A Survey of Error-Correcting Codes for Channels With Symbol Synchronization Errors

Hugues Mercier, *Member, IEEE*, Vijay K. Bhargava, *Fellow, IEEE* and Vahid Tarokh, *Fellow, IEEE*

***Abstract*—We present a comprehensive survey of error-correcting codes for channels corrupted by synchronization errors. We discuss potential applications as well as the obstacles that need to be overcome before such codes can be used in practical systems.**

***Index Terms*—Error-correcting codes, deletion errors, duplication errors, insertion errors, synchronization errors.**

I. INTRODUCTION

EARLY communication systems designers were preoccupied with symbol and word synchronization as much as they were preoccupied with additive noise. Samuel Morse, when designing his eponymous code in the 1830s, allowed three time units of silence for a letter space and six time units of silence for a word space, a high price to pay to reduce the synchronization errors and facilitate the decoding of messages transmitted over telegraph lines. Today, synchronization problems remain an integral part of technological systems operating in environments affected by uncertainties in timing, or time noise. These include data storage systems like magnetic and optical recording [1], semiconductor devices and integrated circuits [2], and synchronous digital communication networks [3]. Time noise introduces insertions and deletions of symbols for the recipient, and as a result, systems corrupted by timing errors do not know their exact position when processing data.

Synchronization and additive noise are usually treated as different problems and overcome using different techniques. This being said, both have the same effect on communication channels, i.e., reducing their capacity. Unfortunately, although it has early [4] and often been conjectured that error-correcting codes capable of correcting timing errors could improve the overall performance of communication systems, they are quite challenging to design, which partly explains why a large collection of synchronization techniques not based on coding were developed and implemented over the years. Channels corrupted by synchronization errors have memory, hence the techniques developed for memoryless channels and additive noise can seldom be used straightforwardly.

Manuscript received 15 October 2008; revised 12 May 2009. This research project was supported in part by the Fonds québécois de la recherche sur la nature et les technologies (FQRNT).

H. Mercier and V. Tarokh are with the Harvard School of Engineering and Applied Sciences, Cambridge, MA 02138, USA (email: {huguesm,vahid}@seas.harvard.edu).

V. K. Bhargava is with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada (email: vijayb@ece.ubc.ca).

Digital Object Identifier 10.1109/SURV.2010.020110.00079

Tools developed for error-correcting codes robust against timing errors could also be of interest for a number of problems that can be tackled using synchronization models, like speech [5] and pattern recognition [6]. Deletion channels with large alphabets can model packet-switched communication networks like the Internet, where packets are either lost or dropped due to buffer overflow at finite-buffer queues [7]. When codes with variable-length symbols are used to transmit information, additive noise can cause synchronization errors by changing the boundaries of the symbols. One such instance is differential pulse-position modulation over wireless optical networks [8]. Insertions and deletions can also occur for a large class of distributed problems involving reconciliation of correlated data [9] such as reconciliation of nucleotide sequences in DNA molecules [10], remote data storage [11], and synchronization of mobile data [12].

In this article, we present a comprehensive survey of synchronization error-correcting codes, their potential applications, as well as the obstacles that need to be overcome before such codes can be used in practical systems. Our motivation for such an overview is threefold. Firstly, most comprehensive books on error-correcting codes do not even mention synchronization-correcting codes, and the very few who do, like [13], only do so in passing. A recent survey by Sloane [14] has frequently been used as a reference source, but its scope is limited to block synchronization-correcting codes able to correct a single deletion error per block. Our opinion that there was a need for a list of pointers to relevant references was somehow reinforced when we realized that several results recently presented seemed unaware of overlapping work done as far as forty years ago. Secondly, although major theoretical advances were made, we are still far from a unified theory of synchronization and many fundamental questions remain unanswered. The lack of mathematical tools makes it difficult to construct good coding schemes and efficient decoding algorithms. Likewise, the lack of understanding of most channels with synchronization errors makes it impossible to assess how good the existing codes are. Finally, most of the early work on synchronization-correcting codes was of little interest to communication systems designers. However, modern communication systems require more and more stringent synchronization constraints. Using increasingly faster and cheaper hardware as well as smaller high-capacity data storage devices, it appears that improved coding schemes might be able to complement existing synchronization techniques.

The outline of the article is as follows. Section II begins with a brief introduction to error-correcting codes. It also describes the types of synchronization errors surveyed in this

1553-877X/10/$25.00 © 2010 IEEE

article and the challenges they introduce. The synchronization-correcting codes are surveyed in Section III. The codes use a wide range of ideas, techniques and tools, some adapted from their counterparts for noisy channels without synchronization errors, others specifically developed to deal with synchronization errors. Due to the lack of connecting thread among the codes, we tried to regroup similar code families. We start with the seminal work of Levenshtein on algebraic block codes in Subsection III-A. This is followed by nonbinary and perfect codes in III-B, bursts-correcting codes in III-C, synchronizable codes in III-D, marker codes in III-E, codes for weak synchronization errors in III-F, convolutional codes in III-G, spectral-null codes in III-H, expurgated codes in III-I, and codes over random synchronization channels in III-J. The preceding categories are of course not mutually exclusive, one might say arbitrary, so we used them loosely and tried to present the results as coherently as possible. We conclude by discussing future directions in Section IV.

II. Preliminaries

In 1948, Shannon [15] showed that information could be encoded and transmitted (or preserved) reliably in the presence of noise at any rate below the capacity of the channel. Unfortunately, although Shannon showed the existence of "good" codes, he gave little insight as how to construct or decode them. Since Shannon's breakthrough, a lot of research was done to construct good error-correcting codes with efficient encoding and decoding algorithms, and today reliable and efficient codes are used in a wide range of digital systems. An overview of error-correcting codes and their applications can be found in one of several books on the subject, like [16] and references therein.

The vast majority of error-correcting codes assume that the transmitter and receiver are synchronized. In particular, the receiver knows where or when the received message starts, the transmitted and received symbols have the same length or duration, and the receiver samples each symbol perfectly. In this setting, the redundancy introduced by the codes is used to correct symbols corrupted by the channel.

In this article, we consider channels that introduce synchronization errors between the transmitted and received versions of a message, and survey the codes designed to correct such errors. As mentioned in the introduction, these codes are rarely discussed in the literature. All the codes surveyed consider discrete symbols as well as discrete synchronization errors. Of course, in some practical scenarios, discrete symbols are ill suited for transmission or storage of information, and the synchronization errors can be small fractions of the length or duration of a symbol. Nevertheless, symbol synchronization errors are easy to model and capture the difficulty of manipulating information on such channels.

A. Mathematical Preliminaries

Let $\Sigma_q \triangleq \{0, 1, \ldots, q-1\}$ be an alphabet with $q \geq 2$ symbols, and let Σ_q^n be the set of all strings of length n over Σ_q. We write $x \triangleq x_1 x_2 \ldots x_n$ to denote a q-ary string of length n in Σ_q^n. A *run* is a substring of identical symbols of maximum length. For example, the string $u = 01100$ has three runs.

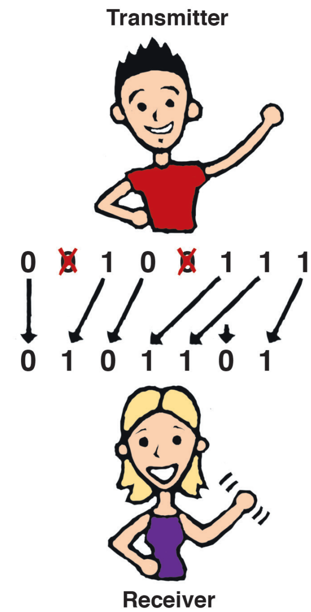

Fig. 1. A message corrupted by synchronization errors.

