tilde{x}_2^N$. Using this π_j we define $x_3^N = \pi_j \tilde{x}_3^N$. Note that by definition $x_3^N \in m_3$. Because of the fixed cyclic shift it is obvious that

$$d_H(x_2^N, x_3^N) = d_H(\tilde{x}_2^N, \tilde{x}_3^N) = d_{cH}(m_2, m_3). \tag{8}$$

From (7) and (8) and the triangle inequality of the Hamming distance we find

$$d_{cH}(m_1, m_2) \leq d_H(x_1^N, x_2^N) \leq d_H(x_1^N, x_3^N) + d_H(x_2^N, x_3^N) = d_{cH}(m_1, m_3) + d_{cH}(m_2, m_3).$$

□

Definition 6. *For a given cyclic code C containing the cyclic classes $m_1, m_2, \ldots, m_K$ as the K code words, the* minimum cyclic Hamming distance*, $d_{cH,\min}(C)$ is defined as*

$$d_{cH,\min}(C) = \min \{d_{cH}(m_1, m_2) : m_1 \in C, m_2 \in C, m_1 \neq m_2\}.$$

As a corollary we will state that a cyclic code C with minimum cyclic Hamming distance $d_{cH,\min}(C)$ can correct $\left\lfloor \frac{d_{cH,\min}(C)-1}{2} \right\rfloor$ errors.

Theorem 4. *Let m_1 and m_2 be two different code words from C. Let m_3 be an arbitrary cyclic class with distance $d_{cH}(m_1, m_3) = e \leq \left\lfloor \frac{d_{cH,\min}(C)-1}{2} \right\rfloor$. Then*

$$d_{cH}(m_2, m_3) > \left\lfloor \frac{d_{cH,\min}(C) - 1}{2} \right\rfloor. \tag{9}$$

The proof follows directly from Theorem 3.

5.2 Code generation

In a general setting the problem of constructing a good error correcting code is not a simple task. In the setting of a CSC the codes that can be used have short word lengths and an almost exhaustive search for good codes can be used.

An example

Let $N = 5$ and $A = 3$. The next table lists the best codes found for practical CSC parameters N and A.

Table 1: Some example CSC codes

N	A	$d_{cH,\min}$	nr words	code
3	3	2	5	{000, 012, 021, 111, 222}
3	3	3	3	{000, 111, 222}
4	3	2	9	{0000, 0011, 0022, 0101, 0202, 1111, 1122, 1212, 2222}
4	3	3	3	{0000, 0111, 0222}
5	3	2	17	{00000, 00011, 00101, 00122, 00212, 00221, 01022, 01112, 01121, 01202, 01211, 02021, 02111, 02222, 11111, 11222, 12122}
5	3	3	6	{00000, 00121, 01022, 02112, 11111, 22222}
6	3	3	10	{000000, 000111, 001021, 002122, 012012, 020202, 022211, 111111, 121212, 222222}
6	3	4	5	{000000, 001122, 002211, 111111, 222222}

5.3 Asymptotic rate bounds

In this section we shall derive both the direct and the asymptotic Gilbert bound. The Gilbert bound lower bounds the number of code words that a good cyclic code will have.

Theorem 5. *Let C be a cyclic code with word length N over an alphabet of size A and with a minimal cyclic Hamming distance $d_{cH,\min}$. Let $|C|$ denote the numer of words in the code. Now, given N, A, and $d_{cH,\min}$, there must exist a cyclic code C with*

$$|C| \geq \left\lceil \frac{A^N}{N \cdot V(N, A, d_{cH,\min} - 1)} \right\rceil, \tag{10}$$

where $V(N, A, d)$ is the volume of a sphere of radius d in $\mathcal{A}^N$, i.e.

$$\begin{aligned} V(N, A, d) &= \left| \{x^N \in \mathcal{A}^N : d_H(x^n, 0^N) \leq d\} \right| \\ &= \sum_{\delta=0}^{d} \binom{N}{\delta} (A-1)^\delta \end{aligned} \tag{11}$$

Asymptotically, as $N \to \infty$ and $d_{cH,\min} = 2f \cdot N + 1$, so that the fraction of correctable remains constant for the codes $C(N)$ over the alphabet $\mathcal{A}$, we find

$$\lim_{N\to\infty} \frac{\log_A |C(N)|}{N} \geq 1 - h_A(2f), \tag{12}$$

where

$$\begin{aligned} h_A(p) = p \log_A(A-1) - p \log_A p - (1-p) \log_A(1-p), \\ \text{for all } p \in [0, 1 - \frac{1}{A}]. \end{aligned} \tag{13}$$

Note that this is the same bound as the asymptotic Gilbert bound for ordinary error correcting codes.

