Asymptotically Optimal Low-Cost Solid Codes[1]

H. Jürgensen[2], S. Konstantinidis[3], N. H. Lâm[4]

Abstract: A new construction of solid codes is introduced. With this construction one obtains classes of solid codes which are asymptotically optimal in terms of information rate and which have low-cost systematic encoding and decoding agorithms.

The construction uses a new result on the asymptotic average length of maximal runs in words.

Key words: Solid code, information rate, redundancy, insertion error, deletion error, synchronization, runs in word, run length.

1. Introduction

In addition to symbol substitution errors as considered in traditional fault models for digital communication, synchronisation errors are likely to occur in communication systems working at very high speed, with very low signal strengths or under adverse conditions. To deal with synchronization problems in such circumstances one can send special synchronization sequences incurring some loss in speed or bandwidth, or one can utilize the synchroniztion capabilities of certain types of codes.

In this paper we consider *solid codes.* These codes are comma-free and are *error-resistant* in the following sense: Any correctly received code word in a received message will be decoded correctly, that is, errors in the received message will not affect the decoding of the correctly received code words.

Solid codes were introduced in [8] as *strongly regular codes.* They were re-named in [10] into *codes without overlaps.* As *solid codes* they were introduced in [14] using a quite different approach, shown to be equivalent in [4]. Section 11 of [2] contains a survey of work on solid codes as of 1996. Since then, a construction of all maximal finite solid codes over a given alphabet has been obtained in [7], which also provides a recursive enumeration of all maximal regular[5] solid codes. Further properties of maximal solid codes are derived

[1] This research was supported by the Natural Sciences and Engineering Council of Canada.

[2] Department of Computer Science, The University of Western Ontario, London, Ontario, Canada, N6A 5B7; and Institut für Informatik, Universität Potsdam, August-Bebel-Straße 89, 14482 Potsdam, Germany; email: helmut@uwo.ca and helmut@cs.uni-potsdam.de

[3] Department of Mathematics and Computing Science, Saint Mary's University, Halifax, Nova Scotia, Canada, B3H 3C3; email: S.Konstantinidis@StMarys.ca

[4] Hanoi Institute of Mathematics, P.O. Box 631, Bo Ho, 10 000 Hanoi; email: nhlam@thevinh.ncst.ac.vn

[5] In the sense of language theory.

in [1], in particular, a combinatorial characterization of all maximal solid codes contained in the set $a^+b^+ \cup a^+b^+a^+b^+$. For the latter class of solid codes a general method for analysing their information rate (or redundancy) is established in [3].

For a class of codes to be useful in a communication system with synchronization problems at least the following requirements have to be met: (1) re-synchronization must be essentially instantaneous; (2) coding and decoding must be simple; (3) the construction must allow one to control the error-detection and error-correction properties; (4) the information rate must be acceptable. With these constraints in mind, we explore the construction of a new class of solid codes as follows: Starting from a code L with known properties we surround each word in L by a protective pair of of strings, its shield, turning L into a solid code L', which inherits at least some of the properties of L.

The shield construction is based on evaluating maximal run lengths in words. As an auxiliary result, which is of interest in its own right, we determine the asymptotics of the average maximal run lengths in words.

We then apply this construction to the important special case of L being a block code. When L is a full block code of length n then the shields can be defined in such a way that the information rate of L' is asymptotically equal to 1 as $n \to \infty$. For a block code L detecting single substitution errors, the solid code L' detects single substitution, insertion or deletion errors.

Encoding and decoding with respect to L', assuming L is a finite code, can be achieved using finite-state transducers concatenated with an encoder and decoder, respectively, for L. Thus the overhead arising from the use of L' is quite small.

Our paper is structured as follows: Following this introduction, Section 1 introduces notation and reviews some basic notions. The construction of the shields uses maximal runs of symbols in words; these are studied in Section 3. Section 4 contains the general construction of solid codes from infix codes using run-length-based shields. In Section 5 we review known results regarding the information rate of solid codes. This prepares the ground for the analysis of the asymptotic information rate of solid codes contructed from full block codes in Section 6. In Section 7, we study error-detection capabilities of solid codes constructed from block codes. Some conclusions are provided in Section 8.

2. Notions and Notation

We introduce the notation used and briefly review some notions from the theory of codes as needed in the sequel. We use [2] as a general reference for the theory of codes, [12] for the theory of block codes[6], [15] for synchronization issues in communication, and [13] for topics in the theory of formal languages.

The symbol $\mathbb{N}$ denotes the set of positive integers, $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$ and $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$. An *alphabet* is a finite non-empty set the elements of which are called *symbols.* In this paper the case of alphabets with only a single symbol is trivial. We assume, therefore, that an alphabet contains at least two distinct symbols, which, for convenience, we call a and b.

Let X be an alphabet. Then X^* denotes the set of *words over* X, that is, the set of finite sequences of symbols in X including the *empty word* ε. The set X^* is a free monoid with respect to the concatenation of words as product. As usual, $X^+ = X^* \setminus \{\varepsilon\}$. The

[6] For the little of this theory used in this paper almost any other book on the topic would also be sufficient.

length of a word $w \in X^*$ is the number of symbols in w denoted by $|w|$, that is, $|\varepsilon| = 0$ and $|wx| = |w| + 1$ for $w \in X^*$ and $x \in X$. For $n \in \mathbb{N}_0$, X^n denotes the set of all words of length n over X. If w is a word then w^R is the *reverse* of w.

A *language* over X is a subset of X^*. For languages L, L' over X and $i \in \mathbb{N}_0$,

$$LL' = \{w \mid \exists u \in L\, \exists v \in L' : w = uv\},$$
$$L^i = \begin{cases} \{\varepsilon\}, & \text{if } i = 0, \\ L^{i-1}L, & \text{if } i > 0, \end{cases}$$
$$L^R = \{w^R \mid w \in L\}$$

and

$$L^+ = \bigcup_{i=1}^{\infty} L^i.$$

The language L is a *code* if L^+ is a free semigroup with L as a set of free generators, that is, if every word in L^+ has exactly one factorization over L.

On X^* we consider the following binary relations, letting $u, v \in X^*$:

- $u \leq_{\mathrm{p}} v$ if $v \in uX^*$, that is, u is a *prefix of* v. The word u is a *proper prefix* of v if $u \leq_{\mathrm{p}} v$ and $\varepsilon \neq u \neq v$.
- $u \leq_{\mathrm{s}} v$ if $v \in X^*u$, that is, u is a *suffix of* v. The word u is a *proper suffix* of v if $u \leq_{\mathrm{s}} v$ and $\varepsilon \neq u \neq v$.
- $u \leq_{\mathrm{i}} v$ if $v \in X^*uX^*$, that is, u is an *infix of* v. The word u is a *proper infix* of v if $u \leq_{\mathrm{i}} v$ and $\varepsilon \neq u \neq v$.
- $u\ \omega_{\mathrm{ol}}\ v$ if there exist $u_1, u_2, v_1 \in X^+$ such that $u = u_1u_2$ and $v = v_1u_1$, that is, u and v have an *overlap*.

When considering non-empty sets of incomparable elements with respect to the these relations one obtains certain classes of languages[7]:

- If $L \in X^+$ is such that, for any $u, v \in L$, $u \leq_{\mathrm{p}} v$ implies $u = v$, then L is a *prefix code*.
- If $L \in X^+$ is such that, for any $u, v \in L$, $u \leq_{\mathrm{s}} v$ implies $u = v$, then L is a *suffix code*.
- If $L \in X^+$ is such that, for any $u, v \in L$, $u \leq_{\mathrm{i}} v$ implies $u = v$, then L is an *infix code*.
- If $L \in X^+$ is such that $u\ \omega_{\mathrm{ol}}\ v$ does not hold for any $u, v \in L$ then L is *overlap-free*.
- A *solid code* is an overlap-free infix code.
- A *uniform code*[8] is a subset of X^n for some $n \in \mathbb{N}$. A uniform code $L \subseteq X^n$ is said to be a *full uniform code* if $L = X^n$.