B. Types of Synchronization Errors

A *synchronization error* is either a *deletion error* or an *insertion error* and excludes substitution errors. In Figure 1, the message 00100111 is transmitted over a channel with synchronization errors. The second and fifth transmitted bits are deleted and a bit is inserted after the seventh transmitted bit, resulting in the received message 0101101. A *duplication* or *repetition error* is a special insertion that replaces a bit by two copies of the same bit. Deletions and duplications arise naturally in systems where errors are caused by the varying rate of imperfect sampling devices. This is shown in Figure 2, where the transmitted message 00100111 is sampled with four duplication errors and a deletion error at the receiver, resulting in the received message 00011100011. A *bitshift error*, also called *peak-shift error*, transforms the substring 01 into the substring 10, or vice-versa. A *bitshift error of size* α transforms the substring $0^\alpha 1$ into the substring 10^α, or vice-versa, and also corresponds to the deletion of α zeroes on one side of a 1 followed by the insertion of α zeroes on the other side. Bitshift errors are of interest because they can be caused by intersymbol interference in magnetic recording systems where information is represented using polarity changes along the data track.

C. Obstacles Specific to Synchronization Error-Correcting Codes

Besides the usual challenges facing code designers like constructing codes with good distance properties and manageable decoding algorithms, synchronization errors introduce difficulties that do not occur for other classes of errors. One of them is that a single uncorrected synchronization error can have catastrophic consequences by causing a huge burst of substitution errors lasting until the system is resynchronized. Consider the example shown in Figure 3. Using the mapping

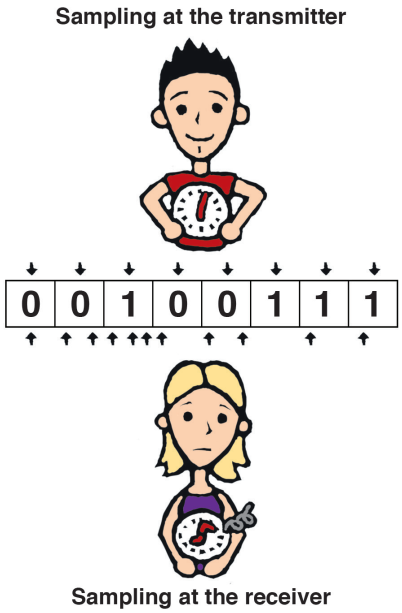

Fig. 2. Deletion and duplication errors caused by the varying rate of sampling devices.

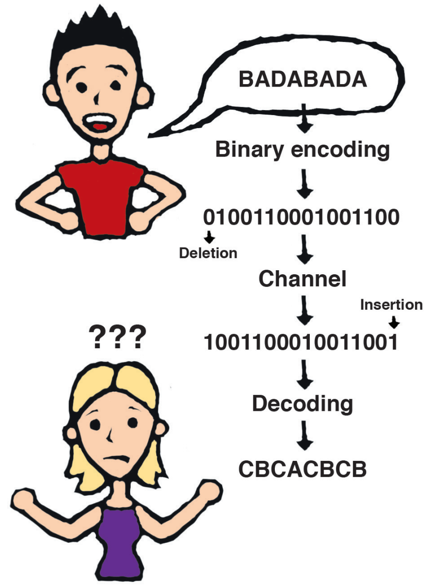

Fig. 3. Huge burst of substitution errors caused by an uncorrected synchronization error.

$\{A \leftrightarrow 00, B \leftrightarrow 01, C \leftrightarrow 10, D \leftrightarrow 11\}$, the word BADABADA is encoded into the binary string 0100110001001100 and transmitted over a channel with synchronization errors. If the first bit is deleted and left uncorrected, then the received binary message corresponds to the word CBCACBCB.

In most communication systems, long messages are divided into blocks or words, thus another challenge introduced by synchronization errors is that the boundaries of the blocks might be unknown to the receiver. It is therefore required to deal with symbol and word synchronization in order to design coding schemes for such systems. Furthermore, a code that can correct a certain number of synchronization errors in one block does not necessarily keep the same correction capability when used to encode a long message using several blocks, because received sequences cannot always be parsed. Consider, for example, the code consisting of the two codewords 000 and 111. This code can clearly correct two deletion errors if a single block is transmitted, but not two deletions per block if a long message is divided into blocks of length 3. For instance, the received string 000000 could have been produced, among others, by the uncorrupted transmission of 000000 and by the transmission of 000000000000 with two deletion errors per block.

III. Synchronization-Correcting Codes

A. Binary Algebraic Block Codes

The Varshamov-Tenengolts codes [17] consist of all the binary vectors of length n satisfying

$$\sum_{i=1}^{n} i \cdot x_i \equiv a \pmod{m} \tag{1}$$

for $m = n + 1$ and a fixed a such that $0 \leq a \leq m - 1$. Binary algebraic synchronization error-correcting codes were first studied by Levenshtein [18], who realized that Varshamov and Tenengolts' codes, originally constructed to correct one asymmetric error, were also asymptotically optimal single-synchronization-correcting codes. Levenshtein proved that the number of codewords satisfying (1) is asymptotically equal to $\frac{2^n}{n}$ and provided a simple decoding algorithm to correct one synchronization error. He also gave bounds on the size of codebooks able to correct more than one synchronization error and explained how to maintain synchronization during long transmissions by inserting separators between blocks. Varshamov [19] proved that the number of vectors satisfying (1) is maximized when $a = 0$, henceforth we will consider the case $a = 0$ when referring to Levenshtein single-synchronization codes. Ullman [20] independently presented a slightly different and more redundant family of codes also capable of correcting one synchronization error.

As we show in this survey, Levenshtein single-synchronization-correcting codes were subsequently modified and adapted in numerous coding schemes. In fact, systems of congruences similar to the one introduced in (1) are behind a large fraction of the work on algebraic synchronization-correcting codes. For instance, Levenshtein [18] showed that the codes consisting of the codewords of length n satisfying (1) with $m = 2n$ could correct one synchronization error or one substitution error. Saowapa, Kaneko, and Fujiwara [21] showed that single-synchronization-correcting codes in systematic form could be constructed using a subset of Levenshtein codes. The number of codewords of length n in their codebooks is exactly $\frac{2^{n-1}}{n}$ when n is a power of 2, which is asymptotically equal to half the number of codewords in Levenshtein codebooks. Tenengolts [22] constructed codes that can correct a substitution error immediately followed by a deletion error, again using a similar family of congruences.

Unfortunately it does not seem that Varshamov-Tenengolts' construction for a single error can be generalized to correct multiple synchronization errors in an asymptotically optimal way. For instance, Helberg and Ferreira [23] extended it for up to five insertion and deletion errors, but the loss in rate is significant, they do not provide efficient encoding and decoding algorithms, and their technique does not scale well for large block sizes. Double-synchronization-correcting codebooks found using a greedy algorithm were studied by Swart and Ferreira [24], who improved upon the results from [23].

Sloane [14] observed that interesting questions remain unanswered when a single bit is deleted from a codeword, like the difference between the size of the largest codebooks and the size of the largest linear codebooks able to correct one deletion error. It should also be pointed out that the size of the spheres associated with deletion-correcting-codewords is not constant. Although the number of subsequences that can be obtained by deleting one symbol from a string is equal to the number of runs of the said string, no such simple expressions exist when two or more bits are deleted from a string [25]. This partly explains why it seems very challenging to construct algebraic codes for more than one synchronization error. Schulman and Zuckerman [26] showed the existence of polynomial-time encodable and decodable codes which are asymptotically good for channels allowing synchronization and transposition errors, but their constructions are not described explicitly and seem to be only of theoretical interest. They use a concatenated scheme consisting of a Reed-Solomon outer code and an inner code found by a greedy algorithm.