The proof of (10) is similar to the proof of the standard Gilbert bound when we realize that one code word now can define N spheres and so will use upto $NV(N, A, d_{cH,\min} - 1)$ words. The minimum number of code words needed to cover the complete space is $\left\lceil \frac{A^N}{N \cdot V(N, A, d_{cH,\min} - 1)} \right\rceil$.

It is well-known that the volume of a A-ary sphere is upper bounded as $V(N, A, \delta N) \leq A^{N h_A(\delta)}$. With (10) and this upper bound we find for the codes $C(N)$, assuming that fN is an integer,

$$|C(N)| \geq \frac{A^N}{N \cdot V(N, A, fN - 1)} \geq \frac{A^N}{N \cdot A^{N h_A(2f + 1/N)}}. \tag{14}$$

(12) follows from (14) and the continuity of $h_A(p)$.

Note that again the difference of $\log_A N$ is visible between the Gilbert bound for error correcting codes and cyclic error correcting codes.

6 Conclusion

In this paper we consider efficient communication over an unsynchronized channel. Every word is a shifed and distorted version of the original word. Due to the scenarios we have in mind, we do know the start of a (shifted) word.

While asymptotically this setting is not essentially different from a memoryless channel, the special conditions require a different way of coding for short code words. We introduced a different means to sychronize words and we considered the behaviour of the mutual information for short word lengths and a method for the design of error correcting codes for this purpose.

A Proof of Theorem 1

Proof. First we use the chain rule on $I(X^N; Z^N, S)$ and because X^N and S are independent we have $I(X^N; S) = 0$, so we obtain

$$I(X^N; Z^N) = I(X^N; Z^N|S) - I(X^N; S|Z^N). \tag{15}$$

Because we can determine Z^N from Y^N and S and vice versa and because S is independent of X^N and Y^N, we have $I(X^N; Y^N, Z^N|S) = I(X^N; Y^N)$. But also $I(X^N; Y^N, Z^N|S) = I(X^N; Z^N|S)$. So we can conclude that

$$I(X^N; Z^N|S) = I(X^N; Y^N) \tag{16}$$

From (15) and (16) we derive

$$I(X^N; Z^N) = I(X^N; Y^N) - I(X^N; S|Z^N). \tag{17}$$

Now from $0 \leq I(X^N; S|Z^N) \leq H(S) \leq \log_A N$, we find

$$\lim_{N\to\infty} \frac{1}{N} I(X^N; S|Z^N) = 0, \tag{18}$$

and thus, from (17) and (18), we prove (3).

$$\lim_{N\to\infty} \sup_{P(X^N)} \frac{1}{N} I(X^N; Z^N) = \lim_{N\to\infty} \sup_{P(X^N)} \frac{1}{N} I(X^N; Y^N).$$

□

B Independence of the Shift variable

Because $X^N, S \multimap V^N \multimap Z^N$ form a Markov chain and $V^n = \pi_s X^N$ for given $S = s$

$$I(X^N, S, V^N; Z^N) = I(X^N; Z^N) + I(S; Z^N|X^N), \tag{19}$$

We conclude that

$$I(X^N; Z^N) = I(V^N; Z^N) - I(S; Z^N|X^N). \tag{20}$$

If the probability distribution of V^N is equal to the probability distribution of X^N then $I(X^N; Y^N) = I(V^N; Z^N)$. If $P(X^N)$ is memoryless then this is true and $P(X^N)$ is allowed to be memoryless. But then we find

$$\begin{aligned} I(X^N; S|Z^N) &= I(S; Z^N|X^N) \\ H(S|Z^N) &= H(S|X^N) = H(S) \end{aligned} \tag{21}$$

So we can conclude that, under the conditions that the underlying channel is memoryless and that the input distribution $P(X^N)$ is also memoryless, the output Z^N is independent of the shift S.