Prefix codes, suffix codes, infix codes, uniform codes and solid codes are, indeed, codes.

For the construction of solid codes we need the notion of *near-inverses* of integer functions from [3].

For $m \in \mathbb{N}_\infty$, let $I_m = \{i \mid i \in \mathbb{N}, i < m\}$. Thus, $I_1 = \emptyset$. As usual, let $\infty + 1 = \infty$. Consider a monotonically increasing function[9] $f : I_m \to \mathbb{N}$. When $m = 1$ then f is the

[7] See [2] for details.

[8] In the theory of error-correcting codes uniform codes are usually called block codes.

[9] The function f need not be strictly monotonically increasing.

empty function. Define $\lim f$ as

$$\lim f = \begin{cases} 0, & \text{if } f \text{ is empty,} \\ \infty, & \text{if } f \text{ is unbounded,} \\ p, & \text{if } f \text{ is bounded by } p \in \mathbb{N} \text{ and } f(j) = p \text{ for some } j \in I_m. \end{cases}$$

Definition 2.1 Let $m, n \in \mathbb{N}_\infty$ and let $f : I_m \to \mathbb{N}$ and $g : I_n \to \mathbb{N}$ be monotonically increasing functions. Then f and g are said to be (m, n)-*near-inverses of each other* if the following conditions are satisfied:

(1) If $n = \infty$, then $\lim g = m$.

(2) If $m = \infty$, then $\lim f = n$.

(3) For all $i \in I_n$,

$$g(i) = \begin{cases} \min\{j \mid j \in I_m, f(j) > i\}, & \text{if such a } j \text{ exists,} \\ m, & \text{otherwise.} \end{cases}$$

(4) For all $j \in I_m$,

$$f(j) = \begin{cases} \min\{i \mid i \in I_n, g(i) > j\}, & \text{if such an } i \text{ exists,} \\ n, & \text{otherwise.} \end{cases}$$

Let F_i and G_j be the sets used in conditions (3) and (4) in Definition 2.1, that is,

$$F_i = \{j \mid j \in I_m, f(j) > i\}$$

and

$$G_j = \{i \mid i \in I_n, g(i) > j\}.$$

The conditions (3) and (4) could, at a first glance, lead to $g(i) = \infty$ or $f(j) = \infty$ in some cases, which is, of course, not allowable. One verifies that this cannot happen [3].

Proposition 2.1 [3] *Let $m, n \in \mathbb{N}_\infty$.*

(1) If f is a monotonically increasing function, $f : I_m \to \mathbb{N}$, bounded by n and such that $\lim f = n$ when $m = \infty$, then there is a unique monotonically increasing function $g : I_n \to \mathbb{N}$ such that f and g are (m, n)-near-inverses of each other.

(2) If g is a monotonically increasing function, $g : I_n \to \mathbb{N}$, bounded by m and such that $\lim g = m$ when $n = \infty$, then there is a unique monotonically increasing function $f : I_m \to \mathbb{N}$ such that f and g are (m, n)-near-inverses of each other.

(3) If f and g are (m, n)-near-inverses of each other then g and f are (n, m)-near-inverses of each other.

Let $m, n \in \mathbb{N}_\infty$ and let $f : I_m \to \mathbb{N}$ be a mapping. We say that the condition $S(f, m, n)$ is satisfied if and only if f is monotonically increasing, bounded by n, and $\lim f = n$ when $m = \infty$. In this case, let $g_{f,m,n}$ be the near-inverse function of f as determined by Proposition 2.1(1).

For the purposes of this paper[10] a *channel* is a binary relation γ on X^*. The fact that $(v, u) \in \gamma$ is interpreted to mean that the output of the channel γ could be v when its input is u. A channel is *error-free* if $(v, u) \in \gamma$ if and only if $u = v$. The errors that we consider are symbol substitutions, insertions or deletions, denoted by σ, ι and δ, respectively. A channel that could have any k of these errors altogether in any l consecutive symbols[11] is denoted by $\sigma \odot \iota \odot \delta(k, l)$. Of course this model of a channel abstracts from the true situation in that it ignores highly improbable events. For a channel γ and a word $w \in X^*$, let $\langle w \rangle_\gamma$ be the set of words w' such that $(w', w) \in \gamma$, that is, w' is a possible output for input w.

Let γ be a channel. A language $L \subseteq X^+$ is *error-detecting* for γ if, for all $w, w' \in L \cup \{\varepsilon\}$, $w' \in \langle w \rangle_\gamma$ implies $w' = w$. The language L is $(\gamma, 1)$-*detecting* if, for all $w \in L \cup \{\varepsilon\}$ and for all $w \in L^*$, $w' \in \langle w \rangle_\gamma$ implies $w' = w$. The language L is $(\gamma, *)$-*detecting* if L^* is error-detecting for γ.

3. Maximal Runs in Words

In this section we determine asymptotic formulæ for maximal runs in words. We consider the alphabet $X = \{a, b\}$. A word $w \in X^+$. has the form

$$w = a^{\alpha_1} b^{\beta_1} a^{\alpha_2} b^{\beta_2} \cdots a^{\alpha_r} b^{\beta_r}$$

for some $r \in \mathbb{N}$, $\alpha_2, \ldots, \alpha_r, \beta_1, \ldots, \beta_{r-1} \in \mathbb{N}$, and $\alpha_1, \beta_r \in \mathbb{N}_0$. Each of the words a^{α_i} and b^{β_i} with $1 \leq i \leq r$ is called a *run* of w. The $2r$-tuple

$$\tau(w) = (\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_r, \beta_r)$$

is the *run pattern* of w.

For $x \in X$, let ξ^x be α if $x = a$ and β if $x = b$. A run $x^{\xi_i^x}$ of w has *length* ξ_i^x. A *maximal run* of w is a run $x^{\xi_i^x}$ of w the length of which is maximal. Let $\bar{\tau}(w)$ be the length of a maximal run of w. Moreover, let $\bar{\tau}_x(w)$ be the maximal length of a run of x in w. In particular, $\bar{\tau}(\varepsilon) = \bar{\tau}_x(\varepsilon) = 0$.

The definitions of $\bar{\tau}(w)$ and $\bar{\tau}_x(w)$ can be extended to alphabets with more than two letters in an obvious way. In the sequel we occasionally assume this generalization without special mention.

For $n \in \mathbb{N}$ consider the set F_n of words in X^n such that, for $w \in F_n$, the first run of w is not maximal. Let $f_n = |F_n|$. We determine $\lim_{n\to\infty} f_n/2^n$.

---

[10] A more general definition involving also infinite words is provided in [2], which specializes to the one used here.

[11] See [5] for details.

Proposition 3.1 *The sequence $f_n/2^n$ converges to 1 with*

$$f_n/2^n < f_{n+1}/2^{n+1} < 1.$$

Proof: For $i, j \in \mathbb{N}$, let $C_i(j)$ be the set of words in X^j the first run of which is maximal and of length i; Let $c_i(j) = |C_i(j)|$. Clearly, $c_i(j) = 0$ for $j < i$. Moreover, $C_i(i) = \{a^i, b^i\}$ and, hence, $c_i(i) = 2$ for all i.

Consider a word $w \in F_{n+1}$ with

$$\tau(w) = (\alpha_1, \beta_1, \ldots, \alpha_r, \beta_r)$$

as its run pattern. Then $w = w'a$ or $w = w'b$ for some word w' and either

- $w' \in F_n$ or
- w' is a word the first and last runs of which are both maximal and of length i, say.

