Correspondence

On a Simple Method for Detecting Synchronization Errors in Coded Messages

Stavros Konstantinidis, *Member, IEEE*, Steven Perron, and L. Amber Wilcox-O'Hearn

***Abstract*—We investigate the problem of designing pairs (p, s) of words with the property that, if each word of a coded message is prefixed by p and suffixed by s, the resulting set of coded messages is error detecting with finite delay. We consider (combinatorial) channels permitting any combination of the substitution, insertion, and deletion (SID) error types, and address the cases of both scattered and burst errors. A pair (p, s) with the above property is evaluated in terms of three parameters: redundancy, delay of decoding, and frequency of the detectable errors. In the case of SID channels with burst errors, we provide a complete and explicit characterization of their error-detecting pairs (p, s), which involves the period of the word sp.**

***Index Terms*—Burst errors, decoding delay, deletion, error detection, insertion, period of word.**

I. Introduction and Basic Notation

We investigate the problem of designing pairs (p, s) of words, called *separators*, with the property that, if each word of a coded message is prefixed by p and suffixed by s, the resulting coded language (set of coded messages) is *error detecting with finite delay*. We consider (combinatorial) channels allowing any combination of the substitution, insertion, and deletion (SID) error types. Such channels were used by Levenshtein in [1], where the method of separators was discussed for *correcting scattered* SID errors in coded messages. This method was first considered by Sellers Jr. [2] for a certain SID channel and, more recently, by Ferreira *et al.* [3]. In the present correspondence, we use the term *uniform error-detector* for a pair (p, s) of words satisfying the above property, and we consider the cases of both *scattered* and *burst* errors. In either case, a uniform error-detector (p, s) is evaluated in terms of three parameters: the redundancy $|p| + |s|$, the delay of decoding, and the frequency of the detectable errors. In the case of burst SID errors, we provide a complete and explicit characterization of all the uniform error-detectors (p, s), which involves the period of the word sp [4], [5].

A. Notation About Alphabet, Words, and Coded Languages

We assume an alphabet X containing at least the two symbols 0 and 1. A *word*, or *message*, is any string of symbols from X including the empty word λ. For a word w, we denote by $|w|$ the length of w. For example, $|11001| = 5$. If $i = 1, \ldots, |w|$, then $w[i]$ denotes the ith symbol of w and, for $j = i, \ldots, |w|$, the notation $w[i \cdots j]$ represents the word $w[i]w[i+1] \cdots w[j]$. If $j < i$, we agree that $w[i \cdots j] = \lambda$.

Manuscript received December 2, 2001; revised November 29, 2002. This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant R220259.

S. Konstantinidis and S. Perron are with the Department of Mathematics and Computing Science, Saint Mary's University, Halifax, NS B3H 3C3, Canada (e-mail: s.konstantinidis@stmarys.ca; steven_perron@hotmail.com).

L. A. Wilcox-O'Hearn is with the Department of Computer Science, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: amber@cs.toronto.edu).

Communicated by C. Carlet, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2003.810665

For two words w_1 and w_2 the word w_1w_2 is the concatenation of w_1 and w_2. For a word w and a nonnegative integer n, w^n denotes the word that consists of n concatenated copies of w. A word p is a *prefix* of w if $w = ps$, for some word s. In this case, s is a *suffix* of w. If $p \neq w$, then p is called a proper prefix of w. Similarly, the suffix s is proper if $s \neq w$. A word v is a *factor* of w if w can be written as xvy for some words x and y. The set of all words is denoted by X^* and the set of all nonempty words by X^+. Every subset of X^* is called a *language*. If u, v are words and L is a language, then uLv is the language $\{uwv \mid w \in L\}$. If F is a finite and nonempty language, then ℓ_F denotes the maximum length of the words in F. For two languages L and L', LL' denotes the language $\{ww' \mid w \in L,\ w' \in L'\}$. If n is a nonnegative integer, then L^n is the language $\{w_1 \cdots w_n \mid w_1, \ldots, w_n \in L\}$, with $L^0 = \{\lambda\}$. Moreover, $L^* = \bigcup_{i=0}^{\infty} L^i$.

A nonempty language C is called a *uniquely decodable code* or simply a *code* if, for all positive integers m and n and for all words

$$v_1, \ldots, v_n, u_1, \ldots, u_m \in C$$

the equation

$$v_1v_2 \cdots v_n = u_1u_2 \cdots u_m$$

implies $m = n$ and $u_i = v_i$ for every $i = 1, \ldots, n$. In this correspondence, we assume that every code contains at least two words. If all the words of a language C have the same length, then C is a code and is called a *uniform code*. A language of the form C^*, where C is a code, is called a *coded language*.

B. Structure of the Correspondence

This correspondence is organized as follows. In Section II, we give the basic terminology about (combinatorial) channels and error detection, and we define SID error types, error specifications, and the particular class of SID channels that permit *burst* errors. Moreover, we obtain a few technical results pertaining to these concepts. In Section III, we define the method of uniform error-detector pairs, discuss the criteria for choosing good pairs, and provide a necessary condition on the structure of such pairs. In Section IV, we focus on channels with burst errors and identify all uniform error-detectors for such channels, including all the optimal ones. In Section V, we consider channels with scattered errors and obtain a set of uniform error-detectors that work for any SID error type. Then, for certain error types, we identify uniform error-detectors with smaller redundancy at the cost of restricting to messages over the binary alphabet. Finally, Section VI contains a few concluding remarks.

II. (Combinatorial) Channels and Error Detection

A. Channels, Error Types, and Error Specifications

A *(combinatorial) channel* (over the alphabet X) is a binary relation $\gamma \subseteq X^* \times X^*$. For the elements of the channel γ we prefer to write $(z \leftarrow w)$ as opposed to (z, w). Then, $(z \leftarrow w) \in \gamma$ means that the message (word) z can be received from w through the channel γ. Note that, in general, the channel is noisy, meaning that, for $(z \leftarrow w) \in \gamma$, it is possible that $z \neq w$; that is, z is received from w with errors.

In this work, we consider channels that permit combinations of the three *basic error types*: substitution, insertion, and deletion, denoted by the symbols σ, ι, and δ, respectively. The *set of error types* is

$$\{\sigma, \iota, \delta, (\sigma \odot \iota), (\sigma \odot \delta), (\iota \odot \delta), (\sigma \odot \iota \odot \delta)\}$$

where the symbol $\odot$ is a logic connective and is used to indicate which combination of the basic error types is permitted. The error types ι and δ are called *synchronization* error types. Each error type τ can be used to define a distance function D_τ. In particular, D_σ is the Hamming distance and, for $\tau \in \{(\iota \odot \delta), (\sigma \odot \iota \odot \delta)\}$, D_τ is the Levenshtein distance for errors of type τ [6].

Definition 1: An *error specification* is an expression of the form $m\boldsymbol{x}\tau$ such that m is a nonnegative integer, τ is an error type, and $\boldsymbol{x}$ is one of the symbols $\boldsymbol{b}$ and $\boldsymbol{s}$ indicating the terms *burst* and *scattered*, respectively. For an error specification $m\boldsymbol{x}\tau$ we shall assume that $m > 0$, unless stated otherwise.

Intuitively, the expression $m\boldsymbol{x}\tau$ specifies possible changes (errors) that one can make in a word to obtain another word. For example, the expression $3\boldsymbol{b}\delta$ specifies a size 3 burst of deletion errors and the expression $4\boldsymbol{s}(\iota \odot \delta)$ specifies four scattered insertion and deletion errors. More specifically, let x and y be two words. We say that y *obtains from* x *using* $m\boldsymbol{s}\tau$, if it is possible to transform x to y using exactly m (scattered) errors of type τ. We say that y obtains from x using *at most* $m\boldsymbol{s}\tau$, if y obtains from x using $k\boldsymbol{s}\tau$, for some integer k with $0 \leq k \leq m$. We say that y *obtains from* x *using (at most)* $m\boldsymbol{b}\tau$, if there are words p, s, u, u' such that $|u| \leq m$, $x = pus$, $y = pu's$, and u' obtains from u using at most $m\boldsymbol{s}\tau$. For example, the word 111111 obtains from 0111110 using $2\boldsymbol{s}(\sigma \odot \delta)$, and the word 00010100 obtains from 0011100 using $2\boldsymbol{b}(\sigma \odot \iota)$.

For an error type τ and two integers m and ℓ, with $1 \leq m < \ell$, the expression $\tau_{\boldsymbol{s}}(m, \ell)$ denotes an *SID channel with scattered errors*. Specifically, $(z \leftarrow w)$ is in the channel if it is possible to obtain z from w using errors of type τ such that no more than m errors can be used in any factor of length ℓ (or less) of w. We note that channels $\tau_{\boldsymbol{s}}(m, \ell)$ with $\tau \in \{\iota, \delta, (\iota \odot \delta), (\sigma \odot \iota \odot \delta)\}$ are considered in [1] in the context of error-correcting codes for such channels. As an example, consider the channel $(\sigma \odot \delta)_{\boldsymbol{s}}(2, 5)$ that permits a total of up to two substitutions and deletions in any factor of length 5 of the message. As 101000 obtains from $w = 0000000$ by deleting $w[7]$ and substituting $w[3]$ with 1 and $w[1]$ with 1, it follows that $(101000 \leftarrow w) \in (\sigma \odot \delta)_{\boldsymbol{s}}(2, 5)$. On the other hand, to obtain 101001 from w using errors of type $(\sigma \odot \delta)$, one symbol of w must be deleted and three of its symbols must be substituted. But it is not possible to choose four such symbols, unless three of them occur in a factor of w of length 5. Hence, $(101001 \leftarrow w)$ is not in the channel $(\sigma \odot \delta)_{\boldsymbol{s}}(2, 5)$.

B. Channels With Burst Errors

Let m be a positive integer. A set B of *m-burst errors* is a set that consists of pairs $(u' \leftarrow u)$ such that $|u| \leq m$. Of particular interest are the SID sets of m-burst errors: For any error type τ, define

$$B_m(\tau) = \{(u' \leftarrow u) \mid |u| \leq m \text{ and } u' \text{ obtains from } u \text{ using at most } m\boldsymbol{b}\tau\}.$$

Given a set B of m-burst errors and an integer $\ell > m$, we define the channel $\gamma_\ell[B]$ as follows: $(z \leftarrow w)$ is in $\gamma_\ell[B]$ if and only if there is an integer $n \geq 0$ and words $x_0, \ldots, x_n, u_1, \ldots, u_n, u'_1, \ldots, u'_n$ such that

- $w = x_0u_1x_1 \cdots u_nx_n$ and $z = x_0u'_1x_1 \cdots u'_nx_n$;
- for all $i = 1, \ldots, n-1$, $|x_i| \geq \ell - 1$;
- $(u'_i \leftarrow u_i)$ is in B and $u'_i \neq u_i$, for all $i = 1, \ldots, n$.

