On the Construction of Maximal Prefix-Synchronized Codes

Hiroyoshi Morita, *Member, IEEE*, Adriaan J. van Wijngaarden, *Student Member, IEEE*, and A. J. Han Vinck, *Senior Member, IEEE*

***Abstract*— We present a systematic procedure for mapping data sequences into codewords of a prefix-synchronized code (PS-code), as well as for performing the inverse mapping. A PS-code, proposed by Gilbert in 1960, belongs to a subclass of comma-free codes and is useful to recover word synchronization when errors have occurred in the stream of codewords. A PS-code is defined as a set of codewords with the property that each codeword has a known sequence as a prefix, followed by a coded data sequence in which this prefix is not allowed to occur. The largest PS-code among all PS-codes of the same code length is called a maximal prefix-synchronized code (MPS-code). We develop an encoding and decoding algorithm for Gilbert's MPS-code with a prefix of the form $11\cdots 10$ and extend the algorithm to the class PS-codes of which the prefix is self-uncorrelated. The computational complexity of the entire mapping process is proportional to the length of the codewords.**

***Index Terms*— Synchronization, frame synchronization, comma-free codes, prefix-synchronized codes, runlength-limited codes, Fibonacci codes, Fibonacci sequences.**

I. Introduction

A BLOCK code C_α^n of length n over an alphabet $\mathcal{A}_\alpha$ of size α is called a *comma-free* code, if and only if for any pair of codewords $a_1a_2\cdots a_n$ and $b_1b_2\cdots b_n$ in C_α^n, the n symbol overlaps

$$a_2a_3\cdots a_nb_1, a_3a_4\cdots a_nb_1b_2, \cdots, a_nb_1b_2\cdots b_{n-1}$$

are not in C_α^n [1]. In a communication system, a comma-free code can be used to enable the receiver to determine the location of the codewords in the incoming stream of symbols. Word synchronization can be recovered after having received at most $2n-2$ error-free symbols.

The cardinality of a comma-free code C_α^n, denoted by $C_\alpha^{(n)}$, is bounded by

$$C_\alpha^{(n)} \le \frac{1}{n}\sum_{d|n}\mu(d)\alpha^{n/d}$$

where μ is the Möbius function [1]. A good approximation for this upper bound is given by $C_\alpha^{(n)} \le \alpha^n/n$, and therefore the redundancy, being equal to $n-\log_\alpha C_\alpha^{(n)}$, is at least $\log_\alpha n$. For any odd n, comma-free codes having maximal size can be constructed [2], [3].

A major disadvantage of general comma-free codes is the need for an exhaustive search in the code set to decide whether or not a given string of n symbols is a codeword. To overcome this difficulty, a subclass of comma-free codes, called *prefix-synchronized* codes (PS-codes), was introduced by Gilbert [4]. These codes have the property that every codeword starts with a prefix $P = p_1p_2\cdots p_k$ of length k, followed by a constrained sequence $c_1c_2\cdots c_m$ of length m. Moreover, for any codeword $p_1\cdots p_kc_1\cdots c_m$, prefix P does not appear as a block of k consecutive symbols anywhere in $p_2\cdots p_kc_1\cdots c_mp_1\cdots p_{k-1}$. Therefore, word synchronization can be easily established at the decoder side by scanning the incoming stream of symbols for the occurrence of prefix P.

Given an alphabet $\mathcal{A}_\alpha$, a code length n, and a prefix P of length $k<n$, there is a unique maximal prefix-synchronized code (MPS-code), denoted by $\mathcal{G}_{P,\alpha}^{(n)}$, with these parameters. Usually, $\mathcal{G}_{P,\alpha}^{(n)}$ is written as $\mathcal{G}_P^{(n)}$ if the value of α can be obtained from the context. Parameter k will be used to represent the length of the prefix. If we stress the prefix length k and the constrained sequence length $m=n-k$, then the notation $\mathcal{G}_P^{(k+m)}$ is used.

We present an encoding and decoding algorithm for a class of MPS-codes with self-uncorrelated prefixes. A sequence X is said to be self-uncorrelated if X has the property that no prefix of X matches any suffix of X. A sequence of the form $P_G = 1\cdots 10$ is an example of self-uncorrelated sequences. It should be noted that a large number of sequences are self-uncorrelated. For example, for k equal to 7, 10, and 15, there exist 40, 284, and 8848 self-uncorrelated binary sequences, respectively. Moreover, it is known that given an alphabet size α with $2\le\alpha\le 4$, self-uncorrelated prefixes, including P_G, maximize the cardinality of $\mathcal{G}_P^{(n)}$ for any given code length n [5].