In the latter case w' has the form $w' = a^i b w''$ or $w' = b^i a w''$ with $|bw''| = n - i$ or $|aw''| = n - i$, respectively; moreover, the last run of bw'' or aw'' also has length i. As this last run is maximal, the first run of the reverse word is maximal and, hence, the reverse word is in $C_i(n-i)$. Note that i can be at most $\lfloor n/2 \rfloor$. This establishes

$$f_{n+1} = 2f_n + \sum_{i=1}^{\lfloor n/2 \rfloor} c_i(n-i).$$

Thus

$$\frac{f_{n+1}}{2^{n+1}} = \frac{f_n}{2^n} + \sum_{i=1}^{\lfloor n/2 \rfloor} \frac{c_i(n-i)}{2^{n+1}}.$$

Note that $f_1 = 0$. Therefore,

$$\frac{f_{n+1}}{2^{n+1}} = \sum_{j=2}^{n} \sum_{i=1}^{\lfloor j/2 \rfloor} \frac{c_i(j-i)}{2^{j+1}}.$$

From this is follows that the sequence $f_n/2^n$ converges as it is bounded and monotonically increasing. Moereover,

$$\begin{aligned}
\lim_{n\to\infty} \frac{f_n}{2^n} &= \sum_{j=2}^{\infty} \sum_{i=1}^{\lfloor j/2 \rfloor} \frac{c_i(j-i)}{2^{j+1}} \\
&= \quad c_1(1)/2^3 \\
&\quad + c_1(2)/2^4 \\
&\quad + c_1(3)/2^5 + c_2(2)/2^5 \\
&\quad + c_1(4)/2^6 + c_2(3)/2^6 \\
&\quad + c_1(5)/2^7 + c_2(4)/2^7 + c_3(3)/2^7 \\
&\quad \cdots \\
&= \sum_{i=1}^{\infty} \sum_{j=0}^{\infty} c_i(i+j)/2^{2i+j+1}.
\end{aligned}$$

Let

$$t_i = \sum_{j=0}^{\infty} c_i(i+j)/2^{2i+j+1}.$$

Then

$$\lim_{n\to\infty} \frac{f_n}{2^n} = \sum_{i=1}^{\infty} t_i.$$

Now consider $C_i(n)$. Let $D_i(n)$ be the set of words of length n having a maximal run of length i and let $d_i(n) = |D_i(n)|$. Removing the first run from each word in $C_i(n)$ results in the disjoint union of the sets $D_1(n-i), \ldots, D_i(n-i)$ and, therefore,

$$c_i(n) = \sum_{l=1}^{i} d_l(n-i).$$

Division by 2^{n+i+1} yields

$$\frac{c_i(n)}{2^{n+i+1}} = \sum_{l=1}^{i} \frac{d_l(n-i)}{2^{n+i+1}}.$$

Therefore,

$$\begin{aligned} t_i - \frac{c_i(i)}{2^{2i+1}} &= t_i - \frac{1}{2^{2i}} \\ &= \sum_{j=1}^{\infty} c_i(i+j)/2^{2i+j+1} \\ &= \sum_{j=1}^{\infty} \sum_{l=1}^{i} d_l(j)/2^{2i+j+1} \\ &= \sum_{l=1}^{i} \left(\sum_{j=1}^{\infty} \frac{1}{2^{2i-l}} \cdot \frac{d_l(j)}{2^{j+l+1}} \right) \end{aligned}$$

using the facts that $c_i(i) = d_i(i) = 2$ and that $c_i(j) = d_i(j) = 0$ for $j < i$. Define

$$\delta_l = \sum_{j=1}^{\infty} \frac{d_l(j)}{2^{j+l+1}}.$$

Then

$$t_i - \frac{c_i(i)}{2^{2i+1}} = \sum_{l=1}^{i} \frac{\delta_l}{2^{2i-l}}.$$

Now consider a word $w \in D_i(n)$. Its first run may or may not be of length i. There are $d_i(n-t)$ words in $D_i(n)$ the first run of which has a length of $t < i$. If the first run of w

has length i consider $w' \in X^{n-i}$ such that $w = a^i w'$ or $w = b^i w'$. The maximal run length in w' is any number from 1 to i. Thus

$$d_i(n) = d_i(n-1) + \ldots + d_i(n-i-1) + d_1(n-i) + \ldots + d_i(n-i).$$

Then

$$\begin{aligned}\frac{d_i(n)}{2^{n+i+1}} &= \sum_{l=1}^{i-1}\left(\frac{1}{2^l}\cdot\frac{d_i(n-l)}{2^{n-l+i+1}}\right) + \sum_{l=0}^{i-1}\left(\frac{1}{2^{i+l}}\cdot\frac{d_{i-l}(n-i)}{2^{n-l+1}}\right)\\ &= \sum_{l=1}^{i}\left(\frac{1}{2^l}\cdot\frac{d_i(n-l)}{2^{n-l+i+1}}\right) + \sum_{l=1}^{i-1}\left(\frac{1}{2^{i+l}}\cdot\frac{d_{i-l}(n-i)}{2^{n-l+1}}\right).\end{aligned}$$

As

$$\delta_i = \sum_{j=1}^{\infty}\frac{d_i(j)}{2^{j+i+1}}$$

one has

$$\begin{aligned}\delta_i - \frac{d_i(i)}{2^{2i+1}} &= \delta_i - \frac{1}{2^{2i}} =\\ &= \sum_{n=i+1}^{\infty}\frac{d_i(n)}{2^{n+i+1}}\\ &= \sum_{l=1}^{i}\left(\frac{1}{2^l}\cdot\sum_{n=i+1}^{\infty}\frac{d_i(n-l)}{2^{n-l+i+1}}\right) + \sum_{l=1}^{i-1}\left(\frac{1}{2^{i+l}}\cdot\sum_{n=i+1}^{\infty}\frac{d_{i-l}(n-i)}{2^{n-l+1}}\right).\end{aligned}$$

As $d_i(j) = 0$ for $j < i$, the first sum is equal to

$$\sum_{l=1}^{i}\left(\frac{1}{2^l}\sum_{n=i}^{\infty}\frac{d_i(n)}{2^{n+i+1}}\right) = \delta_i\cdot\sum_{l=1}^{i}\frac{1}{2^l} = (1-2^{-i})\delta_i$$

and the second sum is equal to

$$\sum_{l=1}^{i-1}\left(\frac{1}{2^{i+l}}\cdot\sum_{n=i-l}^{\infty}\frac{d_{i-l}(n)}{2^{n-l+i+1}}\right) = \sum_{l=1}^{i-1}\left(\delta_{i-l}\cdot\frac{1}{2^{i+l}}\right).$$

Thus,

$$\delta_i - \frac{1}{2^{2i}} = \delta_i\cdot(1-2^{-i}) + \sum_{l=1}^{i-1}\left(\delta_{i-l}\cdot\frac{1}{2^{i+l}}\right),$$

hence

$$\frac{\delta_i}{2^i} = \frac{1}{2^{2i}} + \sum_{l=1}^{i-1}\frac{\delta_{i-l}}{2^{i+l}} = \frac{1}{2^{2i}} + \sum_{l=1}^{i-1}\frac{\delta_l}{2^{2i-l}}.$$