B. Nonbinary and Perfect Codes

Nonbinary synchronization-correcting codes were first studied by Calabi and Hartnett [27] and Tanaka and Kasai [28]. Tenengolts [29] generalized Levenshtein single-synchronization-correcting codes for nonbinary alphabets. Levenshtein [30] derived bounds for the maximum size of nonbinary codes capable of correcting s synchronization errors. He also developed asymptotic bounds for the maximum size of single-deletion-correcting codebooks in Σ_q^n as $\frac{q}{n}$ approaches infinity [31].

In another seminal paper, Levenshtein [31] defined and studied perfect deletion-correcting-codes. A code $C \subseteq \Sigma_q^n$ capable of correcting s deletion errors is perfect if the disjoint spheres associated to its codewords partition Σ_q^{n-s}. Levenshtein proved that all the binary single-synchronization-correcting codes whose codewords satisfy (1) are perfect deletion-correcting codes. Using ordered Steiner systems [32], he also constructed perfect deletion-correcting codes of length three and any alphabet size, as well as perfect codes of length four and even alphabet size. Several authors [33]-[42] followed and studied perfect nonbinary deletion-correcting-codes using combinatorial designs. Besides the Varshamov-Tenengolts codes, all the constructions are derived using ad hoc techniques and without encoding or decoding algorithms.

Hollmann [43] and Konstantinidis [44] studied the relationships between distance functions and the correcting capabilities of codes. Konstantinidis [45], [46] also presented an algorithm that tests whether a code is error-correcting for a large class of channels corrupted by synchronization and substitution errors.

Mitzenmacher [47] developed capacity-approaching low-density parity-check codes for deletion channels with q-ary alphabets. For sufficiently large n and q, he constructed codes of length n with rate $(1-p_d)(1-\epsilon)$ that can achieve a probability of failure in $\frac{n^{O(1/\epsilon^2)}}{q}$ on a channel deleting each symbol with probability p_d. The work is mainly of theoretical interest due to the fact that the computational complexity of the decoding algorithm is in $n^{O(1/\epsilon^2)} \log q$. Inspired by Mitzenmacher's approach, Metzner [48] proposed to encode k information packets of size $r \gg 2n-k$ into a codeword of n packets using an (n,k) linear code with minimum distance d. The verification-based decoding algorithm requires $O(n^3)$ bit operations and can correct $d-3$ packet deletions, even if the packets are not received in order. Corrupted packets can also be discarded with high probability.

Tonien and Safavi-Naini [49] and McAven and Safavi-Naini [50] showed that generalized Reed-Solomon codes can be used to correct deletions and decoded in polynomial time using the list decoding algorithm of Guruswami and Sudan [51]. For instance, they found a 6619-ary code of length 18 that can correct fifteen symbol deletions. The motivation behind their work was to use the codes to trace fingerprinted media corrupted by colluders cropping their pirate copies [52].

As opposed to nonbinary codes like Reed-Solomon codes, which can be used to correct bursts of binary substitution errors, a single bit deletion or insertion in a codeword from a large alphabet is sufficient to corrupt all the transmitted symbols. Consequently, synchronization-correcting codes with large alphabets can only work over channels that delete or insert entire symbols. Furthermore, when the alphabet size is very large, only a small overhead is required to transform a channel that deletes and transposes symbols into a symbol erasure channel. The obvious example here is the transmission over packet-switched networks, where short preambles allow to sort the packets and to know which ones have been deleted by the channel. Capacity-approaching rateless erasure codes [53] with extremely small encoding and decoding complexities can also be used to deal with corrupted, deleted, duplicated, and unsorted packets. Unless improved coding schemes are discovered, it is thus unlikely that synchronization-correcting codes with very large alphabets will become a viable alternative to erasure codes and other techniques used in packet-switched networks in the foreseeable future.

C. Bursts of Synchronization Errors

A few codes were specifically designed to correct bursts of synchronization errors. The first such construction is from Levenshtein [54], who, using a family of congruences similar to the one introduced in (1), constructed asymptotically optimal codes capable of correcting two consecutive deletion errors.

Iizuka, Kasahara and Namekawa [55] designed codes that can correct a burst of substitution errors as well as a burst of insertion or deletion errors occurring in the burst of substitution errors. Their codes consist of a coset code from a cyclic code interleaved with a single burst code. The decoding is done by trying all the possible positions of synchronization bursts of increasing size until it is possible to deinterleave a synchronized codeword and correct the burst of substitution errors. One of their codes has blocks of length 1245, a rate of 0.33 and can correct, in each block, a burst of 8 deletion or insertion errors occurring in a burst of substitution errors of size 83. Iwamura and Imai [56] constructed a code that splits information sequences in k segments of q bits, and can correct one synchronization error and a burst of substitution errors as long as all the errors are located in the same segment. The number of required parity bits is $6q + 7\log k + O(\log q)$.

Bours [57] constructed codes that can correct small infrequent bursts of deletion or insertion errors. His construction is a 2-dimensional array code, where the rows of the array are codewords of a comma-free code used to recover synchronization, and the columns of the array are codewords from the (32,16,8) Reed-Muller code used to correct substitution errors as well as erasures caused by the temporary loss of synchronization.

D. Synchronizable Codes

Several researchers have designed codes to protect communication systems against loss of synchronization, like comma-free codes [58], [59], prefix-synchronized codes [60], codes with bounded synchronization delay [61], and synchronization with timing codes [62]. These synchronizable codes are designed to recover block boundaries efficiently and sometimes to correct a few substitution errors or to estimate the time indexes of the decoded symbols, but not to correct synchronization errors per se. Blocks corrupted by synchronization errors are usually discarded. There is a large body of literature on synchronizable codes and their applications, which is beyond the scope of the present survey. An overview of the early work on the subject was written by Stiffler [63], and more recent results can be found in [64]-[67] and references therein.

Morita, van Wijngaarden, and Vinck [68] used Levenshtein single-synchronization-correcting codes to construct prefix synchronized codes capable to correct one synchronization error per codeword. Calabi and Hartnett [69] presented a construction very similar to comma-free codes using carefully selected subsets of Levenshtein codes so they can be used in a concatenated fashion. The resulting codes can either correct one synchronization error or $t - 2$ substitution errors every t blocks. No decoding algorithm is provided, but explicit codes for various block lengths, as cited in [69], are included in an unpublished report [70]. Calabi and Hartnett's codes were extended by Tanaka and Kasai [28] to correct e synchronization errors every $t \geq 2e + 1$ blocks. Liu and Mitzenmacher [71] also used single-deletion-correcting codes in a concatenated fashion so they can correct one deletion error per block. The rate of their codes is lower than the rate of Calabi and Hartnett's codes, although they provided a simple linear time decoding algorithm.

E. Marker Codes

Another technique to detect and correct synchronization errors is to insert periodic markers in the codewords. Marker codes were first introduced by Sellers [72], who suggested to insert the substring 001 at regular intervals in a codeword from a burst-correcting code. The markers can be used to find the interval where a single synchronization error has occurred, after which a bit is deleted or inserted in the middle of the interval and the resulting burst of substitution errors is corrected by the outer decoder. Sellers also discussed use of longer markers to correct bursts of synchronization errors.

The separators used by Levenshtein [18], [73] in two of his constructions play the same role as markers. Ferreira et al. [74] used markers based on Levenshtein single-synchronization-correcting codes that can recover synchronization if there is at most one substitution or synchronization error per block, where a block contains the bits between consecutive markers plus a marker. Their algorithm can correct synchronization and substitution errors if there is at most one of them per two blocks.