Informally, the above conditions mean that if z obtains from w through $\gamma_\ell[B]$ then, in w, there are zero or more bursts of errors each of size m (or less) such that there is at most one burst (or part of a burst) in any ℓ consecutive symbols of w. In case $B = B_m(\tau)$, we call the channel $\gamma_\ell[B_m(\tau)]$ an *SID channel with burst errors* and we shall use the notation $\tau_{\boldsymbol{b}}(m, \ell)$ instead of $\gamma_\ell[B_m(\tau)]$ which is consistent with the notation we use for SID channels with scattered errors. For example, let $w = 0^2110^7110^81$ and let $z = 0^21000^700^8010$. Then $(z \leftarrow w)$ is in the channel $(\sigma \odot \iota \odot \delta)_{\boldsymbol{b}}(2, 8)$.

C. Error Detection With Finite Delay

Let γ be a channel. A language L is called *error detecting for* γ when the following condition is satisfied: Assuming only words in $L \cup \{\lambda\}$ are sent into γ, if a word y is received through γ and y is in $L \cup \{\lambda\}$ then y must be correct; that is, equal to the transmitted word. This condition can be written more formally as follows [7]:

$$\text{For all words } x \in L \cup \{\lambda\} \text{ and } y \in X^*,$$
$$\text{if } (y \leftarrow x) \in \gamma \text{ and } y \in L \cup \{\lambda\}, \text{ then } y = x.$$

Note that the above condition ensures that a nonempty word of L cannot be received from the empty word, and the empty word cannot be received from a nonempty word of L. The task of verifying that a language is error detecting could require some effort even for apparently simple languages and channels. We invite the reader to show that, for $K = \{1011, 1101\}$, the coded language K^* is error detecting for the SID channel $\delta_{\boldsymbol{b}}(1, 4)$ that allows up to one deletion in any four consecutive symbols of a message.

Although error detection is a basic property of a communications language, the process of decoding a word of such a language might require unbounded memory. This is because, in general, the decoder needs to see the entire message in order to decide whether it is correct. For coded languages, however, it is possible to define the concept of error detection with finite delay as follows. If a codeword v is observed at the beginning of the received message, then v is correct—that is, equal to the first codeword of the transmitted message—provided that there are at least d codewords following v in the received message. Moreover, once v is decoded, the rest of the received message can be decoded in the same way. The number d is the delay of decoding. In case the received message does not begin with $1 + d$ codewords, an error is detected and what follows depends on the communication protocol—usually involving retransmission techniques. More formally, for a channel γ and a nonnegative integer d, we say that the *coded language C^* is error detecting for γ with delay d, when the following condition is satisfied for all $v \in C$, $z \in C^dX^*$, and $w \in C^*$*:

$$\text{if } (vz \leftarrow w) \in \gamma, \text{ then } w = vu \text{ and } (z \leftarrow u) \in \gamma \text{ for some } u \in C^*.$$

For any reasonable channel γ—an SID channel, for instance—if C^* satisfies the above condition then C^* is indeed error detecting for γ [7]. Moreover, if C^* satisfies the above then the code C has finite decoding (deciphering) delay at most d (in the sense of [8]).

Example 1: Consider again the code $K = \{1101, 1011\}$ and recall that the language K^* is error detecting for $\delta_{\boldsymbol{b}}(1, 4)$. However, K^* is not error detecting for that channel with any finite delay. Indeed, assume that K^* is error detecting with delay d, for some nonnegative integer d, and consider the words $w = (1101)^{d+2}$, $v = 1011$, and $z = (1011)^d101$. By deleting the first 1 of w we have that $(vz \leftarrow w)$ is in the channel which implies that w must start with the codeword v; a contradiction.

Example 2: Consider the code $C = \{001, 011\}$ and the channel $\delta_{\boldsymbol{b}}(1, 4)$. Then, C^* is error detecting for $\delta_{\boldsymbol{b}}(1, 4)$ with delay 0. Indeed, suppose $v \in C$, $z \in X^*$, and $w \in C^*$ such that $(vz \leftarrow w)$ is in $\delta_{\boldsymbol{b}}(1, 4)$. Note that w must start with a codeword of the form $0b1$ for some alphabet symbol b. Then, $w = 0b1u$ for some $u \in C^*$. Suppose that $0b1$ results in some word z_1 and u results in some word z_2 such that $vz = z_1z_2$. If there is a deletion in $0b1$, then $|v| = |z_1| + 1$ and there can be no error on the symbol $u[1]$ that follows $0b1$. As $u[1] = 0$, we have that $v = z_10$, which contradicts the assumption $v \in C$. Thus, there can be no deletion in $01b$, which implies that $v = z_1 = 0b1$ and $z_2 = z$ and, therefore, $(z \leftarrow u) \in \delta_{\boldsymbol{b}}(1, 4)$. The code C allows

for the correct decoding of messages in C^* without delay as long as no error occurs. For example, suppose that the concatenation of codewords $001, 011, 001, 001, \ldots$ is transmitted and a deletion occurs in the third codeword, 001, so that we receive $001011010\cdots$. Then, the codewords 001 and 011 can be decoded correctly and an error is detected when 010 is observed.

D. Some Technical Results

We close this section with a few technical results concerning error specifications and SID channels, which are needed in the sequel. The proofs can be found in the Appendix.

Lemma 1: Let $m\boldsymbol{x}\tau$ be an error specification, other than $m\boldsymbol{s}(\iota \odot \delta)$ and $m\boldsymbol{s}(\sigma \odot \iota \odot \delta)$, containing at least two different error types and let v and v' be two words of the same length. Then, v' obtains from v using at most $m\boldsymbol{x}\tau$ if and only if v' obtains from v using at most $m\boldsymbol{x}\sigma$.

Lemma 2: Let $m\boldsymbol{x}\tau$ be an error specification and let p, v, v', s be words such that v and v' are of the same length. Then, $pv's$ obtains from pvs using at most $m\boldsymbol{x}\tau$ if and only if v' obtains from v using at most $m\boldsymbol{x}\tau$.

Next we establish certain relationships between sets of the form $B_m(\tau)$ involving different error types.

Lemma 3: Let m be a positive integer and let u and u' be two words such that $|u| \leq m$ and u' obtains from u using at most $m\boldsymbol{s}(\sigma \odot \iota \odot \delta)$.

1) If $|u'| \leq |u|$, then u' obtains from u using at most $m\boldsymbol{s}(\sigma \odot \delta)$.
2) If $|u'| > |u|$, then u' obtains from u using at most $\max\{1, 2m-2\}\boldsymbol{s}(\sigma \odot \iota)$.
3) If the alphabet X is binary and $|u'| = |u| + 1$, then u' obtains from u using at most $m\boldsymbol{s}(\sigma \odot \iota)$.

Proposition 1: For all integers $m \geq 2$

$$B_m(\sigma \odot \iota \odot \delta) \subseteq B_m(\sigma \odot \delta) \cup B_{2m-2}(\sigma \odot \iota).$$

Proof: The claim follows easily from the previous lemma. □

A natural question that arises is whether Proposition 1 can be strengthened by replacing $B_{2m-2}(\sigma \odot \iota)$ with $B_m(\sigma \odot \iota)$. It turns out that this is possible only when $m < 5$.

Proposition 2: Consider the alphabet X and a positive integer m. Then, $B_m(\sigma \odot \iota \odot \delta) = B_m(\sigma \odot \delta) \cup B_m(\sigma \odot \iota)$ if and only if $m \in \{1, 2\}$, or $m \in \{3, 4\}$ and $X = \{0, 1\}$.

The above considerations motivate us to define a new type of SID channels with burst errors as follows. For two error types τ_1 and τ_2, let $(\tau_1 \vee \tau_2)_{\boldsymbol{b}}(m, \ell)$ be the channel $\gamma_\ell[B_m(\tau_1) \cup B_m(\tau_2)]$. Then, every burst of errors in a transmitted message is of type $m\boldsymbol{b}\tau_1$ or $m\boldsymbol{b}\tau_2$. By Proposition 2, it follows that, for $X = \{0, 1\}$ and $m < 5$, the channel $((\sigma \odot \iota) \vee (\sigma \odot \delta))_{\boldsymbol{b}}(m, \ell)$ coincides with the channel $(\sigma \odot \iota \odot \delta)_{\boldsymbol{b}}(m, \ell)$.

III. The Method of Separators for Detecting SID Errors

In [1], Levenshtein briefly discusses the method of separators for correcting certain scattered SID errors in messages, with finite delay. Loosely speaking, if a coded language K^* is error correcting for a channel γ with finite delay then, for every received message w, it is possible to determine the first codeword of the original message by looking only at a prefix of w of bounded length. The method of separators involves choosing an appropriate pair of words (p, s) such that $(pCs)^*$ is error correcting for γ with finite delay, for any uniform code C that is error correcting for γ. In this section, we use the above idea to define a formal method for obtaining coded languages of the form $(pCs)^*$ that are error detecting with finite delay for SID channels with scattered errors or with burst errors.

Let m be a positive integer. The symbol $\mathcal{U}^m$ denotes the class of all uniform codes of length greater than m. For an error specification $m\boldsymbol{x}\tau$, we write $\mathcal{U}_{m\boldsymbol{x}\tau}$ for the class of all uniform codes C such that C is error detecting for $\tau_{\boldsymbol{x}}(m, \ell_C)$—recall, ℓ_C is the word length of the code C. By the definition of error detection and by Lemma 1, the following obtains.

Lemma 4: Let $m\boldsymbol{x}\tau$ be an error specification.

- If $\tau \in \{\iota, \delta\}$, then $\mathcal{U}_{m\boldsymbol{x}\tau} = \mathcal{U}^m$.
- If $\tau \notin \{\iota, \delta\}$ and $\boldsymbol{x} = \boldsymbol{b}$, then $\mathcal{U}_{m\boldsymbol{x}\tau} = \mathcal{U}_{m\boldsymbol{b}\sigma}$.
- If $\tau \in \{\sigma, (\sigma \odot \delta), (\sigma \odot \iota)\}$ and $\boldsymbol{x} = \boldsymbol{s}$, then $\mathcal{U}_{m\boldsymbol{x}\tau} = \mathcal{U}_{m\boldsymbol{s}\sigma}$.