Let

$$f = \lim_{n\to\infty} \frac{f_n}{2^n}$$

and recall that

$$f = \sum_{k=1}^{\infty} t_k.$$

From

$$t_k - \frac{1}{2^{2k}} = \sum_{l=1}^{k} \frac{\delta_l}{2^{2k-l}}$$

one computes

$$\begin{aligned} f - \frac{1}{3} &= \sum_{k=1}^{\infty} t_k - \sum_{k=1}^{\infty} \frac{1}{2^{2k}} = \sum_{k=1}^{\infty} \sum_{l=1}^{k} \frac{\delta_l}{2^{2k-l}} \\ &= \delta_1 \cdot \sum_{k=1}^{\infty} \frac{1}{2^{2k-1}} + \sum_{l=2}^{\infty} \left(\delta_l \cdot \sum_{k=l}^{\infty} \frac{1}{2^{2k-l}} \right) \\ &= \frac{\delta_1}{2} \cdot \sum_{k=0}^{\infty} \frac{1}{2^{2k}} + \sum_{l=2}^{\infty} \left(\delta_l \cdot \sum_{k=1}^{\infty} \frac{1}{2^{2k} \cdot 2^{l-2}} \right) \\ &= \frac{2}{3} \cdot \delta_1 + \frac{1}{3} \cdot R \end{aligned}$$

where

$$\begin{aligned} R &= \sum_{l=2}^{\infty} \frac{\delta_l}{2^{l-2}} \\ &= \sum_{k=2}^{\infty} \frac{1}{2^{2k-2}} + \delta_1 \cdot \sum_{k=2}^{\infty} \frac{1}{2^{2k-3}} + \sum_{l=2}^{\infty} \left(\delta_l \cdot \sum_{k=l+1}^{\infty} \frac{1}{2^{2k-l-2}} \right) \\ &= \frac{1}{3} + \frac{2}{3} \cdot \delta_1 + \sum_{l=2}^{\infty} \left(\delta_l \cdot \sum_{k=1}^{\infty} \frac{1}{2^{2k} \cdot 2^{l-2}} \right) \\ &= \frac{1}{3} + \frac{2}{3} \cdot \delta_1 + \frac{1}{3} \cdot R. \end{aligned}$$

Observe that $d_1(n) = 2$ for all n and hence

$$\delta_1 = \sum_{n=1}^{\infty} \frac{d_1(n)}{2^{n+2}} = \frac{1}{2}.$$

Therefore,

$$R = \frac{2}{3} + \frac{1}{3} \cdot R,$$

that is, $R = 1$. Thus also $f = 1$. □

Thus the *asymptotic ratio of words with non-maximal first run* is equal to 1. The next quantity to be considered is the *average maximal run length of words of length n,* that is, the quantity

$$\bar{\tau}_n = \frac{1}{2^n} \sum_{w \in X^n} \bar{\tau}(w).$$

Using the fact that, for $x \in X$ and $w \in X^n$,

$$\bar{\tau}(xw) = \begin{cases} 1 + \bar{\tau}(w), & \text{if } w \in xX^* \setminus F_n, \\ \bar{\tau}(w), & \text{otherwise,} \end{cases}$$

one finds

$$\begin{aligned} \bar{\tau}_{n+1} &= \bar{\tau}_n + \frac{1}{2^{n+1}} \cdot (2^n - f_n) \\ &= \bar{\tau}_n + \frac{1}{2} - \frac{f_n}{2^{n+1}}. \end{aligned}$$

The initial value is

$$\bar{\tau}_1 = 1.$$

Thus

$$\bar{\tau}_{n+1} = 1 + \frac{n}{2} - \frac{1}{2} \cdot \sum_{i=1}^{n} \frac{f_i}{2^i}.$$

To determine the asymptotics we consider the *average maximal run length ratio* $\bar{\tau}_n/n$. Thus

$$\begin{aligned} \frac{\bar{\tau}_n}{n} &= \frac{1}{2n} + \frac{1}{2} - \frac{1}{2n} \cdot \sum_{i=1}^{n-1} \frac{f_i}{2^i} \\ &= \frac{1}{2n} + \frac{1}{2} - \frac{n-1}{2n} \cdot \left(\frac{1}{n-1} \cdot \sum_{i=1}^{n-1} \frac{f_i}{2^i} \right). \end{aligned}$$

Using the fact that

$$\lim_{n \to \infty} \left(\frac{1}{n-1} \cdot \sum_{i=1}^{n-1} \frac{f_i}{2^i} \right) = \lim_{n \to \infty} \frac{f_n}{n} = 1$$

one obtains

$$\lim_{n \to \infty} \frac{\bar{\tau}_n}{n} = 0.$$

This proves the following statement.

Theorem 3.1 *The average maximal run length ratio is*

$$\frac{\bar{\tau}_n}{n} = \frac{1}{2n} + \frac{1}{2} - \frac{1}{2n} \cdot \sum_{i=1}^{n-1} \frac{f_i}{2^i}.$$

Its limit is 0.

For $x, y \in X \cup \{\varepsilon\}$ and $n \in \mathbb{N}$, let

$$\bar{\tau}_n^{x,y} = \frac{1}{2^n} \sum_{w \in X^n} \bar{\tau}(xwy).$$

Thus $\bar{\tau}_n^{\varepsilon,\varepsilon} = \bar{\tau}_n$.

Corollary 3.1 *For all $x, y \in X^*$ one has $\lim_{n\to\infty} \bar{\tau}_n^{x,y}/n = 0$.*

Proof: For any $w \in X^n$,

$$\bar{\tau}(w) \leq \bar{\tau}(xwy) \leq \bar{\tau}(w) + |xy|.$$

Moreover, for n large enough, there are words w such that the second inequality is strict. Therefore,

$$\frac{\bar{\tau}_n}{n} \leq \frac{\bar{\tau}_n^{x,y}}{n} < \frac{\bar{\tau}_n}{n} + \frac{|xy|}{n}$$

for almost all n. Thus

$$\lim_{n\to\infty} \frac{\bar{\tau}_n^{x,y}}{n} = \lim_{n\to\infty} \frac{\bar{\tau}_n}{n} = 0.$$

□

4. Solid Codes Constructed Using Maximal Runs

In this section we provide a construction of a vast class of solid codes from infix codes using maximal runs in words. The construction is based on the following idea: One starts with an infix code L. Each word of L is then prefixed by a sufficiently long sequence of a's followed by a symbol different from a and suffixed by a symbol different from b and a sufficiently long sequence of b's. This construction preserves the property of being an infix code; moreover the long prefix with endmarker and the long suffix with start marker guarantee overlap-freeness. The length of these added prefixes and suffixes is determined from the lengths of maximal runs in the word.

In the next section we show – as an aside – that one marker can be omitted in the special case when L is a full uniform code.

Consider a function $f : \mathbb{N} \to \mathbb{N}$ satisfying $S(f, \infty, \infty)$, that is, f is monotonically increasing with $\lim f = \infty$. Let $\hat{f}$ be its near-inverse, $\hat{f} = g_{f,\infty,\infty}$. For a non-empty finite sequence $s \in \mathbb{N}^+$, $s = (s_1, s_2, \ldots, s_{|s|})$, let $\max s = \max\{s_i \mid i = 1, \ldots, |s|\}$. We extend f and $\hat{f}$ to $\mathbb{N}^+$ by

$$f(s) = f(\max s) \text{ and } \hat{f}(s) = \hat{f}(\max s).$$

Lemma 4.1 *With f and $\hat{f}$ as above,*

$$f(s) > \max t \text{ or } \hat{f}(t) > \max s$$

for all $s, t \in \mathbb{N}^+$.

Proof: By definition

$$\hat{f}(t) = \min\{j \mid f(j) > \max t\}.$$

Hence, if $f(s) \leq \max t$ then $\hat{f}(t) > \max s$. □

In particular, when applied to run lengths in words, Lemma 4.1 implies that, for $w \in X^+$ one has

$$f(\bar{\tau}_a(w)) > \bar{\tau}_b(w) \text{ or } \hat{f}(\bar{\tau}_b(w)) > \bar{\tau}_a(w).$$

We use the following observation.