Sethakaset and Gulliver [75] used markers to correct insertion and deletion errors for communications over wireless infrared channels, and obtained improved results in [76] using an inner marker code and an outer Reed-Solomon code. It should be noted here that the synchronization errors are not caused by timing uncertainties of the channel. Instead, the authors use differential pulse-position modulation [8], for which the symbols do not have the same duration, thus additive noise results in the insertion and the deletion of nonbinary symbols.

Additional codes using markers and probabilistic synchronization are discussed in Subsection III-J.

F. Codes for Weak Synchronization Errors

In this section, we survey codes able to correct weak synchronization errors, weak in the sense that the synchronization between the transmitter and receiver is partly preserved. Weak synchronization errors include duplication errors, bit shifts, and binary channels where only 0s (or only 1s) can be deleted or inserted. For all these errors, the channel can only change the length of the runs of the transmitted messages. Runs cannot be completely deleted, nor can new runs be created. This is clear for duplications errors. For bit shifts and binary channels where only 0s can be deleted or inserted, the 1s act as markers between the zeroes, and the decoding problem is to recover the original number of zeroes between each 1.

The first study of codes robust against weak synchronization errors was again done by Levenshtein [73]. Using congruences similar to the one used in (1), he constructed asymptotically optimal binary codes capable of correcting the insertion or deletion of a single one. His construction can be extended to build codes robust against s insertions and deletions of ones with asymptotically minimal redundancy. He also showed that a separator of at least $2s+1$ bits must be inserted between consecutive blocks, and that the separator 0^s1^{s+1} could be used. Dolecek and Anantharam [77] presented binary codes able to correct multiple duplication errors, which are equivalent to codes capable of correcting multiple insertions of ones. Their construction is asymptotically better than Levenshtein's by a constant factor, although it does not correct deletions. Dolecek and Anantharam [78] also described a prefixing method for correcting multiple duplication errors in any binary code. The size of the prefixes grow logarithmically with the size of the codewords.

A (d, k)-constrained runlength-limited code is a code such that there are at least d and at most k zeroes between consecutive ones. Constrained codes are commonly used to reduce intersymbol interference in magnetic recording systems by increasing the distance between consecutive 1s. Some classes of constrained runlength-limited codes have been found to be robust against weak synchronization errors. Kuznetsov and Vinck [79] constructed a concatenated constrained code able to correct a single bitshift of size up to $\frac{k-d}{2}$ or the deletion of $\frac{k-d}{2}$ zeroes between consecutive ones or the insertion of $\frac{k-d}{2}$ zeroes between consecutive ones. The outer code consists of a $(k-d+1)$-ary block code, and the inner code is simply a mapping of the nonbinary symbols into binary codewords of the type $0^{\alpha}1$, for $d \leq \alpha \leq k$.

Similarly to Levenshtein's work on perfect deletion-correcting codes, Levenshtein and Vinck [80] introduced perfect constrained bitshift-correcting codes. They presented perfect binary codes that can correct a bitshift error of size 1, 2, and $\frac{(p-1)}{2}$ where p is a prime number. Kløve [81] proved the existence and provided constructions for perfect constrained codes with various error metrics. For instance, one of his perfect codes can correct one bitshift error per block, and another one can correct a single bitshift error or the insertion of a zero or a deletion of a zero.

Roth and Siegel [82] designed constrained BCH codes based on the Lee metric [83] able to correct bitshift errors as well as insertions and deletions of zeroes. For blocks of length $n = p^m - 1$, the redundancy of their code is at most $1 + (r-1)\log_p(n+1)$, and it can simultaneously correct b bitshift errors and s insertions and deletions of zeroes as long as $2b+s < r$. The codes are decoded using Euclid's algorithm. Constrained codes based on the Lee metric with similar error-correction capabilities and higher rates were independently obtained by Bours [84]. His codes are constructed by mapping codewords from a q-ary code based on the Lee metric into (d, k) constrained codewords.

G. Convolutional Codes

Another approach is to adapt convolutional codes for synchronization error-correction. This was first suggested by Gallager [85], whose work is discussed in Subsection III-J. A technique proposed to facilitate block synchronization of convolutional codes is to invert alternate symbols at the output of the encoder to avoid long blocks of zeroes and ones, based on the assumption that such long runs are more frequent than long sequences of alternate bits. Baumert, McEliece, and van Tilborg [86] studied which convolutional encoders can output infinitely many consecutive alternate bits, and for the other codes provided upper bounds on the maximum length of such sequences.

Bouloutas, Hart and Schwartz [87] generalized the Viterbi decoding algorithm to correct synchronization and substitution errors from data generated by finite state machines. They mentioned that their algorithm could be used with convolutional codes, although they did not present any results. Hart and Bouloutas [88] extended the work to correct more general errors modeled using channel rules. Their approach cannot be used efficiently for channels allowing a large number of insertion errors between consecutive message symbols.

Mori and Imai [89] used convolutional encoders and a metric based on the Levenshtein distance for the Viterbi decoding algorithm. Brink, Ferreira, and Clarke [90] used convolutional codes from which they pruned branches to obtain a subset of Levenshtein single-synchronization-correcting codes. They presented a code of rate $\frac{1}{4}$ that can correct one synchronization error. Further results on pruned convolutional codes for deletion and substitution errors were obtained by Cheng and Ferreira [91]. Swart and Ferreira [92] modified convolutional encoders to correct a small number of insertions or deletions of bits by inverting some of the output bits to eliminate large runs of 0s and 1s. They designed codes of rate $\frac{1}{4}$ and $\frac{1}{3}$ that can correct one deletion error every 8 and 12 bits, respectively. Dos Santos et al. [93] used convolutional codes and parallel Viterbi decoders to correct a small number of deletions of bits. One of their codes of rate $\frac{1}{3}$ has a bit error rate of approximately 10^{-2} on a channel that randomly deletes each transmitted bit with probability 10^{-3}. Cheng, Ferreira and Swart [94] used a metric based on the Levenshtein distance to decode convolutional codes for the deletion channel. One of their codes has a rate of $\frac{1}{4}$ and a bit error rate between 10^{-4} and 10^{-5} on a channel that randomly deletes 1% of the bits.

H. Spectral Properties of Single-Synchronization-Correcting Codes

The design of codes with low power near the zero frequency was considered for data storage devices as well as communication over metallic cables [95]. A codeword $x = x_1 x_2 \ldots x_n$ where $x_i \in \{-1, +1\}$ is of K-th-order zero-disparity if

$$\sum_{i=1}^{n} i^k x_i = 0 \text{ for } 0 \leq k \leq K. \quad (2)$$

Immink and Beenker [95] studied codewords of K-th-order zero-disparity and proved that the first $2K+1$ derivatives of their power spectral density function are zero at zero frequency. Noting the similarities between (1) and (2), Ferreira et al. [74] pointed out that the codes of Immink and Beenker are subsets of constant-weight Levenshtein codes, thus they can correct one synchronization error. They also studied synchronization-correcting codes with nulls at other frequencies, as well as the inclusion of markers between codewords such that the resulting codes can correct one synchronization error per block while preserving their good spectral properties.

Ouahada et al. [96] showed that a family of permutation codes [97] of block length M^2 were subsets of Levenshtein codes and thus could correct one synchronization error. The codes have spectral nulls at the frequency $\frac{\alpha}{M}$ for $\alpha \in \{0, 1, \ldots, M\}$. Helberg, Clarke, and Ferreira [98] constructed a code capable to correct one synchronization error as well as a bitshift. The code has the additional property of having a first order null at the zero frequency, and is again a subset of Levenshtein single-synchronization-correcting code.

I. Expurgated Codes

It has recently been proposed to correct synchronization errors by expurgating existing codes like Reed-Muller and array-based LDPC codes. Costing only a small reduction in rate, the expurgation allows to keep the robustness of the original codes against substitution errors and, to a certain extent, to use their decoding algorithms efficiently in the presence of synchronization errors.