It follows also that

$$\mathcal{U}_{m\boldsymbol{s}(\sigma \odot \iota \odot \delta)} \subseteq \mathcal{U}_{m\boldsymbol{s}\sigma} \subseteq \mathcal{U}_{m\boldsymbol{b}\sigma} \subseteq \mathcal{U}^m.$$

Codes in the classes $\mathcal{U}_{m\boldsymbol{s}\sigma}$ and $\mathcal{U}_{m\boldsymbol{b}\sigma}$ have been studied extensively—see [9], for instance. On the other hand, at a first glance, it appears that codes in the classes $\mathcal{U}_{m\boldsymbol{s}(\sigma \odot \iota \odot \delta)}$ and $\mathcal{U}_{m\boldsymbol{s}(\iota \odot \delta)}$ have not been considered in the past. By the results of [1], however, the following obtains.

Remark 1: Let $\tau \in \{(\iota \odot \delta), (\sigma \odot \iota \odot \delta)\}$ and let m be a positive integer. A uniform code C is error correcting for $\tau_{\boldsymbol{s}}(m, \ell_C)$ if and only if it is error detecting for $\tau_{\boldsymbol{s}}(2m, \ell_C)$.

The above follows when we note that i) a code C is error correcting for $\tau_{\boldsymbol{s}}(m, \ell_C)$ if and only if $D_\tau(C) > 2m$, and ii) a code C is error detecting for $\tau_{\boldsymbol{s}}(m, \ell_C)$ if and only if $D_\tau(C) > m$. We also note that, to our knowledge, only very few general construction methods exist for codes that are error correcting for $(\iota \odot \delta)_{\boldsymbol{s}}(m, \ell_C)$—see, for instance, [10]–[12].

Definition 2: Let $m\boldsymbol{x}\tau$ be an error specification. A pair of words (p, s) is called a *uniform error-detector for* $m\boldsymbol{x}\tau$ (or simply an $m\boldsymbol{x}\tau$-*detector*) if there are two nonnegative integers d and t such that, for every code C in $\mathcal{U}_{m\boldsymbol{x}\tau}$, the coded language $(pCs)^*$ is error detecting for $\tau_{\boldsymbol{x}}(m, \ell_C + |ps| + t)$ with delay d. Then, we say that (p, s) has redundancy $|p| + |s|$, delay d, and offset t.

For a given error specification $m\boldsymbol{x}\tau$, the design of a uniform $m\boldsymbol{x}\tau$-detector should consider the following criteria.

1) Low redundancy of the encoding $C \mapsto pCs$: This is achieved by choosing a pair (p, s) with small redundancy $|p| + |s|$.
2) High frequency of the errors detectable by $(pCs)^*$: This is achieved by choosing a pair (p, s) with small offset t. Indeed, an $m\boldsymbol{x}\tau$-detector (p, s) with offset t ensures the detection of m errors of type τ in any $\ell_C + |ps| + t$ symbols of the transmitted message. Thus, the smaller is the value of t the higher is the ratio $m/(\ell_C + |ps| + t)$.
3) Small amount of memory for decoding words in $(pCs)^*$: This is achieved by choosing a pair (p, s) with small delay d.

Our primary criterion will be the optimization of the redundancy of a uniform error-detector (p, s). With this constraint, we shall attempt to define error-detectors with minimal delay and minimal offset.

The first result gives a necessary condition on the structure of uniform error-detectors that involves the notion of period of a word. A positive integer k is called *a period* of a nonempty word w, if $w[i] = w[k+i]$ for every $i \in \{j \mid 1 \leq j \text{ and } j \leq |w| - k\}$—note that this condition is vacuously true when $k \geq |w|$ and, in this case, the number k is a period of w. The smallest k satisfying this condition is called *the period* of the word w and we denote it by $\mathrm{per}(w)$. It should be clear that $1 \leq \mathrm{per}(w) \leq |w|$. This concept is important in various domains

including pattern matching algorithms and game theory, [13], and word combinatorics [4].

Lemma 5:

1) For any nonempty word w there are words $u,\ x,\ y$ such that $w = ux = yu$ and $|x| = |y| = \mathrm{per}(w)$.
2) For every nonempty words $u_1,\ x_1,\ u_2,\ x_2$ with $|u_1| = |u_2|$ and $u_1x_1 = x_2u_2$, if $\mathrm{per}(u_1x_1) > |x_1|$ then $u_1 \neq u_2$.

Proof: The statements follow easily if we note that k is a period of w if and only if, either $k \geq |w|$ or $w = xu = uy$ for some $u \in X^+$ and $x,\ y \in X^k$. □

Proposition 3: Let $m\boldsymbol{x}\tau$ be an error specification not in $\{m\boldsymbol{x}\sigma,\ m\boldsymbol{s}(\sigma \odot \iota \odot \delta),\ m\boldsymbol{s}(\iota \odot \delta)\}$. If a pair of words $(p,\ s)$ is a uniform error-detector for $m\boldsymbol{x}\tau$, then $\mathrm{per}(sp) > m$ and, therefore, the redundancy of $(p,\ s)$ is greater than m.

Proof: Assume $(p,\ s)$ is a uniform $m\boldsymbol{x}\tau$-detector and consider first the case where $|sp| > 0$. Then, $sp = ux = yu$ for some words $u,\ x,\ y$ with $|x| = |y| = \mathrm{per}(sp)$. Suppose $\mathrm{per}(sp) \leq m$. We shall obtain a contradiction by constructing a uniform code C that is error detecting for the channel $\sigma_{\boldsymbol{s}}(m,\ \ell_C)$ but the coded language $(pCs)^*$ is not error detecting for $\tau_{\boldsymbol{x}}(m,\ \ell_C + |ps| + t)$ with finite delay, for any $\boldsymbol{x} \in \{\boldsymbol{b},\ \boldsymbol{s}\}$ and for any nonnegative integer t. Let $v = (1^{|x|}0^{|x|})^{1+\lceil m/(2|x|)\rceil}$. Then, the words xv and vy are of the same length and they differ in at least $m+1$ positions; therefore, the code $C = \{xv,\ vy\}$ is error detecting for $\sigma_{\boldsymbol{s}}(m,\ \ell_C)$. Now assume $(pCs)^*$ is error detecting for $\tau_{\boldsymbol{x}}(m,\ \ell_C + |ps| + t)$ with delay d, for some nonnegative integers t and d. Let w be the word $(pxvs)^{d+3} = pxvyuxv(yuxv)^{d+1}s$ if τ contains δ, or $(pvys)^{d+2}$ otherwise. Consider also the word

$$z = \begin{cases} pxvuxv(yuxv)^{d+1}s, & \text{if } \tau \text{ contains } \delta \\ pxvys(pvys)^{d+1}, & \text{otherwise.} \end{cases}$$

In the first case, z obtains from w by deleting the word y. In the second case, z obtains from w by inserting the word x. Moreover, as the word z can be written as

$$\begin{cases} (pxvs)(pvys)(pvys)^d pvs, & \text{if } \tau \text{ contains } \delta \\ pxvs(pxvs)^d pxvys, & \text{otherwise} \end{cases}$$

the assumption about $(pCs)^*$ implies that $pxvs = pvys$ which in turn implies that $xv = vy$; a contradiction.

Finally, consider the case where $|sp| = 0$. Define the code $C = \{0^m1^m,\ 1^m0^m\}$ which is error detecting for the channel $\sigma_{\boldsymbol{s}}(m,\ \ell_C)$. Then, depending on τ, one can choose words y and u such that

$$\left(0^m1^m(0^m1^m)^d y \leftarrow (1^m0^m)^{d+1}u\right) \in \tau_{\boldsymbol{x}}(m,\ \ell_C)$$

for any nonnegative integer d. Hence, C^* is not error detecting for $\tau_{\boldsymbol{x}}(m,\ \ell_C + t)$ with finite delay, for any offset t. □

IV. The Case of Burst Errors

In this section, we provide a detailed analysis on the structure of uniform error-detectors for SID channels with burst errors. The analysis allows us to identify all such error-detectors, including all the ones that are optimal in terms of redundancy, delay, and offset.

Proposition 4: Let $m\boldsymbol{b}\tau$ be a burst-error specification and let $(p,\ s)$ be a pair of words with $|sp| > 0$. If $\mathrm{per}(sp) > m$, then $(p,\ s)$ is a uniform error-detector for $m\boldsymbol{b}\tau$ with offset 1 and delay 2. Moreover, if p is empty then $(p,\ s)$ has delay 1.

Proof: Suppose $\mathrm{per}(sp) > m$ and let C be any code in $\mathcal{U}_{m\boldsymbol{b}\sigma}$. We show that $(pCs)^*$ is error detecting for $\gamma = \tau_{\boldsymbol{b}}(m,\ \ell_C + |sp| + 1)$ with delay $d \in \{1,\ 2\}$, where $d = 1$ if $|p| = 0$. For this, assume $(pv_0s\cdots pv_dsy \leftarrow pw_0spw_1su) \in \gamma$ where $w_0,\ w_1,\ v_0,\ \ldots,\ v_d \in C$, $y \in X^*$ and $u \in (pCs)^*$. We need to show that $v_0 = w_0$ and $(pv_1s\cdots pv_dsy \leftarrow pw_1su) \in \gamma$. If there are no errors in pw_0s, then we are done. So assume there is a burst $(x' \leftarrow x) \in B_m(\tau)$, with $x \neq x'$, that affects pw_0s; that is, there are words $z_1,\ z_2,\ z_3,\ z_3'$ with $|z_1| < |pw_0s|$ and $|z_2| \geq \ell_C + |sp|$ such that $pw_0spw_1su = z_1xz_2z_3$ and $pv_0s\cdots pv_dsy = z_1x'z_2z_3'$ and $(z_3' \leftarrow z_3) \in \gamma$. Obviously, this is the only burst that affects pw_0s. Let $q = |x'| - |x|$. Then, $0 \leq |q| \leq m$. We distinguish six cases about the sign of q and the position of the burst $(x' \leftarrow x)$ in pw_0spw_1su.

Case 1: $q = 0$ and the burst occurs before the factor w_1 of pw_0spw_1su; that is, $|z_1x| \leq |pw_0sp|$ and pw_0sp results in pv_0sp, which implies that $v_0 = w_0$, as $pCsp$ is in $\mathcal{U}_{m\boldsymbol{b}\tau}$, and

$$(v_1s\cdots pv_dsy \leftarrow w_1su) \in \gamma.$$

Hence, also $(pv_1s\cdots pv_dsy \leftarrow pw_1su) \in \gamma$, as required.