Remark 4.1 *If $L \subseteq X^+$ is an infix code then also $(X \setminus \{a\})L(X \setminus \{b\})$ is an infix code.*

Proof: Let $L' = (X \setminus \{a\})L(X \setminus \{b\})$ and suppose that L' is not an infix code. Then there are words $w_1, w_2 \in L'$ such that $w_1 \leq_{\mathrm{i}} w_2$ and $w_1 = x_1 v_1 y_1$, $w_2 = x_2 v_2 y_2$ with $x_1, x_2 \in X \setminus \{a\}$, $v_1, v_2 \in L$ and $y_1, y_2 \in X \setminus \{b\}$. This implies that $v_1 \leq_{\mathrm{i}} v_2$, hence $v_1 = v_2$ as L is an infix code. As a consequence, $w_1 = w_2$. □

Theorem 4.1 *Let $L \subseteq X^+$ be an infix code, let $f : \mathbb{N} \to \mathbb{N}$ be a function satisfying $S(f, \infty, \infty)$ and let $\hat{\tau}_a, \hat{\tau}_b : X^* \to \mathbb{N}_0$ such that*

$$\hat{\tau}_a(w) \geq \bar{\tau}_a(w) \text{ and } \hat{\tau}_b(w) \geq \bar{\tau}_b(w)$$

for all $w \in X^*$.

Then the language

$$K_{f,L,\hat{\tau}_a,\hat{\tau}_b} = \{a^{f(\hat{\tau}_b(w))} w b^{\hat{f}(\hat{\tau}_a(w))} \mid w \in (X \setminus \{a\})L(X \setminus \{b\})\}$$

is a solid code.

Proof: To simplify the notation, let $L' = (X \setminus \{a\})L(X \setminus \{b\})$ and let $\gamma(u) = a^{f(\hat{\tau}_b(u))} u b^{\hat{f}(\hat{\tau}_a(u))}$ for any $u \in L'$.

First, assume that $K_{f,L,\hat{\tau}_a,\hat{\tau}_b}$ is not an infix code. Then there exist $u, v \in L'$ such that $\gamma(u) \leq_{\mathrm{i}} \gamma(v)$ and $\gamma(u) \neq \gamma(v)$. It follows that $u \leq_{\mathrm{i}} \gamma(v)$. As u starts with a symbol different from a, the occurence of u lies completely outside $a^{f(\hat{\tau}_b(u))}$; similarly, as u ends with a symbol different from b, its occurence lies completely outside $b^{\hat{f}(\hat{\tau}_a(u))}$. Thus, $u \leq_{\mathrm{i}} v$. By Remark 4.1 L' is an infix code. Therefore, $u = v$ contradicting the assumption that $\gamma(u)$ and $\gamma(v)$ are different.

Next assume that $K_{f,L,\hat{\tau}_a,\hat{\tau}_b}$ is not overlap-free. Then there exist $u, v \in L'$ – not necessarily distinct – such that a proper prefix p of $\gamma(u)$ is a proper suffix of $\gamma(v)$ and p has the form $a^{f(\hat{\tau}_b(u))} w b^{\hat{f}(\hat{\tau}_a(v))}$. We distinguish four cases.

Case 1: $p = a^{f(\hat{\tau}_b(u))} u b^k = a^l v b^{\hat{f}(\hat{\tau}_a(v))}$ with $k < \hat{f}(\hat{\tau}_a(u))$ and $l < f(\hat{\tau}_b(v))$. Then $u = v$, hence $\gamma(u) = \gamma(v)$ and $f(\hat{\tau}_b(u)) = f(\hat{\tau}_b(v)) = l$, contradiction.

Case 2: $p = a^{f(\hat{\tau}_b(u))} u b^k = a^l v' b^{\hat{f}(\hat{\tau}_a(v))}$ where v' is a proper suffix of v with $k < \hat{f}(\hat{\tau}_a(u))$ and $l \leq \hat{\tau}_a(v)$. Then u is a proper suffix of v, contradicting the fact that L' is an infix code.

Case 3: $p = a^{f(\hat{\tau}_b(u))} u' b^k = a^l v b^{\hat{f}(\hat{\tau}_a(v))}$ where u' is a proper prefix of u with $k \leq \hat{\tau}_b(u)$ and $l < f(\hat{\tau}_b(v))$. Then $u' = v$, that is, v is a proper prefix of u, contradicting the fact that L' is an infix code.

Case 4: $p = a^{f(\hat{\tau}_b(u))} u' b^k = a^l v' b^{\hat{f}(\hat{\tau}_a(v))}$ where u' is a proper prefix of u, v' is a proper suffix of v and $k \leq \hat{\tau}_b(u)$ and $l \leq \hat{\tau}_a(v)$. Then $f(\hat{\tau}_b(u)) \leq \hat{\tau}_a(v)$ and $\hat{f}(\hat{\tau}_a(v)) \leq \hat{\tau}_b(u)$ contradicting Lemma 4.1. □

In the next section, we determine the information rate of codes of the form $K_{f,L,\hat{\tau}_a,\hat{\tau}_b}$ for the alphabet $X = \{a, b\}$ and some specific choices of the parameters:

- Codes K_n for $n \in \mathbb{N}$: $L = X^n$, $\hat{\tau}_a(w) = \hat{\tau}_b(w) = \bar{\tau}(w)$ for all $w \in X^*$ and $f(k) = k$, hence $\hat{f}(k) = k + 1$, for all $k \in \mathbb{N}$.
- Codes $\hat{K}_n$ for $n \in \mathbb{N}$: $L = X^n$, $\hat{\tau}_a(w) = \bar{\tau}_a(w)$ and $\hat{\tau}_b(w) = \bar{\tau}_b(w)$ for all $w \in X^*$, and $f(k) = k$, hence $\hat{f}(k) = k + 1$, for all $k \in \mathbb{N}$.

It turns out that the asymptotical information rate of these codes is optimal.

5. Information Rate and Redundancy of Languages

An important parameter for assessing the efficiency of a code is its information rate – or its redundancy. We define these notions for the case of $|X| = 2$. The definitions easily generalize to $|X| > 2$ by changing the base of the logarithms to $|X|$ instead of base 2.

Consider a finite, non-empty code $L \subseteq X^+$. Let

$$s(L) = \sum_{w \in L} |w|$$

be the sum of the word lengths of words in L and

$$\hat{s}(L) = \max\{|w| \mid w \in L\}$$

be the maximal word length of L. The *information rate* of L is

$$r(L) = \frac{\log |L|}{\frac{s(L)}{|L|}} = \frac{|L| \cdot \log |L|}{s(L)}.$$

One has $0 \leq r(L) \leq 1$. The *minimal information rate* of L is

$$\hat{r}(L) = \frac{\log |L|}{\hat{s}(L)}.$$

One has $0 \leq \hat{r}(L) \leq r(L) \leq 1$.

For an infinite language $L \subseteq X^+$ we need to adjust this definition. We assume that L is ordered by length, that is,

$$L = \{w_1, w_2, \ldots\}$$

with $|w_i| \leq |w_{i+1}|$ for $i = 1, 2, \ldots$ Let

$$L_n = \{w_1, w_2, \ldots, w_n\}$$

for $n \in \mathbb{N}$. The *information rate* and the *minimal information rate* of L are defined as

$$r_n(L) = r(L_n) \text{ and } \hat{r}_n(L) = \hat{r}(L_n),$$

respectively.

The *per-symbol-redundancy* of L is $1 - \hat{r}(L)$ or $1 - \hat{r}_n(L)$, respectively. The notion of redundancy as used in [9, 10, 2] is that of absolute redundancy. The per-symbol-redundancy is asymptotically equal to the absolute redundancy divided by the largest word length.

By translating results of [9, 10, 11, 3] we recall the following facts.

Theorem 5.1 *Let $X = \{a, b\}$.*

(1) For almost all $n \in \mathbb{N}$ and all solid codes in $L \subseteq X^n$ one has

$$r(L) < 1 - \frac{\log(n-1)}{n} - \frac{\log e}{n}.$$

(2) For almost all $n \in \mathbb{N}$ there is a solid code $L \subseteq X^n$ with

$$r(L) \gtrsim 1 - \frac{\log n}{n} - \frac{2 + \log\log e}{n}.$$

(3) There are constants $c_1, c_2 \in \mathbb{R}$ with $c_1 > 1$ such that, for every infinite solid code $L \subseteq X^+$, one has

$$\hat{r}_n(L) < \frac{1}{c_1 + \frac{c_2}{\log n}}$$

for almost all n.