Dolecek and Anantharam [99] pruned Reed-Muller $\mathrm{RM}(1,m)$ codes of length n to obtain codes that can correct one deletion or one duplication error per block while remaining robust against substitution errors. Their decoding algorithm is based on the fast Hadamard transform decoder for Reed-Muller codes, and its complexity is in $O(n \log n)$. The expurgation reduces the number of information bits by one. By further pruning $\mathrm{RM}(1,m)$ codes, Dolecek [100] constructed codes that can correct one deletion error and one duplication error, although no decoding algorithm is given. Dolecek and Anantharam [101] expurgated array-based LDPC codes so they can also correct one duplication error per block. The rate loss is asymptotically equal to zero as the code length approaches infinity. It should be mentioned that all the schemes in this subsection assume that the boundaries between the codewords are known.

J. *Codes for Random Synchronization Channels*

All the synchronization-correcting codes presented so far have the drawback of only working under restrictive noise assumptions, like one synchronization error every k bits, the insertion and deletion of zeroes or the corruption of large symbols. On a channel which, say, randomly deletes 1% of the bits, they will either suffer from rates far below capacity or fail to maintain synchronization, resulting in catastrophic bursts of substitution errors. The codes that follow were designed for more adversarial noise assumptions.

In 1961, Gallager [85] studied a binary synchronization channel where each input bit is either deleted, complemented, or replaced by two random bits. He suggested to add pseudo-random sequences to the output of convolutional encoders and to correct synchronization errors using a sequential decoding algorithm. He also derived a computational cut-off rate above which the expected computation required by the decoding algorithm becomes unbounded. One should note that Gallager's work precedes the Viterbi decoding algorithm [102], and that like his early work on low-density parity-check codes, it has somehow been overlooked for a long time. Bahl and Jelinek [103] studied very general channels corrupting input sequences into variable-length output sequences. They also designed a stack decoding algorithm for these channels.

Chen et al. [104] constructed concatenated codes for deletion channels. Their inner codes, used to recover synchronization, combine Levenshtein single-synchronization-correcting codes and a marker, whereas the outer codes are low-density parity-check codes. One of their codes has a rate of 0.21 and can achieve a bit error rate of 10^{-4} on a channel that randomly deletes 8% of the bits.

Davey and MacKay [105] designed a concatenated scheme using a nonlinear inner code named watermark code and a low-density parity-check outer code over a nonbinary field. The output of the LDPC encoder is mapped into a sparse binary string, which is then added modulo 2 to a pseudo-random watermark vector known to the transmitter and the receiver. The watermark can be seen as a marker uniformly distributed throughout the codeword and is used to recover synchronization using a probabilistic algorithm. The outer LDPC code is used to correct substitution errors caused by the channel and by the imperfect synchronization provided by the inner encoder. One of the codes they presented has a rate of 0.7 and can correct 30 synchronization errors per block of length 5000. Davey and MacKay's approach can also be used to acquire block boundaries, thus synchronization can be maintained during long transmissions.

Ratzer and MacKay [106] replaced the watermark inner code by fixed and pseudo-random markers of different size also decoded with a probabilistic algorithm. The best markers performed better than random watermarks at very low error rates. Ratzer [107] implemented a complete iterative system for marker codes, where the soft output of the inner decoder is transmitted to the LDPC outer decoder. The output of LDPC decoder can then be used by the inner decoder to improve the synchronization. One of his iterative codes has a rate of 0.5, blocks of 4000 bits, and can achieve a block error rate of 10^{-3} on a channel that randomly deletes 7% of the bits.

IV. Future Directions

Synchronization error-correcting codes have a long and rich history, and have been studied in various forms both for their intrinsic appeal and their potential applications. This being said, the number of fundamental problems remaining unsolved is quite remarkable.

A. *Binary Block Codes for Multiple Synchronization Errors*

Most of the good binary block codes capable of correcting one synchronization error are related to the original construction of Varshamov and Tenengolts [17], and the gap from one to two synchronization errors seems very hard to bridge. There is not a single known algebraic binary block code with a reasonably good rate and reasonably simple encoding and decoding algorithms that can correct more than one synchronization error per block. The maximum size of binary codes capable of correcting two synchronization errors is also unknown for very small block lengths [108].

B. *Codes for Synchronization Channels*

Estimating the capacity of channels corrupted by synchronization errors is a very challenging problem, and a recent survey on the subject was written by Mitzenmacher [109]. Dobrushin [110] generalized Shannon's channel coding theorem and its weak converse for channels with random synchronization errors, but his work does not lead to useful ways of computing capacity bounds. Tremendous progress has been made for deletion channels [111], [112] and channels with duplication errors [113], for instance, but almost nothing is known about the capacity of channels where random deletion and insertion errors can occur simultaneously or about the capacity of more practical channels with synchronization errors and additive noise. It is not surprising, then, that constructing synchronization-correcting codes achieving channel capacity with vanishing bit error probability remains a somewhat distant objective.

C. From Theory to Practice

If synchronization-correcting codes are ever to be used in practical communication systems, we believe that they should at the very least have the following three properties. Firstly, global loss of synchronization has catastrophic consequences and must be avoided, even if the correction of some of the synchronization errors results in small bursts of substitution errors. Otherwise, it will not be possible to curtail the role of current techniques used to ensure synchronization of communication systems. Secondly, the codes should be able to correct errors caused by additive noise or intersymbol interference at rates close to capacity. Finally, the additional complexity required to correct synchronization errors should not be prohibitive. Unfortunately, none of the codes surveyed in this article satisfies all three properties. This being said, it seems to us that concatenated coding schemes where an inner code is used to recover synchronization and an outer code is used to correct additive noise and imperfect synchronization from the inner decoder are the most promising avenue. To conclude, although it is not clear where and how exactly synchronization-correcting codes could be used, we believe that once good codes are discovered, systems designers will find ways interesting ways to use them.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their comments and suggestions, which greatly improved the presentation of this survey, and Amélie Truchon for her help with the figures.