Case 2: $q = 0$ and the burst affects the factor w_1 of pw_0spw_1su; that is, $|z_1x| > |pw_0sp|$. Moreover, as $|x| \leq m$ and $|z_1| < |pw_0s|$ and $|sp| > m$, we have that x is of the form s_2px_1 and x' is of the form s_2py_1, for some prefix x_1 of w_1 and for some prefix y_1 of v_1 and for some suffix s_2 of s. Now as there are no errors in z_2, it follows that the codewords v_1 and w_1 differ in at most $|x_1|$ symbols. Hence, $w_1 = v_1$ which implies that $x_1 = y_1$. This, however, contradicts the fact that $x \neq x'$.

In the next four cases we assume $|q| \neq 0$. Then, $sp = x_1u_1 = u_2x_2$ for some words $x_1,\ x_2,\ u_1,\ u_2$ such that $|x_1| = |x_2| = |q|$ and $|u_1| = |u_2| = |sp| - |q|$. As $\mathrm{per}(sp) > |q|$, it follows that $u_1 \neq u_2$. Let k be the largest position of u_1 and u_2 such that $u_1[k] \neq u_2[k]$.

Case 3: $q < 0$ and the burst occurs before the position $|q| + k$ of the factor $sp = x_1u_1$ of pw_0spw_1su; that is,

$$|z_1x| \leq |pw_0x_1u_1[1\cdots k-1]|.$$

In this case, $pw_0x_1u_1[1\cdots k-1]$ results in the prefix $pv_0u_2[1\cdots k-1]$ of $pv_0s\cdots pv_dsy$ and the next symbol $u_1[k]$ results in $u_2[k]$ with no errors. This contradicts the fact that $u_1[k] \neq u_2[k]$.

Case 4: $q < 0$ and the burst contains the position $|q| + k$ of the factor $sp = x_1u_1$ of pw_0spw_1su; that is,

$$|z_1x| \geq |pw_0x_1u_1[1\cdots k]|.$$

In this case, there can be no error in the prefix $x_1u_1[1\cdots k]$ of the second sp in pw_0spw_1sp. Then, $pw_0spw_1x_1$ results in pv_0spv_1 and the next k symbols $u_1[1\cdots k]$ result in the prefix $u_2[1\cdots k]$ of sp with no errors. This contradicts the fact that $u_1[k] \neq u_2[k]$.

Case 5: $q > 0$ and the burst occurs before the position k of $sp = u_2x_2$; that is,

$$|z_1x| \leq |pw_0u_2[1\cdots k-1]|.$$

In this case, $pw_0u_2[1\cdots k-1]$ results in the prefix $pv_0x_1u_1[1\cdots k-1]$ of $pv_0s\cdots pv_dsy$ and the next symbol $u_2[k]$ results in $u_1[k]$ with no errors. This contradicts the fact that $u_1[k] \neq u_2[k]$.

Case 6: $q > 0$ and the burst contains the position k of the factor $sp = u_2x_2$ of pw_0spw_1su; that is,

$$|z_1x| \geq |pw_0u_2[1\cdots k]|.$$

In this case, there can be no error in the prefix $u_2[1\cdots k]$ of the second sp in pw_0spw_1sp. Then, pw_0spw_1 results in $pv_0spv_1x_1$ and the next k symbols $u_2[1\cdots k]$ result in the suffix $u_1[1\cdots k]$ of sp with no errors. This contradicts the fact that $u_1[k] \neq u_2[k]$. □

For burst-error specifications, Propositions 3 and 4 provide a complete characterization of the structure of their uniform error-detectors. Moreover, it is possible to characterize precisely all such error-detectors $(p,\ s)$ having optimal redundancy. Indeed, Proposition 3 implies

that $m+1$ is the smallest value of $|p|+|s|$ and this value is possible when $\mathrm{per}(sp) \geq m+1$. But, as $\mathrm{per}(sp) \leq |sp|$, it follows that $(p,\ s)$ is an error-detector with optimal redundancy when $|sp| = \mathrm{per}(sp)$. This condition is equivalent to the constraint that the word sp is *unbordered [4] (or self-uncorrelated [14]):* no proper and nonempty prefix of sp is also a suffix of sp. It turns out that there are many unbordered words even for binary alphabets: about 27% of all binary words are unbordered, and this quantity increases for larger alphabets [13]. For example, the words 0^71 and 0^21^20101 are unbordered and, therefore, the pairs $(1,\ 0^7)$ and $(1^20101,\ 0^2)$ are uniform $7\boldsymbol{b}\tau$-detectors with delay at most 2 and offset 1 for every error type τ. We summarize the preceding remarks as follows.

Corollary 1: Let $m\boldsymbol{b}\tau$ be a burst-error specification. A pair of words $(p,\ s)$ is a uniform $m\boldsymbol{b}\tau$-detector with optimal redundancy if and only if $|sp| = m+1$ and the word sp is unbordered.

The next result concerns the question of whether an error-detector with optimal redundancy can have offset 0—the proof can be found in the Appendix.

Proposition 5: Let $m\boldsymbol{b}\tau$ be a burst-error specification containing at least two different basic error types and such that $m\boldsymbol{b}\tau \neq 1\boldsymbol{b}(\iota \odot \delta)$. If $(p,\ s)$ is an error-detector for $m\boldsymbol{b}\tau$ with redundancy $m+1$, then $(p,\ s)$ has offset greater than zero.

Consider an error type τ that permits insertion errors. In [7], it is shown that there exists no coded language that is error-detecting with delay 0 for any SID channel of the form $\tau_{\boldsymbol{s}}(m,\ \ell)$. The argument used for proving this statement can also be repeated for channels of the form $\tau_{\boldsymbol{b}}(m,\ \ell)$. On the other hand, for an error type τ in $\{\delta,\ (\sigma \odot \delta)\}$ it is possible to define $m\boldsymbol{b}\tau$-detectors with delay 0. In fact, we obtain a precise characterization of all those error-detectors which shows that the process of choosing a pair $(p,\ s)$ from a word $w = sp$ is important when it comes to the delay of $(p,\ s)$ as an error-detector. This observation follows also from Proposition 4, where an empty p ensures that $(p,\ s)$ has delay 1.

Lemma 6: Let $m\boldsymbol{b}\tau$ be a burst-error specification and let ℓ be an integer greater than m. For every $(z \leftarrow w)$ in the channel $\tau_{\boldsymbol{b}}(m,\ \ell)$ one has that

$$|w| - |z| \leq m\lfloor |w|/(\ell - 1 + m)\rfloor + \min\{m,\ |w|\%(\ell - 1 + m)\}$$

where $|w|\%(\ell - 1 + m)$ is the remainder of the integer division $|w|/(\ell - 1 + m)$.

Proposition 6: Let τ be an error type in $\{\delta,\ (\sigma \odot \delta)\}$. A pair of words $(p,\ s)$ is a uniform error-detector for $m\boldsymbol{b}\tau$ with delay 0 and offset 1 if and only if $s \in X^*a$ and $p \in (X \setminus \{a\})^m X^*$ for some symbol $a \in X$.

Proof: First we consider the "if" part. Assume that $s \in X^*a$ and $p \in (X \setminus \{a\})^m X^*$, and consider a code C in $\mathcal{U}_{m\boldsymbol{b}\sigma}$ and the channel $\gamma = \tau_{\boldsymbol{b}}(m,\ \ell_C + |sp| + 1)$. We show that $(pCs)^*$ is error detecting for γ with delay 0. For this, suppose $(pv_0sy \leftarrow pw_0su) \in \gamma$, where $v_0,\ w_0 \in C$, $y \in X^*$, and $u \in (pCs)^*$. Then, pw_0s results in a prefix z_0 of pv_0sy and u results in the word z with $z_0z = pv_0sy$; therefore, $(z \leftarrow u) \in \gamma$. If $|z_0| = |pw_0s|$, then $z = y$ and $z_0 = pv_0s$ which implies that $(y \leftarrow u)$ is in γ and that $pv_0s = pw_0s$ as C is in $\mathcal{U}_{m\boldsymbol{b}\sigma}$. In this case, we are done. Now assume that $|z_0| < |pw_0s|$. Then a burst of errors affects pw_0s and there are $d = |pw_0s| - |z_0|$ deletions in pw_0s. Moreover, $u = pw_1su'$ for some w_1 in C and $u' \in (pCs)^*$. Now pv_0s can be written as z_0z_1 such that $|z_1| = d$ and z_1 obtains from some prefix of pw_1s. In particular, $z_1[d]$, the last symbol of z_1, obtains from some symbol $p[i] \in X \setminus \{a\}$ of p for some i in $\{1,\ \ldots,\ m\}$. Moreover, it follows that z_1 obtains from $p[1\cdots i]$ through γ. If $\tau = \delta$, then $z_1[d] = p[i]$ which is impossible and the "if" part is complete for the case where $\tau = \delta$. So suppose that $\tau = (\sigma \odot \delta)$ and that $p[i]$ is substituted with $a = z_1[d]$.

First we argue that there is only one burst of errors in $pw_0sp[1\cdots i]$. Indeed, if there are two bursts then there will be at least $\ell_C + |sp|$ symbols of $pw_0sp[1\cdots i]$ between the bursts and, therefore, the first burst would occur in some prefix $p[1\cdots i_1]$ of p, where $i > i_1 \geq d$. Moreover, the second burst would occur after the prefix $p[1\cdots i_1]$ of $p[1\cdots i]$. But then the last symbol a of pv_0s would be equal to the symbol $p[d]$ of $p[1\cdots i_1]$ which is impossible. Hence, only one burst of $(\sigma \odot \delta)$ errors in $pw_0sp[1\cdots i]$ such that the last symbol of $p[1\cdots i]$ is substituted with $a = z_1[d]$. The beginning of the burst will be at or after the position $|p| + \ell_C - m + 2$ of pw_0 which implies that v_0 differs from w_0 in at most $m-2$ consecutive symbols. Hence, as C is in $\mathcal{U}_{m\boldsymbol{b}\sigma}$, it follows that $pw_0s = pv_0s$. It remains to show that $(y \leftarrow u)$ is in γ.

Consider in more detail the first burst, say

$$(u_1'z_1u_2' \leftarrow u_1p[1\cdots i]u_2)$$

that occurs in pw_0spw_1su' to get z_0z_1y, such that $pw_0s = xu_1$ for some word x, $z_0 = xu_1'$, $p[i+1\cdots|p|]w_1su' = u_2x_2$ for some word x_2, and $y \in u_2'X^*$ with $(y \leftarrow u_2x_2) \in \gamma$. Then, also $(y \leftarrow p[1\cdots i]u_2x_2)$ is in γ by deleting the prefix $p[1\cdots i]$ of $p[1\cdots i]u_2x_2$ and keeping the rest of the error bursts that existed in u_2x_2. Note that this is possible since the i deletions required for that already existed in the part $(u_1'z_1 \leftarrow u_1p[1\cdots i])$ of the burst $(u_1'z_1u_2' \leftarrow u_1p[1\cdots i]u_2)$—recall the d deletions in u_1 and the $|p[1\cdots i]| - |z_1|$ deletions in $p[1\cdots i]$ to obtain z_1. Hence, the "if" part is complete when we recall that $u = p[1\cdots i]u_2x_2$.