(4) There is an infinite overlap-free prefix code $L \subseteq X^+$ such that

$$\hat{r}_n(L) \approx \frac{1}{1 + \frac{\frac{3}{2}\log\log n}{\log n}}.$$

(5) For every infinite solid code $L \subseteq a^+b^+a^+b^+$ one has

$$\hat{r}_n(L) \lesssim \frac{\log n}{\sqrt{2n}}.$$

Asymptotically, the bounds of Theorem 5.1(1-2) are equal to 1. The bound of (3) converges to $1/c_1 < 1$; the size of c_1 is not known. For (4), one has $\hat{r}_n(L) \approx r_n(L) \to 1$. In the case of (5), $\hat{r}_n(L)$ converges to 0; however, the asymptotic behaviour of $r_n(L)$ is unknown.

Coding and decoding for the codes constructed for Theorem 5.1 (2) and (4) seems to be highly complicated. In the next sections we show how to construct efficient solid codes with controllable error characteristics for which coding and decoding is very simple.

6. Information Rate of Solid Codes from Runs

In this section we determine the information rate – or, alternatively, the redundancy – of codes $K_{f,L,\hat{\tau}_a,\hat{\tau}_b}$ constructed according to Theorem 4.1 when $X = \{a, b\}$, $L = X^n$ for some $n \in \mathbb{N}$, $\hat{\tau}_a(w) = \hat{\tau}_b(w) = \bar{\tau}(w)$ for all $w \in X^*$, and $f(k) = k$, hence $\hat{f}(k) = k + 1$, for all $k \in \mathbb{N}$. To simplify notation, we write K_n instead of $K_{f,L,\hat{\tau}_a,\hat{\tau}_b}$ in this case. Thus K_n consists of exactly the words

$$a^{\bar{\tau}(bwa)}bwab^{\bar{\tau}(bwa)+1}$$

for $w \in X^n$. As X^n is an infix code, the set K_n is a solid code. By construction, $|K_n| = 2^n$.

Let

$$s_n = \sum_{v \in K_n} |v|.$$

Then the information rate of K_n is

$$r_n = r(K_n) = \frac{n2^n}{s_n}.$$

For s_n one computes

$$\begin{aligned} s_n &= \sum_{w \in X^n} (n + 2\bar{\tau}(bwa) + 3) \\ &= 2^n(n+3) + 2 \sum_{w \in X^n} \bar{\tau}(bwa). \\ &= 2^n(n+3) + 2^{n+1}\bar{\tau}_n^{b,a}. \end{aligned}$$

Thus

$$r_n = \frac{1}{1 + \frac{3}{n} + \frac{2\bar{\tau}_n^{a,b}}{n}}.$$

This proves the following asymptotic information rate for the solid codes K_n.

Proposition 6.1 *Let $X = \{a, b\}$ and let r_n be the information rate of the solid code K_n for $n \in \mathbb{N}$. Then*

$$\lim_{n \to \infty} r_n = 1.$$

Thus the asymptotic information rate r of the codes K_n is equal to 1, that is, asymptotically the codes are optimal with respect to information rate.

A slightly more efficient construction – omitting one of the markers – than the one of K_n is possible when L is restricted to be a full uniform code as in the present section. For any $n \in \mathbb{N}$, let

$$K'_n = \{a^{\bar{\tau}(wa)} wab^{\bar{\tau}(wa)+1} \mid w \in X^n\}.$$

Proposition 6.2 *The set K'_n is a solid code for each $n \in \mathbb{N}$.*

Proof: To simplify notation, let $\gamma(w) = a^{\bar{\tau}(wa)} wab^{\bar{\tau}(wa)+1}$ for any $w \in X^n$. First, suppose that K'_n is not an infix code. Then there are distinct words $u, v \in X^n$ such that $\gamma(u) \leq_{\mathrm{i}} \gamma(v)$. Thus there are words v_1 and v_2 with $\gamma(v) = v_1\gamma(u)v_2$ and $v_1v_2 \neq \varepsilon$. Let $v'_1 = v_1 a^{\bar{\tau}(ua)}$ and $v'_2 = ab^{\bar{\tau}(ua)+1}v_2$. Thus $\gamma(v) = v'_1 u v'_2$.

As u and v are distinct and have the same length, $u \not\leq_{\mathrm{i}} v$. Therefore, only the following two cases are possible.

Case 1. $v'_2 \in a^*vab^{\bar{\tau}(va)+1}$: Thus $u = a^n$ and $\bar{\tau}(ua) = n + 1$ resulting in $\gamma(u) = a^{n+1}a^nab^{n+2}$. Moreover, $v'_1 \in a^*$ and, therefore,

$$\bar{\tau}(va) \geq |v'_1| + n = |v_1| + 2n + 1.$$

This implies that va must have run of length at least $2n+1$, which is impossible.

Case 2. There is a proper suffix s of v such that $v_2' = sab^{\bar{\tau}(va)+1}$: Then u has a proper prefix of the form $a^{|s|}$ and $s = ab^{|s|-1}$. Thus $\bar{\tau}(ua) \geq |s|$. As

$$v_2' = ab^{\bar{\tau}(ua)+1}v_2 = sab^{\bar{\tau}(va)+1}$$

and $|s| > 0$, it follows that $ab^{\bar{\tau}(ua)+1}$ is a prefix of of s, hence $|s| > \bar{\tau}(ua)$, a contradiction. This proves that K_n' is an infix code.

Now suppose that K_n' is not overlap-free. Hence there are words $u, v \in X^n$, not necessarily distinct, such that a proper prefix $\gamma(u)$ is a proper suffix of $\gamma(v)$.

Thus, let $u_1, u_2, v_1 \in X^+$ such that $\gamma(u) = u_1u_2$ and $\gamma(v) = v_1u_1$. It follows that $u_1 \in a^{\bar{\tau}(ua)}\bar{u}b^{\bar{\tau}(va)+1}$ for some $\bar{u} \in X^*$ and $v_1 \in a^+X^*$. Only the following cases are possible.

Case 1. $u_2 \in X^*ab^{\bar{\tau}(ua)+1}$: Then ua has a run of b's of length at least $\bar{\tau}(va)+1$, hence

$$\bar{\tau}(va) < \bar{\tau}(ua) \leq n.$$

Moreover,

$$|\bar{u}| + \bar{\tau}(va) + 1 \leq n = |u| = |v|.$$

We distinguish two subcases.

Case 1a. $|v_1| \geq \bar{\tau}(va)$: Then va has a run of a's of length at least $\bar{\tau}(ua)$. Hence $\bar{\tau}(ua) \leq \bar{\tau}(va)$, a contradiction.

Case 1b. $|v_1| < \bar{\tau}(va)$: Then va has a run of a's of length at least $n - |\bar{u}|$. Hence $\bar{\tau}(va) \geq n - |\bar{u}|$, but $n - |\bar{u}| \geq \bar{\tau}(va) + 1$, a contradiction.

Case 2. $u_2 \in b^+$: Then $\bar{u} \in uab^*$, $|\bar{u}| \geq n+1$ and $\bar{\tau}(va) < \bar{\tau}(ua)$. Consequently, $\bar{u} \in X^*va$ and $\bar{\tau}(ua) \leq \bar{\tau}(va)$, again a contradiction.

This proves that K_n' is overlap-free, hence a solid code. □

The information rate of the codes K_n' is only marginally better than that of the codes K_n. Moreover, asymptotically their information rates are the same.