REFERENCES

[1] K. A. S. Immink, *Codes for Mass Data Storage Systems*, 2nd ed. Shannon Foundation Publishers, 2004.
[2] D. J. Kinniment, *Synchronization and Arbitration in Digital Systems*. Wiley, 2008.
[3] B. Sklar, *Digital Communications: Fundamentals and Applications*, 2nd ed. Prentice Hall Communications Engineering and Emerging Technologies Series, 2001.
[4] S. W. Golomb, J. R. Davey, I. S. Reed, H. L. V. Trees, and J. J. Stiffler, "Synchronization," *IEEE Trans. Commun. Syst.*, vol. 11, no. 4, pp. 481–491, 1963.
[5] R. Cole, J. Mariani, H. Uszkoreit, G. B. Varile, A. Zaenen, and A. Zampolli, Eds., *Survey of the State of the Art in Human Language Technology*. Cambridge University Press, 1998.
[6] D. Sankoff and J. Kruskal, Eds., *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. CSLI Publications, 1999.
[7] S. N. Diggavi and M. Grossglauser, "On information transmission over a finite buffer channel," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1226–1237, Mar. 2006.
[8] D. Shiu and J. M. Kahn, "Differential pulse-position modulation for power-efficient optical communication," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1201–1210, 1999.
[9] A. Orlitsky, "Interactive communication of balanced distributions and of correlated files," *SIAM J. Discrete Mathematics*, vol. 6, no. 4, pp. 548–564, 1993.
[10] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.
[11] G. Cormode, M. Paterson, S. Sahinalp, and U. Vishkin, "Communication complexity of document exchange," in *Proc. eleventh ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2000, pp. 197–206.
[12] S. Agarwal, D. Starobinski, and A. Trachtenberg, "On the scalability of data synchronization protocols for PDAs and mobile devices," *IEEE Netw.*, July/Aug. 2002.
[13] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
[14] N. J. A. Sloane, "On single-deletion-correcting codes," in *Codes and Designs (Ray-Chaudhuri Festschrift)*, ser. Ohio State University Mathematical Research Institute Publications, K. T. Aréasu and A. Seress, Eds., vol. 10. de Gruyter, 2002, pp. 273–291.
[15] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, pp. 379–423 and 623–656, July and October 1948.
[16] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Paerson Prentice Hall, 2004.
[17] R. R. Varshamov and G. M. Tenengolts, "Codes which correct single asymmetric errors," *Automation and Remote Control*, vol. 26, no. 2, pp. 286–290, 1965, translated from *Avtomatika i Telemekhanika*, 26(2):288-292, 1965 (in Russian).
[18] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics-Doklady*, vol. 10, no. 8, pp. 707–710, Feb. 1966, translated from *Doklady Akademii Nauk SSSR*, 163(4):845-848, 1965 (in Russian).
[19] R. R. Varshamov, "An arithmetic function applicable to coding theory," *Soviet Physics-Doklady*, vol. 10, pp. 185–187, 1965, translated from *Doklady Akademii Nauk SSSR*, 161(3):540-543, 1965 (in Russian).
[20] J. D. Ullman, "Near-optimal, single-synchronization-error-correcting code," *IEEE Trans. Inf. Theory*, vol. 12, no. 4, pp. 418–424, 1966.
[21] K. Saowapa, H. Kaneko, and E. Fujiwara, "Systematic binary deletion/insertion error-correcting codes capable of correcting random bit errors," *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, vol. E83-A, no. 12, pp. 2699–2705, 2000.
[22] G. M. Tenengolts, "Class of codes correcting bit loss and errors in the preceding bit," *Automation and Remote Control*, vol. 37, pp. 797–802, 1976, translated from *Avtomatika i Telemekhanika*, 37(5):174-179, 1976 (in Russian).
[23] A. S. J. Helberg and H. C. Ferreira, "On multiple insertion/deletion correcting codes," *IEEE Trans. Inf. Theory*, vol. 48, no. 1, pp. 305–308, 2002.
[24] T. G. Swart and H. C. Ferreira, "A note on double insertion/deletion correcting codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 1, pp. 269–273, Jan. 2003.
[25] H. Mercier, M. Khabbazian, and V. K. Bhargava, "On the number of subsequences when deleting symbols from a string," *IEEE Trans. Inf. Theory*, vol. 54, no. 7, pp. 3279–3285, 2008.
[26] L. J. Schulman and D. Zuckerman, "Asymptotically good codes correcting insertions, deletions, and transpositions," *IEEE Trans. Inf. Theory*, vol. 45, no. 7, pp. 2552–2557, 1999.
[27] L. Calabi and W. E. Hartnett, "Some general results of coding theory with applications to the study of codes for the correction of synchronization errors," *Information and Control*, vol. 15, no. 3, pp. 235–249, Sep. 1969.
[28] E. Tanaka and T. Kasai, "Synchronization and substitution error-correcting codes for the Levenshtein metric," *IEEE Trans. Inf. Theory*, vol. 22, no. 2, pp. 156–162, Mar. 1976.
[29] G. Tenengolts, "Nonbinary codes, correcting single deletion or insertion," *IEEE Trans. Inf. Theory*, vol. 30, no. 5, pp. 766–769, 1984.
[30] V. I. Levenshtein, "Bounds for deletion/insertion correcting codes," in *Proc. 2002 IEEE International Symposium on Information Theory (ISIT)*, 2002, p. 370.
[31] ——, "On perfect codes in deletion and insertion metric," *Discrete Mathematics and Applications*, vol. 2, no. 3, pp. 241–258, 1992, translated from *Diskretnaya Matematika*, 3(1):3-20, 1991 (in Russian).
[32] C. J. Colbourn and J. H. Dinitz, Eds., *Handbook of Combinatorial Designs*, 2nd ed. Chapman & Hall/CRC, 2006.
[33] P. A. H. Bours, "On the construction of perfect deletion-correcting codes using design theory," *Designs, Codes and Cryptography*, vol. 6, pp. 5–20, 1995.
[34] A. Mahmoodi, "Existence of perfect 3-deletion-correcting codes," *Designs, Codes and Cryptography*, vol. 14, pp. 81–87, 1998.
[35] R. Mathon and T. V. Trung, "Directed t-packings and directed t-Steiner systems," *Designs, Codes and Cryptography*, vol. 18, pp. 187–198, 1999.
[36] R. Fuji-Hara, Y. Miao, J. Wang, and J. Yin, "Directed $B(K, 1; v)$ with $K = \{4, 5\}$ and $\{4, 6\}$ related to deletion/insertion-correcting codes," *J. Combinatorial Designs*, vol. 9, no. 2, pp. 147–156, 2001.
[37] J. Yin, "A combinatorial construction for perfect deletion-correcting codes," *Designs, Codes and Cryptography*, vol. 23, pp. 99–110, 2001.
[38] N. Shalaby, J. Wang, and J. Yin, "Existence of perfect 4-deletion-correcting codes with length six," *Designs, Codes and Cryptography*, vol. 27, pp. 145–156, 2002.
[39] A. Klein, "On perfect deletion-correcting codes," *J. Combinatorial Designs*, vol. 12, no. 1, pp. 72–77, 2004.