We turn now to the "only if" part. Assume that $(p,\ s)$ is a uniform error-detector for $m\boldsymbol{b}\tau$ with delay 0 and offset 1. Then, $|sp| > m$ by Proposition 3. Suppose it is not the case that $s \in X^*a$ and $p \in (X\setminus\{a\})^m X^*$. We distinguish three cases. In the first case, $s = \lambda$. Let $y = p[1\cdots m]$ and let $C = \{yz,\ zy\}$, where z is any word of length m differing from y in all positions; then C is in $\mathcal{U}_{m\boldsymbol{b}\sigma}$. Now it is the case that $(pzyp[m+1\cdots|p|]yz \leftarrow pyzpyz)$ is in $\tau_{\boldsymbol{b}}(m,\ \ell_C + |sp| + 1)$ by deleting the word y in $pyzpyz$, which implies that $pyz = pzy$; a contradiction.

In the second case, $s = s_1a$, for some word s_1, and $|p| \geq m$, but p is of the form p_1ap_2 with $|p_1| < m$. Let C be any code in $\mathcal{U}_{m\boldsymbol{b}\sigma}$ with $\ell_C = 2m|sp| - |sp| + m + |p_1|$, and let v be any word in C. The word $(pvs)^{2|sp|+3}$ is in $(pCs)^*$ and can be written as $(pvs_1a)(p_1ap_2vs)p_1z$, where $z = ap_2vs(pvs)^{2|sp|}$. Then, z can be written as $z_1\cdots z_{2|sp|}y$ such that $|y| = m$ and each z_i is of length $m + \ell_C + |sp|$. Now consider the word $(pvs_1a)p_2vsp_1z'$ such that $(pvs_1a)p_2vsp_1$ obtains from $(pvs_1a)(p_1ap_2vs)p_1$ by deleting the word ap_1 and z' obtains from $z_1\cdots z_{2|sp|}y$ by deleting the suffix y and the prefix of length m of every z_i. Then, $|z| - |z'| = (2|sp|+1)m$. Moreover, it follows that $((pvs_1a)p_2vsp_1z' \leftarrow (pvs_1a)(p_1ap_2vs)p_1z)$ is in the channel $\tau_{\boldsymbol{b}}(m,\ \ell_C + |sp| + 1)$ and the assumption about $(p,\ s)$ implies that $(p_2vsp_1z' \leftarrow p_1ap_2vsp_1z)$ is also in the channel. Let $w = p_1ap_2vsp_1z$. By Lemma 6

$$\begin{aligned}|w| &- |p_2vsp_1z'| \\ &\leq m\lfloor |w|/(\ell_C + |sp| + m)\rfloor + \min\{m,\ |w|\%(\ell_C + |sp| + m)\}\end{aligned}$$

which gives

$$1 + |p_1| + |z| - |z'| \leq m(2|sp|+1) + |p_1|.$$

This is impossible, however, when we recall that $|z| - |z'| = m(2|sp|+1)$.

In the third case, $s = s_1a$, for some word s_1, and $|p| < m$. This case can be eliminated by considering the code

$$C = \{a^{\ell_C},\ a^{\ell_C-(m+1)}b^{m+1}\}$$

in $\mathcal{U}_{m\boldsymbol{b}\sigma}$, with $\ell_C = 2|s|m + m - |s|$ and $b \in X \setminus \{a\}$, and the word $(pa^{\ell_C}s)^{2|s|+3}$ and then continuing as in the second case. We leave the details to the reader. □

V. The Case of Scattered Errors

Our first result gives a set of error-detectors for $m\boldsymbol{s}\tau$, for any error type τ. Then, we show that, for $\tau \in \{\iota, \delta, (\sigma \odot \delta), (\sigma \odot \iota)\}$, the result can be improved in terms of the redundancy of the error-detectors.

Proposition 7: Let $m\boldsymbol{s}\tau$ be a scattered-error specification and let (p, s) be a pair of words. If $p \in u^m X^*$, for some word u with $\mathrm{per}(u) > m$, then (p, s) is a uniform error-detector for $m\boldsymbol{s}\tau$ with delay 1 and offset $m|u|$.

Proof: Assume $p = u^m x$ for some word x and consider a code C in $\mathcal{U}_{m\boldsymbol{s}\tau}$. Let

$$\gamma = \tau_{\boldsymbol{s}}(m, \ell_C + |sp| + m|u|)$$

and take any $(pv_1su^m xv_2sy \leftarrow pw_1su^m xw_2su)$ in γ, where $v_1, v_2, w_1, w_2 \in C$ and $y \in X^*$ and $u \in (pCs)^*$. We need to show that $v_1 = w_1$ and $(u^m xv_2sy \leftarrow u^m xw_2su) \in \gamma$. We agree to write u_i for the ith factor u of u^m, where $i \in \{1, \ldots, m\}$. Suppose that pw_1s results in some word z_1, each u_i results in some word u_i', and xw_2su results in some word z_2 such that $z_1u_1' \cdots u_m' z_2 = pv_1su^m xv_2sy$. If $|z_1| = |pv_1s|$, then $z_1 = pv_1s$ and we are done. So assume that $d \geq 1$, where $d = \|z_1| - |pv_1s\|$; then, at least d errors occur in pw_1s and, therefore, $d \leq m$. Moreover, there is $k \in \{1, \ldots, m\}$ such that there is no error in u_k, namely, $u_k' = u_k$, and there are at most $m-d$ errors in $u_1 \cdots u_{k-1}$. Then, $z_1u_1' \cdots u_{k-1}'u_k'$ obtains from $pw_1su_1 \cdots u_{k-1}u_k$ and

$$\begin{aligned} |pw_1su_1 \cdots u_{k-1}u_k| - m &\leq |z_1u_1' \cdots u_{k-1}'u_k'| \\ &\leq |pw_1su_1 \cdots u_{k-1}u_k| + m. \end{aligned}$$

At the same time, as the word $z_1u_1' \cdots u_{k-1}'u_k'$ is a prefix of the word $pv_1su_1 \cdots u_k xv_2sy$, it follows that u_k' is a factor of $pv_1su_1 \cdots u_k xv_2sy$ such that, either u_k' begins in the factor u_k of the word $pv_1su_1 \cdots u_k xv_2sy$ (when $|z_1u_1' \cdots u_{k-1}'u_k'| \geq |pw_1su_1 \cdots u_{k-1}u_k|$), or u_k' ends in the factor u_k of the word $pv_1su_1 \cdots u_k xv_2sy$ (when $|z_1u_1' \cdots u_{k-1}'u_k'| \leq |pw_1su_1 \cdots u_{k-1}u_k|$). In either case, as $u_k' = u_k = u$ and the period of u is greater than m, it follows that the factor u_k' coincides with the factor u_k; therefore,

$$|z_1u_1' \cdots u_{k-1}'u_k'| = |pv_1su_1 \cdots u_{k-1}u_k|.$$

This implies that

$$(u_{k+1} \cdots u_m xv_2sy \leftarrow u_{k+1} \cdots u_m xw_2su) \in \gamma$$

and

$$(pv_1su_1 \cdots u_k \leftarrow pw_1su_1 \cdots u_k) \in \gamma.$$

The former relation gives $(u^m xv_2sy \leftarrow u^m xw_2su) \in \gamma$ and the latter one implies that $pv_1su_1 \cdots u_k$ obtains from $pw_1su_1 \cdots u_k$ using at most $m\boldsymbol{s}\tau$. Moreover, Lemma 2 implies that v_1 obtains from w_1 using at most $m\boldsymbol{s}\tau$ and, therefore, $v_1 = w_1$. □

The preceding statement allows us to define various error-detectors for any scattered-error specification. The most efficient error-detectors based on this method are those of the form (u^m, λ), where u is an unbordered word of length $m + 1$.

In [7], it is shown that the coded language $(0^m X^{\ell-2m-1}1)^*$ is error detecting for $\delta_{\boldsymbol{s}}(m, \ell)$ with delay 0, and for $\iota_{\boldsymbol{s}}(m, \ell)$ with delay 1. With the terminology of the present correspondence, it follows that the pair $(0^m, 1)$ is a uniform $m\boldsymbol{s}\delta$-detector with delay 0 and offset m, and a uniform $m\boldsymbol{s}\iota$-detector with delay 1 and offset m. Moreover, by Proposition 3, $(0^m, 1)$ is an error-detector with optimal redundancy. Unlike the case of burst error-detectors, where the offset can be independent of m, the offset m of $(0^m, 1)$ cannot be improved.

Proposition 8: If the pair $(0^m, 1)$ is a uniform $m\boldsymbol{s}\delta$-detector with delay 0 and offset t, then $t \geq m$.

Proof: For the sake of contradiction assume that the offset $t < m$ and consider the code $C = \{w_1, w_2, w_3\}$, where $w_1 = 0^m(10)^k$, $w_2 = 0^{m-1}(10)^k1$, $w_3 = 1^{2k+m}$, $k = 1 + \lfloor m/2 \rfloor$. Then, $C \in \mathcal{U}^m$. Now consider the words $0^m w_1 10^m w_3 1$ and $0^{m-1}w_1 1 w_3 1$. Then,

$$(0^{m-1}w_1 1 w_3 1 \leftarrow 0^m w_1 10^m w_3 1) \in \delta_{\boldsymbol{s}}(m, \ell_C + m + 1 + t).$$

Moreover, as $0^{m-1}w_1 1 w_3 1 \in 0^m w_2 1 X^*$ and $(0^m, 1)$ has delay 0, it follows that $w_2 = w_1$ which is impossible. □

Before we present uniform error-detectors for $(\sigma \odot \iota)$ and $(\sigma \odot \delta)$, we establish certain notation and utility results some of which are of interest in their own right.

Every nonempty word w can be written in the form $a_1^{n_1}a_2^{n_2} \cdots a_r^{n_r}$, where r and $n_1, \ldots, n_r$ are positive integers, $a_1, \ldots, a_r \in X$, and $a_i \neq a_{i+1}$ for all $i \in \{1, \ldots, r-1\}$. In this case, each factor $a_i^{n_i}$ of w is called a *run*. We use the symbol $w\langle i \rangle$ to denote the ith run of w. Now let r, n be positive integers with $r \geq 2$. An (r, n)-*alternating word* is a word w of the form $a_1^n a_2^n \cdots a_r^n$, where $w\langle i \rangle = a_i^n$ for all $i \in \{1, \ldots, r\}$. In the sequel, when we use the term (r, n)-alternating word we assume without mention that r and n are positive integers with $r \geq 2$.