We now turn to the class of codes $\hat{K}_n$ mentioned at the end of the previous section. In this case $X = \{a, b\}$, $L = X^n$, $\hat{\tau}_a = \bar{\tau}_a$, $\hat{\tau}_b = \bar{\tau}_b$, and again $f(k) = k$ for all $k \in \mathbb{N}$. Thus $\hat{K}_n$ is the set of all words of the form

$$a^{\hat{\tau}_b(bwa)}bwab^{\hat{\tau}_a(bwa)+1}.$$

As the information rates $\hat{r}_n$ of the codes $\hat{K}_n$ satisfy $r_n \leq \hat{r}_n$, it follows that also $\lim_{n\to\infty} \hat{r}_n = 1$.

7. Error-Detection Capabilities

In this section we determine error-detection capabilities of codes constructed according to Theorem 4.1. For a non-empty finite language $L \subseteq X^+$ let $l_L = \max\{|w| \mid w \in L\}$. Let γ_L be a channel admitting at most one symbol substitution, insertion or deletion in any consecutive l_L symbols; in the terminology of [2, 5] one has $\gamma_L = \sigma \odot \iota \odot \delta(1, l_L)$. We apply the construction to block-codes detecting single substitution errors. We show that the resulting languages L are $(\gamma_L, *)$-detecting. As a preparation, we prove an auxiliary result being of some interest in its own right.

Proposition 7.1 *Let $n \in \mathbb{N}$, $X = \{a, b\}$, and let $C \subseteq X^n$, $C \neq \emptyset$, such that C is error-detecting for the channel*[12] *$\sigma(1, n)$. Let $h_a, h_b : X^+ \to \mathbb{N}$ be mappings. Then the language*

$$L_{C,h_a,h_b} = \{a^{h_a(bwa)}bwab^{h_b(bwa)} \mid w \in C\}$$

is error-detecting for the channel $\gamma_{L_{C,h_a,h_b}}$.

Proof: Let $w, w' \in L_{C,h_a,h_b} \cup \{\varepsilon\}$ and $w' \in \langle w \rangle_{\gamma_{L_{C,h_a,h_b}}}$. We show that $w' = w$.

Note that every word in L_{C,h_a,h_b} has at least a length of $n + 4$. Hence, a non-empty word w as input cannot result in an empty output. Thus $w = \varepsilon$ if and only if $w' = \varepsilon$.

Now assume that $w, w' \in L_{C,h_a,h_b}$. Then

$$w = a^{h_a(bua)}buab^{h_b(bua)} \text{ and } w' = a^{h_a(bu'a)}bu'ab^{h_b(bu'a)}$$

for some $u, u' \in C$. If $w = w'$ nothing needs to be proved. Hence, assume that $w \neq w'$. As $\gamma_{L_{C,h_a,h_b}}$ permits at most one error, we distinguish three cases by error type and, for each error type we consider the various possible error locations.

- *Case 1: One substitution error has occurred.* We distinguish five cases.
  - *Case 1.1: $w' = a^{t_1}ba^{t_2}buab^{h_b(bua)}$ with $t_1 + t_2 + 1 = h_a(bua)$.* As $w' \in L_{C,h_a,h_b}$, $a^{t_2}bu \in C$, but $|a^{t_2}bu| > n$, a contradiction.
  - *Case 1.2: $w' = a^tbsab^{h_b(bua)}$ for a suffix bs of u.* Then $u' = s$ and $|u'| < n$, a contradiction.
  - *Case 1.3: $w' = a^{h_a(bua)}bu'ab^{h_b(bua)}$ with $u \neq u'$.* Then $u, u' \in C$ and u and u' differ in one symbol, contradicting the assumption about C.
  - *Case 1.4: $w' = a^{h_a(bua)}pab^t$ for a prefix pa of u.* This is the dual of Case 1.2.
  - *Case 1.5: $w' = a^{h_a(bua)}uab^{t_2}ab^{t_1}$ with $t_1 + t_2 + 1 = h_b(bua)$.* This is the dual of Case 1.1.
- *Case 2: One deletion error has occurred.* Again we distinguish five cases.
  - *Case 2.1: $w' = a^tbuab^{h_b(bua)}$ with $t = h_a(bua) - 1$.* This is impossible as h_a is a mapping.
  - *Case 2.2: $w' = a^{h_a(bua)}uab^{h_b(bua)}$.* Then $|u'| < |u| = n$, a contradiction.
  - *Case 2.3: $w' = a^{h_a(bua)}bu'ab^{h_b(bua)}$ with $|u'| < |u|$.* A contradiction.

  The cases 2.4 and 2.5 are the duals of the cases 2.2 ans 2.1, respectively.
- *Case 3: One insertion error has occurred.* Again we distinguish five cases.
  - *Case 3.1: $w' = a^{h_a(bua)+1}buab^{h_b(bua)}$.* This is impossible as h_a is a mapping.
  - *Case 3.2: $w' = a^{t_1}ba^{t_2}buab^{h_b(bua)}$ with $t_1 + t_2 = h_a(bua)$.* Then $u' = a^{t_2}bu$, hence $|u'| > n$, a contradiction.
  - *Case 3.3: $w' = a^{h_a(bua)}bu'ab^{h_b(bua)}$ with $|u'| = |u| + 1$.* Then $|u'| > n$, a contradiction.

  The cases 3.4 and 3.5 are the duals of the cases 3.1 and 3.2, respectively.

This shows that the assumption $w' \neq w$ is false. Thus L_{C,h_a,h_b} is error-detecting for $\gamma_{L_{C,h_a,h_b}}$ as claimed. □

[12] This channel permits 1 substitution error in every n consecutive symbols. The notion of error-detection in this sense for $\sigma(1, n)$ is a weaker than the notion normally used in the theory of block codes, but possibly more natural. See [2] for more detailed explanations.

To prove the main result of this section we need the following theorem from [6].

Theorem 7.1 *A finite solid code L is $(\gamma_L, *)$-detecting if and only if it is $(\gamma_L, 1)$-detecting.*

Theorem 7.2 *Let $X = \{a, b\}$, $n \in \mathbb{N}$, and let C be a uniform code of length n which is error-detecting for the channel $\sigma(1, n)$. Let $f : \mathbb{N} \to \mathbb{N}$ be a monotonically increasing function satisfying $S(f, \infty, \infty)$, and let $\hat{f}$ be its near-inverse. Let*

$$K_{f,C,\hat{\tau}_a,\hat{\tau}_b} = \{a^{f(\hat{\tau}_b((bwa)))}bwab^{\hat{f}(\hat{\tau}_a((bwa)))} \mid w \in C\}.$$

The language $K_{f,C,\hat{\tau}_a,\hat{\tau}_b}$ has the following properties.
(1) $K_{f,C,\hat{\tau}_a,\hat{\tau}_b}$ is error-detecting for $\gamma_{K_{f,C,\hat{\tau}_a,\hat{\tau}_b}}$.
(2) $K_{f,C,\hat{\tau}_a,\hat{\tau}_b}$ is $(\gamma_{K_{f,C,\hat{\tau}_a,\hat{\tau}_b}}, *)$-detecting when

$$\max\{f(r) + \hat{f}(n + 2 - r) \mid 1 \leq r \leq n + 1\} < n + 6.$$

This condition is satisfied, in particular, when $f(r) = r$ for all $r \in \mathbb{N}$.

Proof: Statement (1) follows from Proposition 7.1. For Statement (2), we show that $K_{f,C,\hat{\tau}_a,\hat{\tau}_b}$ is $(\gamma_{K_{f,C,\hat{\tau}_a,\hat{\tau}_b}}, 1)$-detecting. As $K_{f,C,\hat{\tau}_a,\hat{\tau}_b}$ is a solid code by Theorem 4.1, it is also $(\gamma_{K_{f,C,\hat{\tau}_a,\hat{\tau}_b}}, *)$-detecting by Theorem 7.1.