[40] J. Wang and L. Ji, "Existence of $T^*(3,4,v)$-codes," *J. Combinatorial Designs*, vol. 13, no. 1, pp. 42–53, 2005.
[41] J. Wang and J. Yin, "Constructions for perfect 5-deletion-correcting codes of length 7," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3676–3685, 2006.
[42] J. Wang, "Some combinatorial constructions for optimal perfect deletion-correcting codes," *Designs, Codes and Cryptography*, vol. 48, no. 3, pp. 331–347, 2008.
[43] H. D. L. Hollman, "A relation between levenshtein-type distances and insertion-and-deletion correcting capabilities of codes," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1424–1427, 1993.
[44] S. Konstantinidis, "Relationships between different error-correcting capabilities of a code," *IEEE Trans. Inf. Theory*, vol. 47, no. 5, pp. 2065–2069, 2001.
[45] ——, "Structural analysis of error-correcting codes for discrete channels that involve combinations of three basic error types," *IEEE Trans. Inf. Theory*, vol. 45, no. 1, pp. 60–77, 1999.
[46] ——, "An algebra of discrete channels that involve combinations of three basic error types," *Information and Computation*, vol. 67, pp. 120–131, 2001.
[47] M. Mitzenmacher, "Polynomial time low-density parity-check codes with rates very close to the capacity of the q-ary random deletion channel for large q," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5496–5501, 2006.
[48] J. J. Metzner, "Packet symbol decoding for reliable multipath reception with no sequence numbers," in *Proc. 2007 IEEE International Conference on Communications (ICC)*, 2007, pp. 809–814.
[49] D. Tonien and R. Safavi-Naini, "Construction of deletion correcting codes using generalized Reed-Solomon codes and their subcodes," *Designs, Codes and Cryptography*, vol. 42, pp. 227–237, 2007.
[50] L. McAven and R. Safavi-Naini, "Classification of the deletion-correcting capabilities of Reed-Solomon codes of dimension 2 over prime fields," *IEEE Trans. Inf. Theory*, vol. 53, no. 6, pp. 2280–2294, June 2007.
[51] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometry codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 1757–1767, 2007.
[52] R. Safavi-Naini and Y. Wang, "Traitor tracing for shortened and corrupted fingerprint," in *Proc. 2002 ACM Workshop on Digital Rights Management*, ser. Lectures Notes in Computer Science, vol. 2696. Springer-Verlag, 2003, pp. 81–100.
[53] D. J. C. MacKay, "Fountain codes," *IEE Proc.-Commun.*, vol. 152, no. 6, pp. 1058–1062, 2005.
[54] V. I. Levenshtein, "Asymptotically optimum binary codes with correction for losses of one or two adjacent bits," *Systems Theory Research*, vol. 19, pp. 298–304, 1970, translated from *Problemy Kibernetiki*, 19:293-298, 1967 (in Russian).
[55] I. Iizuka, M. Kasahara, and T. Namekawa, "Block codes capable of correcting both additive and timing errors," *IEEE Trans. Inf. Theory*, vol. 26, no. 4, pp. 393–400, 1980.
[56] K. Iwamura and H. Imai, "A code to correct synchronization errors," *Electronics and Communications in Japan, Part 3*, vol. 76, no. 6, pp. 60–71, 1993, translated from *Denshi Joho Tsushin Gakkai Ronbunshi*, 75-A(11):1718-1726, 1992 (in Japanese).
[57] P. A. H. Bours, "Codes for correcting insertion and deletion errors," Ph.D. dissertation, Eindhoven University of Technology, June 1994, available at http://alexandria.tue.nl/extra3/proefschrift/PRF10A/9421971.pdf.
[58] F. H. C. Crick, J. S. Griffith, and L. E. Orgel, "Codes without commas," in *Proc. National Academy of Sciences in the United States of America*, vol. 43, no. 5, 1957, pp. 416–421.
[59] S. W. Golomb, B. Gordon, and L. R. Welch, "Comma-free codes," *Canadian Journal of Mathematics*, vol. 10, no. 2, pp. 202–209, 1958.
[60] E. N. Gilbert, "Synchronization of binary messages," *IRE Trans. Information Theory*, vol. 6, no. 4, pp. 470–477, 1960.
[61] S. W. Golomb and B. Gordon, "Codes with bounded synchronization delay," *Information and Control*, vol. 8, pp. 355–372, 1965.
[62] N. Kashyap and D. L. Neuhoff, "Data synchronization with timing," *IEEE Trans. Inf. Theory*, vol. 47, no. 4, pp. 1444–1460, 2001.
[63] J. J. Stiffler, *Theory of Synchronous Communications*. Prentice-Hall, 1971.
[64] L. J. Cummings, "Aspects of synchronizable coding," *Journal of Combinatorial Mathematics and Combinatorial Computing*, vol. 1, pp. 67–84, 1987.
[65] H. Morita, A. J. van Wijngaarden, and A. J. Han Vinck, "On the construction of maximal prefix-synchronized codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 2158–2166, 1996.
[66] S. Konstantinidis, S. Perron, and L. A. Wilcox-O'Hearn, "On a simple method for detecting synchronization errors in coded messages," *IEEE Trans. Inf. Theory*, vol. 49, no. 5, pp. 1355–1363, 2003.
[67] N. Kashyap and D. L. Neuhoff, "Periodic prefix-synchronized codes: A generating function approach," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 538–548, 2006.
[68] H. Morita, A. J. V. Wijngaarden, and A. J. H. Vinck, "Prefix synchronized codes capable of correcting insertion/deletion errors," in *Proc. 1997 IEEE International Symposium on Information Theory (ISIT)*, 1997, p. 409.
[69] L. Calabi and W. E. Hartnett, "A family of codes for the correction of substitution and synchronization errors," *IEEE Trans. Inf. Theory*, vol. 15, no. 1, pp. 102–106, Jan. 1969.
[70] T. Hatcher, "The number of sequences in A_n^c for $0 \leq c < 2n$, $0 \leq n \leq 13$, and listings of selected A_n^c codes," Parke Mathematical Lab., Inc., Carlisle, Massachusetts, 0030-TM-13, 1968.
[71] Z. Liu and M. Mitzenmacher, "Codes for deletion and insertion channels with segmented errors," in *Proc. 2007 IEEE International Symposium on Information Theory (ISIT)*, 2007, pp. 846–850.
[72] F. F. Sellers, "Bit loss and gain correction code," *IRE Trans. Information Theory*, vol. 8, no. 1, pp. 35–38, 1962.
[73] V. I. Levenshtein, "Binary codes capable of correcting spurious insertions and deletions of ones," *Problems of Information Transmission*, vol. 1, no. 1, pp. 8–17, 1965, translated from *Problemy Peredachi Informatsii*, 1(1):12-25, 1965 (in Russian).
[74] H. C. Ferreira, W. A. Clarke, A. S. J. Helberg, K. A. S. Abdel-Gaffar, and A. J. H. Vinck, "Insertion/deletion correction with spectral nulls," *IEEE Trans. Inf. Theory*, vol. 43, no. 2, pp. 722–732, 1997.
[75] U. Sethakaset and T. A. Gulliver, "Marker codes to correct insertion/deletion errors in differential pulse-position modulation for wireless infrared communications," in *Proc. 2005 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, 2005, pp. 181–184.
[76] ——, "Performance of differential pulse-position modulation (DPPM) with concatenated coding over optical wireless communications," *IET Communications*, vol. 2, no. 1, pp. 45–52, 2008.
[77] L. Dolecek and V. Anantharam, "On subsets of binary strings immune to repetition errors," in *Proc. 2007 IEEE International Symposium on Information Theory (ISIT)*, 2007, pp. 1691–1695.
[78] ——, "Prefixing method for correcting repetition errors," in *Proc. 2008 IEEE International Symposium on Information Theory (ISIT)*, 2008, pp. 965–969.
[79] A. V. Kuznetsov and A. J. H. Vinck, "A coding scheme for single peak-shift correction in (d,k)-constrained channels," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1444–1450, 1993.
[80] V. I. Levenshtein and A. J. H. Vinck, "Perfect (d,k)-codes capable of correcting single peak-shifts," *IEEE Trans. Inf. Theory*, vol. 39, no. 2, pp. 656–662, 1993.
[81] T. Kløve, "Codes correcting a single insertion/deletion of a zero or a single peak-shift," *IEEE Trans. Inf. Theory*, vol. 41, no. 1, pp. 279–283, 1995.
[82] R. M. Roth and P. H. Siegel, "Lee-metric BCH codes and their application to constrained and partial-response channels," *IEEE Trans. Inf. Theory*, vol. 40, no. 4, pp. 1083–1096, 1994.
[83] C. Y. Lee, "Some properties of nonbinary error-correcting codes," *IRE Trans. Inf. Theory*, vol. 4, no. 2, pp. 77–82, 1958.
[84] P. A. H. Bours, "Construction of fixed-length insertion/deletion correcting runlength-limited codes," *IEEE Trans. Inf. Theory*, vol. 40, no. 6, pp. 1841–1856, 1994.
[85] R. G. Gallager, "Sequential decoding for binary channels with noise and synchronization errors," Lincoln Lab Group Report 2502, 1961, available at http://web.mit.edu/gallager/www/pages/pubs.html.
[86] L. D. Baumert, R. J. McEliece, and H. C. A. van Tilborg, "Symbol synchronization in convolutionally coded systems," *IEEE Trans. Inf. Theory*, vol. 25, no. 3, pp. 362–365, 1979.
[87] A. Bouloutas, G. W. Hart, and M. Schwartz, "Two extensions of the viterbi algorithm," *IEEE Trans. Inf. Theory*, vol. 37, no. 2, pp. 430–436, 1991.
[88] G. W. Hart and A. T. Bouloutas, "Correcting dependent errors in sequences generated by finite-state processes," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1249–1260, 1993.
[89] T. Mori and H. Imai, "Viterbi decoding considering synchronization errors," *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, vol. E79-A, no. 9, pp. 1324–1329, 1996.
[90] B. Brink, H. C. Ferreira, and W. A. Clarke, "Pruned convolutional codes for flexible unequal error protection against insertion/deletion/reversal errors," in *Proc. 2000 IEEE International Symposium on Information Theory (ISIT)*, 2000, p. 260.