Lemma 7: Assume that $X = \{0, 1\}$ and $a_1^n \cdots a_r^n$ is an (r, n)-alternating word. Consider the word $w = a_1^{n-t}a_2^n \cdots a_r^n$, where t is a positive integer with $t < n$, and suppose that a prefix p of $a_1^n \cdots a_r^n$ obtains from w using $k_1\boldsymbol{s}\sigma$ and $k_2\boldsymbol{s}\delta$, for some nonnegative integers k_1 and k_2. Then, in obtaining p from w, the following statements hold true about $k_1\boldsymbol{s}\sigma$ and $k_2\boldsymbol{s}\delta$.

1) If fewer than $|w\langle i \rangle|$ errors are used in $w\langle i \rangle$ for every $i = 1, \ldots, r$, then $k_1 \geq r - 1$.
2) If there is an $i \in \{1, \ldots, r-1\}$ such that $|w\langle i \rangle|$ errors are used in $w\langle i \rangle$, then $k_1 + k_2 \geq 2n - t$.

Proposition 9: Assume $X = \{0, 1\}$ and let k be a nonnegative integer. Let u be an (r, n)-alternating word and let v be a proper and nonempty suffix of u.

1) If a prefix of u obtains from v using $k\boldsymbol{s}(\sigma \odot \delta)$, then $k \geq \min\{r - 1, 2n - (|u| - |v|)\}$.
2) If v obtains from a prefix of u using $k\boldsymbol{s}(\sigma \odot \iota)$, then $k \geq \min\{r - 1, 2n - (|u| - |v|)\}$.

Proposition 10: Assume $X = \{0, 1\}$ and let k be a nonnegative integer. Let v be a nonempty word and let u be an (r, n)-alternating word.

1) If a prefix of vu obtains from u using $k\boldsymbol{s}(\sigma \odot \delta)$, then $k \geq \min\{r - 1, 2n - |v|\}$.
2) If u obtains from a prefix of vu using $k\boldsymbol{s}(\sigma \odot \iota)$, then $k \geq \min\{r - 1, 2n - |v|\}$.

Proposition 11: Let m be a positive integer, let $n = \lfloor m/2 \rfloor + 1$, and assume $X = \{0, 1\}$. For any (r, n)-alternating word $a_1^n \cdots a_r^n$ the following statements hold true.

1) If $r \geq m + 1$, then $(\lambda, a_1^n \cdots a_r^n)$ is a uniform error-detector for $m\boldsymbol{s}(\sigma \odot \delta)$ with delay 1 and offset $2n^2$.
2) If $r \geq 2n + 1$, then $(\lambda, a_1^n \cdots a_r^n)$ is a uniform error-detector for $m\boldsymbol{s}(\sigma \odot \delta)$ with delay 1 and offset 0.

Proof: We prove both statements simultaneously. Moreover, we only consider the case where $m \geq 2$ and, therefore, $n \geq 2$. One can verify that the claim also holds for $m = 1$. Let C be any code in $\mathcal{U}_{m\boldsymbol{s}\sigma}$, let $s = a_1^n \cdots a_r^n$, and let $\gamma = (\sigma \odot \delta)_{\boldsymbol{s}}(m, \ell_C + |s| + t)$, where $t = 2n^2$ if $r = 2n$, and $t = 0$ if $r \geq 2n+1$. Suppose $(v_1 s v_2 s y \leftarrow w_1 s w_2 s u) \in \gamma$ for some words $v_1, v_2, w_1, w_2 \in C$ and $y \in X^*$ and $u \in (Cs)^*$. We need to show that $v_1 = w_1$ and $(v_2 s y \leftarrow w_2 s u) \in \gamma$. If there are no errors in $w_1 s$, then we are done. If there are only substitution errors in $w_1 s$, then they all occur in w_1 and, therefore, v_1 obtains from w_1 using up to m substitutions which contradicts the fact that $C \in \mathcal{U}_{m\boldsymbol{s}\sigma}$. Now assume there is at least one deletion in $w_1 s$. In particular let d_1 be the number of deletions in w_1, let d_2 be the number of deletions in the first s of $w_1 s w_2 s$, and let d_3 be the number of deletions in w_2. Then $d_1 + d_2 \geq 1$.

If $d_1 \geq 1$, then w_1 results in a prefix of length $\ell_C - d_1$ of v_1 and s results in a prefix of xs, where x is the suffix of length d_1 of v_1. Then there are at least $\min\{r-1, 2n-|x|\} \geq \min\{m, 2n-d_1\}$ errors in s and, therefore, at least $d_1 + \min\{m, 2n-d_1\}$ errors in $w_1 s$ which is of length $\ell_C + |s|$. This is a contradiction, however, as $2n > m$. So in the rest of the proof we assume that $d_1 = 0$ and $d_2 \geq 1$; then, $d_2 + d_3 \leq m$. Also, $w_1 s w_2$ results in a prefix of length $\ell_C + |s| + \ell_C - (d_2 + d_3)$ of $v_1 s v_2$ and the second s of $w_1 s w_2 s$ results in a prefix of xs, where x is the suffix of length $d_2 + d_3$ of $v_1 s v_2$. Then, there are at least $2n - |x| = 2n - (d_2 + d_3)$ errors in the second s which gives at least $m+1$ errors in $s w_2 s$ whose length is $\ell_C + |s| + |s|$. Then a contradiction arises when $r = 2n$ and $t = 2n^2$, and the first statement is proved. So in the sequel we assume that $r \geq 2n+1$ and $t = 0$; that is, the channel γ is $(\sigma \odot \delta)_{\boldsymbol{s}}(m, \ell_C + |s|)$. Consider the run $s\langle i \rangle = a_i^n$ of s containing the first of the d_2 deletions.

Let w be the word $a_i^n a_{i+1}^n \cdots a_r^n w_2 a_1^n \cdots a_{i-1}^n$; then $|w| = \ell_C + |s|$ and w results in a prefix of the word $a_i^n a_{i+1}^n \cdots a_r^n v_2 a_1^n \cdots a_{i-1}^n$ using $k\boldsymbol{s}(\sigma \odot \delta)$, for some integer k with $d_2 + d_3 \leq k < 2n$. We shall show that our assumptions lead to a contradiction. First note that the word $z = a_i^{n-1} a_{i+1}^n \cdots a_r^n w_2 a_1^n \cdots a_{i-1}^n$ obtains from w using the first deletion in a_i^n and then a prefix of $a_i^n \cdots a_r^n v_2 a_1^n \cdots a_{i-1}^n$ obtains from z using $(k-1)\boldsymbol{s}(\sigma \odot \delta)$. Hence, if $i \leq r-1$ then there are at least $\min\{r-i, 2n-1\}$ errors in the prefix $a_i^{n-1} a_{i+1}^n \cdots a_r^n$ of z. Also, note that $a_1^n \cdots a_{i-1}^n$ results in a prefix of $v a_1^n \cdots a_{i-1}^n$, where v is the suffix of length $d_2 + d_3$ of v_2; that is, $|v| = d_2 + d_3$. Hence, if $i \geq 3$, there are at least $\min\{i-2, 2n-(d_2+d_3)\}$ errors in the suffix $a_1^n \cdots a_{i-1}^n$ of z.

If $i \leq 2$, then $k - 1 \geq \min\{r-2, 2n-1\}$ which implies that $k \geq 2n$; a contradiction. If $i \geq 3$ and $i - 2 \geq 2n - (d_2 + d_3)$, then $k \geq (d_2+d_3) + 2n - (d_2+d_3)$ which is a contradiction. Finally, if $i \geq 3$ and $i - 2 < 2n - (d_2 + d_3)$, then $k - 1 \geq \min\{r-i, 2n+1\} + (i-2)$ which implies that $k \geq 2n$; a contradiction. □

Using similar arguments as above, one can verify that also the following statement holds true.

Proposition 12: Let m be a positive integer, let $n = \lfloor m/2 \rfloor + 1$, and assume $X = \{0, 1\}$. For any (r, n)-alternating word $a_1^n \cdots a_r^n$, we have the following statements.

1) If $r \geq m+1$, then $(\lambda, a_1^n \cdots a_r^n)$ is a uniform error-detector for $m\boldsymbol{s}(\sigma \odot \iota)$ with delay 1 and offset $2n^2$.
2) If $r \geq 2n+1$, then $(\lambda, a_1^n \cdots a_r^n)$ is a uniform error-detector for $m\boldsymbol{s}(\sigma \odot \iota)$ with delay 1 and offset 0.

VI. Discussion

We have presented an analysis of the method of separators for detecting synchronization (and substitution) errors in the messages of a coded language. For the case of burst errors, we were able to obtain a simple necessary and sufficient condition on the structure of the separators. It would be interesting to find such a condition on separators that detect scattered errors as well. This would allow us to evaluate the various separators for scattered errors designed in the correspondence. We note that this question is related to the problem of frame synchronization in the presence of scattered substitution errors—see [15] for details.

Regarding separators for error correction [1], it is straightforward to define the parameters of redundancy and offset as in this correspondence. On the other hand, the parameter of delay should be defined with some care as the definition of "error correction with finite delay" given in [1] involves automata with output (finite-state machines). If we ignore that parameter for now, the results of [1] on "error-correctors" (separators for error correction) can be rephrased as follows.

1) The pair $(0^m, 1^m)$ is a uniform error-corrector for $m\boldsymbol{s}\delta$ with redundancy $2m$ and offset 1.
2) The pair $(\lambda, 1^m 0^m)$ is a uniform error-corrector for $m\boldsymbol{s}\iota$ with redundancy $2m$ and offset $2m$.
3) The pair $(\lambda, (1^{m+1}0^{m+1})^{m+1}1^m)$ is a uniform error-corrector for $m\boldsymbol{s}(\sigma \odot \iota \odot \delta)$ and for $m\boldsymbol{s}(\iota \odot \delta)$ with redundancy $2(m+1)^2 + m$ and offset $2(m+1)^2 + m$.

If we ignore lower order terms, it follows that, for the same error specification, Levenshtein's error-correctors are twice as long as the error-detectors designed here. Intuitively, this observation is consistent with the view that the amount of redundancy for error correction is (roughly) twice the amount for error detection.

Appendix

This appendix contains the proofs of several lemmas and propositions.