Consider $w \in K^*_{f,C,\hat{\tau}_a,\hat{\tau}_b}$ and $w' \in \langle w\rangle_{\gamma_{K_{f,C,\hat{\tau}_a,\hat{\tau}_b}}}$ such that $w' \in K_{f,C,\hat{\tau}_a,\hat{\tau}_b} \cup \{\varepsilon\}$. We show that $w' = w$.

As in the previous proof, $w = \varepsilon$ if and only if $w' = \varepsilon$. Hence assume now that $w \in K^m_{f,C,\hat{\tau}_a,\hat{\tau}_b}$ for some $m \in \mathbb{N}$. Then $w' \in K_{f,C,\hat{\tau}_a,\hat{\tau}_b}$.

If $m = 1$, then $w' = w$ by Statement (1). Hence, assume that $m \geq 2$. As $f(1)+\hat{f}(1) > 2$, one has

$$|w| \geq m(n + 2 + f(1) + \hat{f}(1)) \geq mn + 5m.$$

There can be at most m deletions or insertions in w. Therefore,

$$mn + 4m \leq |w'|.$$

As $w' \in K_{f,C,\hat{\tau}_a,\hat{\tau}_b}$,

$$|w'| \leq n + 2 + f(r_1) + \hat{f}(r_2)$$

for some $r_1, r_2 \in \mathbb{N}$ with $r_1 + r_2 \leq n + 2$. Thus, using the assumptions about f,

$$|w'| < n + 2 + n + 6 = 2n + 8$$

whereas $mn + 4m \geq 2n + 8$, a contradiction.

Finally, when $f(r) = r$ for all r, then $\hat{f}(r) = r + 1$. Hence

$$f(r) + \hat{f}(n + 2 - r) = r + n + 2 - r + 1 = n + 3$$

for all r with $1 \leq r \leq n+1$ (actually for all r). This proves the second part of Statement (2). □

We conclude this section with two examples involving the alphabet $X = \{0, 1\}$. As a first example, we consider for every positive integer n the even parity uniform code $C_n = \{wp(w) \mid w \in X^{n-1}\}$, of length n, such that $p(w) = 1$ if w contains an odd number of 1's, or $p(w) = 0$ otherwise. Let $f(r) = r$, for all $r \in \mathbb{N}$, and let $\hat{\tau}_a(w) = \bar{\tau}_a(w)$ and $\hat{\tau}_b(w) = \bar{\tau}_b(w)$ for all words w in X^+. By Theorem 7.2, the code $K_n = K_{f,C_n,\hat{\tau}_a,\hat{\tau}_b}$ is $(\gamma_{K_{f,C_n,\hat{\tau}_a,\hat{\tau}_b}}, *)$-detecting. Obviously, $|K_n| = |C_n| = 2^{n-1}$. Moreover, the sum of the code word lengths of K_n is

$$\sum_{w \in X^{n-1}} (n + 3 + \bar{\tau}_b(1wp(w)0) + \bar{\tau}_a(1wp(w)0)),$$

which is bounded by $2^{n-1}(n + 7) + 2^n \bar{\tau}_{n-1}$. This implies that $r(K_n)$ tends to 1, as $n \to \infty$.

As a second example we assume that X is equipped with the structure of GF(2) and consider the *Hamming code* of length $n = 7$ given by the (transposed) parity check matrix

$$H^T = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Thus,

$$\begin{aligned} C = \{&0000000, 1110000, 1001100, 1011010, \\ &1100110, 1101001, 1010101, 1000011, \\ &1111111, 0111100, 0101010, 0100101, \\ &0110011, 0010110, 0011001, 0001111\}. \end{aligned}$$

Using $h_0(w) = \bar{\tau}(w)$ and $h_1(w) = \bar{\tau}(w) + 1$ (and $a = 0$, $b = 1$), the construction yields the following set C' of words:

000000001000000000111111111,	00000111100000111111,
0001100110001111,	00110110100111,
0001110011001111,	0001110100101111,
00110101010111,	000011000011011111,
000000001111111110111111111,	000010111100011111,
00101010100111,	00101001010111,
00101100110111,	00100101100111,
00100110010111,	000010001111011111.

The average code word length of C' is

$$\frac{1}{16} \cdot (2 \cdot 26 + 1 \cdot 20 + 3 \cdot 18 + 3 \cdot 16 + 7 \cdot 14) = 17.$$

The information rate of C' is equal to $4/17$ while the information rate of C is $4/7$. The Hamming code C is error-detecting for the channel $\sigma(2, 7)$ and error-correcting for the channel $\sigma(1, 7)$. By Theorem 7.2, C' is error-detecting for the channel $\sigma \odot \iota \odot \delta(1, 26)$. It can be shown further that C' is also error-detecting for the channel that permits either two substitutions or one synchronization error (but not both) in any word of length 26.

8. Concluding Remarks

A new construction of solid codes yields classes of asymptotically optimal codes for synchronization, the coding and decoding for which is very simple to implement. Moreover, to some extent, the construction permits one to control other parameters like the error-detection capability of the codes. What makes the construction particularly attractive is that it can exploit known properties of block codes to yield codes for synchronization.

Several lines of research are suggested by our results: How are error-detection and error-correction properties passed on from the block codes to the solid codes? How does the representation of the block code influence the information rate of the resulting solid code? Can one strengthen the results by using non-block codes as the basis of the construction?

We suggest that the results of the present paper provide enough evidence to warrant the investigation of solid codes for actual applications in communication systems.

References

[1] H. Jürgensen, M. Katsura, S. Konstantinidis: Maximal solid codes. *Journal of Automata, Languages and Combinatorics* **6** (2001), 25–50.

[2] H. Jürgensen, S. Konstantinidis: Codes. In Rozenberg and Salomaa [13], 511–607.

[3] H. Jürgensen, S. Konstantinidis: Redundancy of solid codes. In D. L. Van, M. Ito (editors): *The Mathematical Foundation of Informatics, Proceedings of a Conference, Hanoi, October 25–28, 1999.* World Scientific, Singapore, 2002, to appear.

[4] H. Jürgensen, S. S. Yu: Solid codes. *J. Inform. Process. Cybernet., EIK* **26** (1990), 563–574.

[5] S. Konstantinidis: An algebra of discrete channels that involve combinations of three basic error types. *Inform. and Comput.* **167** (2001), 120–131.

[6] S. Konstantinidis, A. O'Hearn: Error-detecting properties of languages. *Theoret. Comput. Sci.* (to appear).

[7] N. H. Lâm: Finite maximal solid codes. *Theoret. Comput. Sci.* **262** (2001), 333–347.

[8] V. I. Levenshtein: Decoding automata, invariant with respect to the initial state. *Problemy Kibernet.* **12** (1964), 125–136, in Russian.

[9] V. I. Levenshtein: On the redundancy and delay of decodable coding of natural numbers. *Problemy Kibernet.* **20** (1968), 173–179, in Russian. English translation: *Systems Theory Research* **20** (1971), 149–155.

[10] V. I. Levenshtein: Maximum number of words in codes without overlaps. *Problemy Peredachi Informatsii* **6**(4) (1970), 88–90, in Russian. English translation: *Problems Inform. Transmission* **6**(4) (1973), 355–357.

[11] A. A. Markov: Some properties of infinite prefix codes. *Problemy Peredachi Informatsii* **6**(1) (1970), 97–98, in Russian. English translation: *Problems Inform. Transmission* **6**(1) (1973), 85–87.

[12] W. W. Peterson, E. J. Weldon, Jr.: *Error-Correcting Codes*. MIT Press, Cambridge, MA, second ed., 1972.

[13] G. Rozenberg, A. Salomaa (editors): *Handbook of Formal Languages*. Springer-Verlag, Berlin, 1997.

[14] H. J. Shyr, S. S. Yu: Solid codes and disjunctive domains. *Semigroup Forum* **41** (1990), 23–37.

[15] J. J. Stiffler: *Theory of Synchronous Communications*. Prentice-Hall, Englewood Cliffs, New Jersey, 1971.