[91] L. Cheng and H. C. Ferreira, "Rate-compatible path-pruned convolutional codes and their applications on channels with insertion, deletion and substitution errors," in *Proc. 2005 IEEE Information Theory Workshop (ITW)*, 2005.
[92] T. G. Swart and H. C. Ferreira, "Insertion/deletion correcting coding schemes based on convolutional coding," *Electronics Letters*, vol. 38, no. 16, pp. 871–873, 2002.
[93] M. P. F. dos Santos, W. A. Clarke, H. C. Ferreira, and T. G. Swart, "Correction of insertions/deletions using standard convolutional codes and the Viterbi decoding algorithm," in *Proc. 2003 IEEE Information Theory Workshop (ITW)*, 2003, pp. 187–189.
[94] L. Cheng, H. C. Ferreira, and T. G. Swart, "Bidirectional Viterbi decoding using the Levenshtein distance metric for deletion channels," in *Proc. 2006 IEEE Information Theory Workshop (ITW)*, 2006, pp. 254–258.
[95] K. A. S. Immink and G. F. M. Beenker, "Binary transmission codes with higher order spectral zeros at zero frequency," *IEEE Trans. Inf. Theory*, vol. 33, no. 3, pp. 452–454, 1987.
[96] K. Ouahada, T. G. Swart, H. C. Ferreira, and L. Cheng, "Binary permutation sequences as subsets of Levenshtein codes, spectral null codes, run-length limited codes and constant weight codes," *Designs, Codes and Cryptography*, vol. 48, pp. 141–154, 2008.
[97] I. F. Blake, G. Cohen, and M. Deza, "Coding with permutations," *Information and Control*, vol. 43, pp. 1–19, 1979.
[98] A. S. J. Helberg, W. A. Clarke, and H. C. Ferreira, "A class of dc free, synchronization error-correcting codes," *IEEE Trans. Magn.*, vol. 29, no. 6, pp. 4048–4049, 1993.
[99] L. Dolecek and V. Anantharam, "Using Reed-Muller RM$(1, m)$ codes over channels with synchronization and substitution errors," *IEEE Trans. Inf. Theory*, vol. 53, no. 4, pp. 1430–1443, 2007.
[100] L. Dolecek, "On structural properties of Reed-Muller codes and their use in channels with synchronization and substitution errors," Electrical Engineering and Computer Sciences, University of California at Berkeley, Tech. Rep. UCB/EECS-2006-43, 2006, available at http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-43.html.
[101] L. Dolecek and V. Anantharam, "A synchronization technique for array-based LDPC codes in channels with varying sampling rate," in *Proc. 2006 IEEE International Symposium on Information Theory (ISIT)*, 2006, pp. 2057–2061.
[102] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, 1967.
[103] L. R. Bahl and F. Jelinek, "Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition," *IEEE Trans. Inf. Theory*, vol. 21, no. 4, pp. 404–411, 1975.
[104] J. Chen, M. Mitzenmacher, C. Ng, and N. Varnica, "Concatenated codes for deletion channels," in *Proc. 2003 IEEE International Symposium on Information Theory (ISIT)*, 2003, p. 218.
[105] M. C. Davey and D. J. C. MacKay, "Reliable communication over channels with insertions, deletions and substitutions," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 687–698, 2001.
[106] E. A. Ratzer and D. J. C. MacKay, "Codes for channels with insertions, deletions and substitutions," in *Proc. 2nd International Symposium on Turbo Codes and Applications*, 2000, pp. 149–156.
[107] E. A. Ratzer, "Marker codes for channels with insertions and deletions," *Annals of telecommunications*, vol. 60, no. 1-2, pp. 29–44, 2005.
[108] N. J. A. Sloane, "Challenge problems: Independent sets in graphs," 2005, published electronically at http://www.research.att.com/~njas/doc/graphs.html.
[109] M. Mitzenmacher, "A survey of results for deletion channels and related synchronization channels," *Probability Surveys*, vol. 6, pp. 1–33, 2009.
[110] R. L. Dobrushin, "Shannon's theorems for channels with synchronization errors," *Problems of Information Transmission*, vol. 3, no. 4, pp. 11–26, 1967, translated from *Problemy Peredachi Informatsii*, 3(4):18-36, 1967 (in Russian).
[111] M. Mitzenmacher and E. Drinea, "A simple lower bound for the capacity of the deletion channel," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4657–4660, Oct. 2006.
[112] E. Drinea and M. Mitzenmacher, "Improved lower bounds for the capacity of i.i.d. deletion and duplication channels," *IEEE Trans. Inf. Theory*, vol. 53, no. 8, pp. 2693–2714, 2007.
[113] M. Mitzenmacher, "Capacity bounds for sticky channels," *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 72–77, 2008.

Hugues Mercier received the B.Sc. degree in mathematics from Université Laval, Québec, Canada, in 1997, the M.Sc. degree in computer science from the Université de Montréal, Montréal, Canada, in 2003, and the Ph.D. degree in electrical and computer engineering from the University of British Columbia, Vancouver, Canada, in 2008. Currently, he is a postdoctoral research fellow at the School of Engineering and Applied Sciences, Harvard University, Cambridge, USA. His research interests include combinatorics, coding theory, and communication complexity.

Vijay K. Bhargava received his B.Sc., M.Sc. and Ph.D. degrees from Queen's University, Kingston, Ontario in 1970, 1972 and 1974 respectively. Vijay has held regular/visiting appointments at the Indian Institute of Science, University of Waterloo, Concordia University, Ecole Polytechnique de Montreal, UNIDO, NTT Wireless Communications Labs, Tokyo Institute of Technology, University of Indonesia, the Hong Kong University of Science and Technology, The Hong Kong University, the City University of Hong Kong and the University of Victoria. Currently he is a Professor in the Department of Electrical and Computer Engineering at the University of British Columbia where he served as Department Head from 2003 to 2008.

Vijay served as the Founder and President of Binary Communications Inc. (1983-2000). He is a co-author (with D. Haccoun, R. Matyas and P. Nuspl) of Digital Communications by Satellite (New York: Wiley 1981), a co-editor (with S. Wicker) of Reed Solomon Codes and their Applications (IEEE Press 1994), a co-editor (with V. Poor, V. Tarokh and S. Yoon) of Communications, Information and Network Security (Kluwer: 2003) and a co-editor (with E. Hossain) of Cognitive Wireless Communications Networks (Springer: 2007). In January 2007, he was appointed Editor-in-Chief of the IEEE Transactions on Wireless Communications and in January 2009 he was renewed for a further two year term.

A Fellow of the IEEE, the Engineering Institute of Canada (EIC), the Royal Society of Canada, and the Canadian Academy of Engineering, Vijay has been honoured many times by his colleagues and has received numerous awards. Vijay is very active in the IEEE and has served as the President of the Information Theory Society. He is a past member of the Board of Governors of the IEEE Communications Society and the IEEE Information Theory Society.

Vahid Tarokh worked at AT&T Labs-Research and AT&T wireless services until August 2000, where he was the head of the Department of Wireless Communications and Signal Processing. In September 2000, he joined the Department of Electrical Engineering and Computer Sciences (EECS) at MIT as an associate professor. In June 2002, he joined Harvard University, where he is a Professor and Senior Fellow of Electrical Engineering. His research is mainly focused in the areas of Signal processing, Communications (wireline and wireless) and Networking. He has received a number of awards and holds 2 honorary degrees.