Proof of Lemma 1: The "only if" part is obvious. For the "if" part, first assume that $\tau \in \{(\sigma \odot \iota), (\sigma \odot \delta)\}$. Then the statement follows easily when we note that, if v' obtains from v using at most $m\boldsymbol{x}\tau$, only substitution errors can occur in v. Now assume that $\tau \in \{(\iota \odot \delta), (\sigma \odot \iota \odot \delta)\}$; then $\boldsymbol{x} = \boldsymbol{b}$ and $v = pus$ and $v' = pu's$ such that $(u' \leftarrow u)$ is in $B_m(\tau)$. As $|v| = |v'|$, one has that $|u| = |u'|$ which implies that $D_\sigma(v, v') \leq m$ and, therefore, $(u' \leftarrow u)$ is in $B_m(\sigma)$. Hence, v' obtains from v using at most $m\boldsymbol{x}\sigma$. □

Proof of Lemma 2: If $v = v'$ then the statement is obvious. So assume $v \neq v'$; then $\tau \notin \{\iota, \delta\}$ and, either $\tau = \sigma$ or τ contains at least two different error types. If $m\boldsymbol{x}\tau$ is other than $m\boldsymbol{s}(\iota \odot \delta)$ and $m\boldsymbol{s}(\sigma \odot \iota \odot \delta)$, then the statement follows easily from Lemma 1. If $m\boldsymbol{x}\tau$ is either of $m\boldsymbol{s}(\iota \odot \delta)$ and $m\boldsymbol{s}(\sigma \odot \iota \odot \delta)$, then the statement follows from the fact that $D_\tau(v, v') = D_\tau(pvs, pv's)$ [1]. □

Proof of Lemma 3: For the first claim, we write u in the form xy for some words x and y with $|x| = |u'|$. Then the claim follows when we note that u' obtains from u by deleting y and substituting $D_\sigma(u', x)$ symbols in x. For the second claim, if no deletions are used to obtain u' from u then we are done. If at least one deletion is used then there must be at least two insertions in u—so that $|u'| > |u|$—and, therefore, $|u'| - |u| \leq m-2$. In this case, u' can be written as xy with $|x| = |u|$ and $|y| \leq m-2$. Then the claim follows when we note that u' obtains from u by substituting $D_\sigma(u, x)$ symbols in u and then inserting y at the end.

For the last claim, let $X = \{0, 1\}$ and let $k \leq m$ be such that u' obtains from u using $k\boldsymbol{s}(\sigma \odot \iota \odot \delta)$. Without loss of generality, suppose $u[1] = 0$. First assume that, for all $i = 1, \ldots, |u|$, $u'[i] \neq u[i]$ and $u'[i+1] \neq u[i]$. Then it follows that $u = 0^{|u|}$ and $u' = 1^{|u|+1}$.

In this case, to obtain u' from u, all symbols of u must be deleted or substituted and at least a 1 must be inserted. Hence, $k \geq |u| + 1$ which implies $|u| < m$. Then, u' obtains from u by substituting u with $1^{|u|}$ and inserting a 1 at the end. Therefore, u' obtains from u using at most $m\boldsymbol{s}(\sigma \odot \iota)$. Now assume that there is $i \in \{1, \ldots, |u|\}$ such that $u'[i] = u[i]$ or $u'[i+1] = u[i]$. In the former case, suppose i is the smallest with this property. Then, u' can be written as $u'_1 u[i] u'_2$ and u as $u_1 u[i] u_2$ such that $|u'_1| = |u_1|$ and $D_\sigma(u'_1, u_1) = |u_1|$. In the latter case, suppose i is the largest with the property $u'[i+1] = u[i]$. Then, u' can be written as $u'_1 u[i] u'_2$ and u as $u_1 u[i] u_2$ such that $|u'_2| = |u_2|$ and $D_\sigma(u'_2, u_2) = |u_2|$. In either case, u' obtains from u using at most $(|u_1| + |u_2|)\boldsymbol{s}\sigma$ and then $1\boldsymbol{s}\iota$, which proves the claim. □

Proof of Proposition 2: Assume that

$$B_m(\sigma \odot \iota \odot \delta) = B_m(\sigma \odot \delta) \cup B_m(\sigma \odot \iota).$$

First, we show that $m \leq 4$. Indeed, suppose $m \geq 5$. Then there are positive integers q and r such that $m = 3q + r$ and $r \geq 2q$ (for example, $q = 1$ and $r = m - 3$). Consider the words $u = 0^{q+r}1^q0^q$ and $u' = 1^{2q+r}0^{q+r}1^q$. Then, $|u| = m$, $|u'| = |u| + q + r$, and u' obtains from u using $(2q + r)\boldsymbol{s}\iota$ and $q\boldsymbol{s}\delta$; therefore, $(u' \leftarrow u) \in B_m(\sigma \odot \iota \odot \delta)$. Suppose now that u' obtains from u using $k\boldsymbol{s}(\sigma \odot \iota)$, for some integer k with $q + r \leq k \leq m$. Then also u obtains from u' using $k\boldsymbol{s}(\sigma \odot \delta)$ which implies that there is a word z such that z obtains from u' using $(q + r)\boldsymbol{s}\delta$ and u obtains from z using $(k - q - r)\boldsymbol{s}\sigma$—see [6] for instance. Then, z is of the form $1^{2q+r-i}0^{q+r-j}1^{q-t}$ for some nonnegative integers i, j, t with $t \leq q$ and $i + j + t = q + r$. By distinguishing two cases depending on whether $|1^q0^q| \leq |0^{q+r-j}1^{q-t}|$, it follows that $D_\sigma(z, u) > 2q$; therefore, $k > q + r + 2q = m$ which is a contradiction. Hence, $m \leq 4$. To complete the "only if" part we need to show that, if $m \in \{3, 4\}$, then X must be $\{0, 1\}$. For the sake of contradiction, suppose there is a symbol $a \in X \setminus \{0, 1\}$ and consider the words $u = 01a^{m-2}$ and $u' = 1a0^{m-1}$, with $m \in \{3, 4\}$. Then, u' obtains from u using $m\boldsymbol{s}(\sigma \odot \iota \odot \delta)$ but u cannot obtain from u' using at most $m\boldsymbol{s}(\sigma \odot \delta)$; therefore, u' cannot obtain from u using at most $m\boldsymbol{s}(\sigma \odot \iota)$.

For the converse, we note first that, obviously

$$B_m(\sigma \odot \delta) \cup B_m(\sigma \odot \iota) \subseteq B_m(\sigma \odot \iota \odot \delta).$$

For the reverse inclusion, let $(u' \leftarrow u) \in B_m(\sigma \odot \iota \odot \delta)$ and consider the following cases.

Case 1: $m = 1$. One verifies by inspection that $(u' \leftarrow u) \in B_1(\sigma \odot \delta) \cup B_1(\sigma \odot \iota)$.

Case 2: $m = 2$. In this case, $2m - 2 = 2$ and it follows from Proposition 1 that $(u' \leftarrow u) \in B_2(\sigma \odot \delta) \cup B_2(\sigma \odot \iota)$.

Case 3: $m = 3$ and $X = \{0, 1\}$. We use Lemma 3. If $|u'| \leq |u|$ then $(u' \leftarrow u) \in B_m(\sigma \odot \delta)$. If $|u'| > |u|$ there are two subcases. If no deletions are used to obtain u' from u then clearly $(u' \leftarrow u) \in B_m(\sigma \odot \iota)$. If a deletion is used in obtaining u' from u then $|u'| = |u| + 1$ and, therefore, $(u' \leftarrow u) \in B_m(\sigma \odot \iota)$.

Case 4: $m = 4$ and $X = \{0, 1\}$. We use again Lemma 3. If $|u'| \leq |u|$ then $(u' \leftarrow u) \in B_m(\sigma \odot \delta)$. So assume $|u'| > |u|$. If $|u'| = |u|+1$ then we are done as in Case 3. If $|u'| > |u| + 1$ then, as $|u| \leq m = 4$, it follows that $|u'| = |u| + 2$. If no deletion is used in u then we are done. If a deletion is used in u then it is the only one and there are also exactly three insertions to obtain u'. Hence, u' obtains from u using $3\boldsymbol{s}\iota$ and $1\boldsymbol{s}\delta$. Then, using a long case distinction, one can verify that again $(u' \leftarrow u) \in B_m(\sigma \odot \iota)$. □

Proof of Proposition 5: The assumptions about $m\boldsymbol{b}\tau$ imply that τ is in $\{(\sigma \odot \delta), (\sigma \odot \iota), (\iota \odot \delta), (\sigma \odot \iota \odot \delta)\}$. Assume that (p, s) is an $m\boldsymbol{b}\tau$-detector with redundancy $m + 1$, offset 0, and delay d, for some nonnegative integer d. We shall obtain a contradiction by constructing a code $C \in \mathcal{U}_{m\boldsymbol{b}\tau}$ such that $(pCs)^*$ is not error detecting for $\tau_{\boldsymbol{b}}(m, \ell_C + |ps|)$ with delay d. First, write the word sp in the form axb, for some $a, b \in X$ and $x \in X^*$, and let $C = \{xby, yax\}$, where $y = 1^m0^m1^m$. Then, C is error detecting for $\tau_{\boldsymbol{b}}(m, \ell_C)$ and, therefore, $C \in \mathcal{U}_{m\boldsymbol{b}\tau}$. Now let w be the word $(pxbys)^{d+3}$ if τ contains δ, or $(pyaxs)^{d+3}$ otherwise; then

$$w = pxby(axb)xby(axb)(xby(axb))^d xbys$$

or

$$w = pyax(axb)yax(axb)(yax(axb))^d yaxs.$$

Consider also the word $z =$

$$\begin{cases} pxby(a)xby(axa)(xby(axa))^d xbys, & \text{if } \tau \text{ contains } \delta \\ pyax(axbxb)yax(bxb)(yax(bxb))^d yaxs, & \text{otherwise.} \end{cases}$$

Then, z is equal to either

$$(pxbys)(pyaxs)(pyaxs)^d pys$$

or

$$(pyaxs)(pxbys)(pxbys)^d pxbyaxs.$$

If τ contains δ, then z obtains from w by deleting the suffix xb in the prefix $pxbyaxb$ of w, and then by transforming $d + 1$ b's in w to a's using the errors permitted. If τ does not contain δ, then z obtains from w by inserting the word xb at the end of the prefix $pyaxa$ of w, and then by transforming $d + 1$ a's in w to b's using the errors permitted. In either case, it follows that $(z \leftarrow w) \in \tau_{\boldsymbol{b}}(m, \ell_C + m + 1)$ which implies that $pyaxs = pxbys$; a contradiction. □

Proof of Lemma 6: Let $q = \lfloor |w|/(\ell - 1 + m) \rfloor$. Then w can be written as $w_1 \cdots w_q x$ with $|x| = |w| \% (\ell - 1 + m)$ and $|w_i| = \ell - 1 + m$. Also z can be written as $w'_1 \cdots w'_q x'$ such that $(x' \leftarrow x)$ and every $(w'_i \leftarrow w_i)$ are in the channel. As $|w_i| = \ell - 1 + m$ and $\ell > m$, in each w_i either there are at most two bursts of errors, or three bursts of errors that involve only insertions—the latter case is possible only when $\ell = m + 1$. It follows then that the number of deleted symbols in the burst(s) is at most m and, therefore, $|w_i| - |w'_i| \leq m$. Now for the word x we have two cases. If $|x| \leq m$ then at most $|x|$ symbols can be deleted in x and, therefore, $|x| - |x'| \leq |x|$. If $|x| > m$ then, as $|x| < \ell - 1 + m$, one can use the same argument as above to infer that $|x| - |x'| \leq m$. In either case, $|x| - |x'| \leq \min\{|x|, m\}$. Hence, $|w| - |z| \leq qm + \min\{|x|, m\}$ as required. □

Proof of Lemma 7: For the first statement, note first that there is a word y such that y obtains from w using $k_2\boldsymbol{s}\delta$ and p obtains from y using $k_1\boldsymbol{s}\sigma$. Then, y is of the form $a_1^{n-s_1}a_2^{n-s_2}\cdots a_r^{n-s_r}$, where each s_i is a nonnegative integer with $s_i < n$ and $n - s_1 \leq n - t$. Then, to get p from y using $k_1\boldsymbol{s}\sigma$, consider the fact that at least one symbol from each run of y will appear in p with no error. It follows then that at least the first symbol of each run $y\langle i \rangle$, with $i \in \{2, \ldots, r\}$, will be substituted. Hence, $k_1 \geq r - 1$.

For the second statement, there is a word y such that y obtains from w using $|w\langle i \rangle|$ errors in $w\langle i \rangle$ and p obtains from y using $(k_1 + k_2 - |w\langle i \rangle|)\boldsymbol{s}(\sigma \odot \delta)$. If $i = 1$, then $w\langle i \rangle = a_1^{n-t}$ results in a_2^s, for some $s \in \{0, \ldots, n-t\}$; therefore, y begins with a prefix of the form $a_2^s a_2^n$. As $a_1 \neq a_2$, obtaining p from y requires at least n errors in $a_2^s a_2^n$. Hence, $k_1 + k_2 \geq (n-t) + n = 2n - t$. If $i \geq 2$ then $w\langle i \rangle = a_i^n$ and y will contain a factor of the form $a_{i-1}^{n-t}a_{i-1}^s a_{i-1}^n$, where a_{i-1}^s results from $w\langle i \rangle$ and $s \in \{0, \ldots, n\}$. Thus, to get p from y at least $n - t + s$ errors are needed in the factor a_{i-1}^{2n-t+s} of y. Hence, $k_1 + k_2 \geq 2n - t$. □

Proof of Proposition 9: Assume $u = pv$ with $p, v \neq \lambda$. For the first part, suppose that v results in a prefix of u using $k\boldsymbol{s}(\sigma \odot \delta)$. If $2n \leq |u| - |v|$ we are done. So assume $|u| - |v| < 2n$. If $|u| - |v| = n + t$, for some integer t with $0 \leq t < n$, then $v = a_2^{n-t}a_3^n \cdots a_r^n$

results in a prefix of $a_1^n \cdots a_r^n$ using $k\boldsymbol{s}(\sigma \odot \delta)$. But, as $a_1 \neq a_2$, there must be $n-t$ errors in the prefix a_2^{n-t} of v. Hence, $k \geq n-t = 2n-(n+t)$ as required. Now assume $|u|-|v| = t$ with $t < n$; then $v = a_1^{n-t} a_2^n \cdots a_r^n$ results in a prefix of $a_1^n \cdots a_r^n$ using $k_1\boldsymbol{s}\sigma$ and $k_2\boldsymbol{s}\delta$, where $k_1 + k_2 = k$. We consider three cases. In the first case, no run $v\langle i\rangle$ of v has $|v\langle i\rangle|$ errors. Then, $k_1 \geq r-1$ which implies $k \geq r-1$ as required. In the second case, some run $v\langle i\rangle$, with $i \in \{1, \ldots, r-1\}$, has $|v\langle i\rangle|$ errors. Then, $k \geq 2n-t$ as required. In the third case, there are $|v\langle n\rangle|$ errors in the last run a_r^n of v and, therefore, $k \geq n$. If $r = 2$ we are done. If $r \geq 3$ then note that the word $a_1^{n-t} a_2^n \cdots a_{r-1}^n$ results in a prefix of $a_1^n a_2^n \cdots a_{r-1}^n$ using $(k-n)\boldsymbol{s}(\sigma \odot \delta)$ and, by Lemma 7, it follows that $k-n \geq \min\{r-2,\ n-t\}$. Hence, $k \geq \min\{r-1,\ 2n-t\}$ as required.

For the second part, assume that v results from a prefix of u using $k\boldsymbol{s}(\sigma \odot \iota)$. Then the prefix of u results from v using $k\boldsymbol{s}(\sigma \odot \delta)$ and the claim follows from the first part. □

Proof of Proposition 10: We only show the first part; the second part follows easily as in the previous proposition. If u results in a prefix of v then at least $|u| - |v|$ deletions must be used in u; therefore, $k \geq 2n - |v|$ as required. So assume that u results in vp, where p is a nonempty prefix of u. Then, there is a prefix x of u, with $|x| \geq |v|$, such that v obtains from x using $k_1\boldsymbol{s}(\sigma \odot \delta)$, and p obtains from y using $k_2\boldsymbol{s}(\sigma \odot \delta)$, where $k_1 + k_2 = k$ and y is such that $u = xy$. As there must be exactly $|x| - |v|$ deletions in $|x|$, we have that $k_1 \geq |x| - |v|$. If $|x| \geq 2n$ then we are done. So assume $|x| < 2n$. If $|x| = n + t$ with $0 \leq t < n$ then $x = a_1^n a_2^t$ and $y = a_2^{n-t} a_3^n \cdots a_r^n$ which results in a prefix of $a_1^n \cdots a_r^n$. Then, as $a_1 \neq a_2$, there must be $n-t$ errors in the prefix a_2^{n-t} of y and, therefore, $k_2 \geq n-t$. Hence, $k_1 + k_2 \geq 2n - |v|$ as required. Now assume $|x| < n$. Then, $x = a_1^{|x|}$, $y = a_1^{n-|x|} a_2^n \cdots a_r^n$, and a prefix of $a_1^n \cdots a_r^n$ obtains from y using $k_2\boldsymbol{s}(\sigma \odot \delta)$. This implies that $k_2 \geq \min\{r-1,\ 2n-|x|\}$, which, in turn, implies that $k_1 + k_2 \geq \min\{r-1,\ 2n-|v|\}$ as required. □

ACKNOWLEDGMENT

The authors wish to thank the anonymous referees for suggesting references [12] and [15].

REFERENCES

[1] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Sov. Phys.–Dokl.: Cybern. Contr. Theory*, vol. 10, pp. 707–710, 1966.

[2] F. F. Sellers, Jr., "Bit loss and gain correction code," *IEEE Trans. Inform. Theory*, vol. IT–8, pp. 35–38, Jan. 1962.

[3] H. C. Ferreira, W. A. Clarke, A. S. J. Helberg, K. A. S. Abdel-Ghaffar, and A. J. H. Vinck, "Insertion/deletion correction with spectral nulls," *IEEE Trans. Inform. Theory*, vol. 43, pp. 722–732, Mar. 1997.

[4] M. Lothaire, *Algebraic Combinatorics on Words*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[5] L. J. Guibas and A. Odzylko, "Periods in strings," *J. Combin. Theory, Ser. A*, vol. 30, pp. 19–42, 1981.

[6] S. Konstantinidis, "Relationships between different error-correcting capabilities of a code," *IEEE Trans. Inform. Theory*, vol. 47, pp. 2065–2069, July 2001.

[7] S. Konstantinidis and A. O'Hearn, "Error-detecting properties of languages," *Theor. Comput. Sci.*, vol. 276, pp. 355–375, 2002.

[8] J. Berstel and D. Perrin, *Theory of Codes*. Orlando, FL: Academic, 1985.

[9] W. W. Peterson and E. J. Weldon, *Error Correcting Codes*, 2nd ed. Cambridge, MA: MIT Press, 1972.

[10] V. I. Levenshtein, "On perfect codes in the deletion/insertion metric," *Discr. Math. and Appl.*, vol. 2, pp. 241–258, 1992.

[11] G. Tenengolts, "Nonbinary codes correcting single deletion or insertion," *IEEE Trans. Inform. Theory*, vol. IT–30, pp. 766–769, Sept. 1984.

[12] A. S. J. Helberg and H. C. Ferreira, "On multiple insertion/deletion correcting codes," *IEEE Trans. Inform. Theory*, vol. 48, pp. 305–308, Jan. 2002.

[13] L. J. Guibas and A. Odzylko, "String overlaps, pattern matching, and nontransitive games," *J. Combin. Theory, Ser. A*, vol. 30, pp. 183–208, 1981.

[14] H. Morita, A. J. van Wijngaarden, and A. J. H. Vinck, "On the construction of maximal prefix-synchronized codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 2158–2166, Nov. 1996.

[15] P. Bylanski and D. G. W. Ingram, *Digital Transmission Systems*. Herts, U.K.: Peter Peregrinus Ltd., 1976.

Stavros Konstantinidis (M'02) received the B.Sc. degree in mathematics from the University of Athens, Athens, Greece, in 1988 and the M.Sc. and Ph.D. degrees in computer science from the University of Western Ontario, London, ON, Canada, in 1992 and 1996, respectively.

From 1996 to 1998, he was an Assistant Professor of Computer Science at the University of Lethbridge, Lethbridge, AB, Canada. In 1998, he joined the Department of Mathematics and Computing Science at Saint Mary's University, Halifax, NS, Canada, where he currently is an Associate Professor. His research interests include automata, formal languages, and coding theory.

Steven Perron is an undergraduate student at Saint Mary's University, Halifax, NS, Canada. He expects to receive the B.Sc. degree in May 2003. He plans to go to graduate school and study computer science.

His research interests include complexity theory, coding theory, and algorithms.

L. Amber Wilcox-O'Hearn received the B.A. degree (with honors) in Russian language from the University of Waterloo, Waterloo, ON, Canada in 1997, and the B.Sc. degree in computing science from Saint Mary's University, Halifax, NS, Canada, in 2002, and is currently a graduate student in the Department of Computer Science at the University of Toronto, Toronto, ON, Canada.

Her research interests include computational linguistics, especially semantics and its relations to information theory and the theory of codes.