In the next section, we will give an overview of earlier work on PS-codes. Then, in Section III, we present the recursive structure of the constrained part of $\mathcal{G}_{P_G}^{(k+m)}$ from which a constructive mapping procedure of a data sequence to the constrained part of a codeword can be obtained. The encoding and decoding algorithms for $\mathcal{G}_{P_G}^{(k+m)}$ are presented in Section IV, as well as a proof of the correctness of both algorithms. The time complexity of the proposed coding scheme is proportional to the code length.

Manuscript received August 22, 1995; revised May 9, 1996. This research was supported in part by the Japanese and German science foundations JSPS and DFG. The material in this paper was presented in part at the 1995 IEEE International Symposium on Information Theory, Whistler, BC, Canada, September 1995.

H. Morita is with the Graduate School of Information Systems, University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo 182, Japan.

A. J. van Wijngaarden and A. J. Han Vinck are with the Institute for Experimental Mathematics, University of Essen, 45326 Essen, Germany.

Publisher Item Identifier S 0018-9448(96)06373-0.

In Section V, we address the coding algorithm for the class of PS-codes with self-uncorrelated prefixes. It is known that if P is self-uncorrelated, then $\mathcal{G}_P^{(n)}$ has the same size as $\mathcal{G}_{P_G}^{(n)}$ [5]. However, neither any encoding nor decoding algorithm for such a $\mathcal{G}_P^{(n)}$ have been found in the literature. We give a two-step algorithm for encoding and decoding $\mathcal{G}_P^{(n)}$ for any self-uncorrelated P. The total time complexity is also proportional to the code length.

II. Previous Results

Gilbert [4] has shown that the redundancy of a binary PS-code $\mathcal{G}_P^{(n)}$ is upper-bounded by $\log_2 n + 1.52$, if P is of the form $11\cdots 10$ of length $\lfloor \log_2 (n \log_2 e) \rfloor$. This specific prefix is referred to as Gilbert's prefix, and the corresponding PS-code will be simply denoted by $\mathcal{G}^{(n)}$.

Gilbert conjectured that for a given code length n, $\mathcal{G}^{(n)}$ is optimal in the sense that it is the largest PS-code among all binary PS-codes of length n. For alphabet size $\alpha \leq 4$, the conjecture is proved by Guibas and Odlyzko [5] for sufficiently large n. Surprisingly, it is also proved that for $\alpha \geq 5$ infinitely many values of n exist for which Gilbert's prefix is not optimal [5]. Although it would be interesting to find out which prefix gives an optimal PS-code for a general finite alphabet, we will not consider this open problem in the current paper.

From an engineering point of view, the main practical difficulty of using PS-codes still remains. The encoding and decoding procedures generally become more complex as the length of the codewords increases. In fact, finding a constructive coding method for $\mathcal{G}^{(n)}$ without use of a lookup table has remained as an open problem.

A nearly-optimal construction method has been developed by Mandelbaum [6]. He presents an encoding and decoding procedure for a PS-code based on Fibonacci codes, as proposed by Kautz [7]. This method is constructive in the sense that no lookup table is required. Mandelbaum shows that a binary PS-code, denoted by $\mathcal{M}_{P_G}^{(k+m)}$ can be constructed by applying Kautz's coding method. The redundancy of $\mathcal{M}_{P_G}^{(k+m)}$ is shown to be approximately equal to $(\log_2 n) + 2$, if $n \cong 2^k$. His method was extended to runlength-limited codes [8]–[10]. However, being a PS-code, $\mathcal{M}_{P_G}^{(k+m)}$ is not optimal among all PS-codes of length $k+m$ and prefix P_G of length k. In fact, $\mathcal{G}_{P_G}^{(k+m)}$ is always larger than $\mathcal{M}_{P_G}^{(k+m)}$ for any n which is shown in Section III where the exact difference of size between $\mathcal{M}_{P_G}^{(k+m)}$ and $\mathcal{G}_{P_G}^{(k+m)}$ is determined.

Capocelli [11] proposes another coding scheme for $\mathcal{G}_{P_G}^{(k+m)}$ as a part of unbounded integer coding by showing an example of the scheme for $k = 3$. In fact, for a given k, the infinite union

$$\bigcup_{m=0}^{\infty} \mathcal{G}_{P_G}^{(k+m)}$$

is a code set which can be used to encode arbitrary positive integers. In his method, $\mathcal{G}_{P_G}^{(3+m)}$ is partitioned into two subsets: one set of codewords starting with 0 and the other set of codewords starting with 1. The integer to be encoded is compared with the size of the first subset to obtain the first bit of the corresponding codeword. Continuing these steps recursively, the codeword will be determined bit by bit. The size of the latter set is easily shown to be equal to the second-order Fibonacci number. Therefore, the size of the former set can be also represented using these Fibonacci numbers, although the obtained formula will be complicated. A more formal description of Capocelli's algorithm for any k is found in [12].

Unlike Capocelli's algorithm, we partition $\mathcal{G}_{P_G}^{(k+m)}$ into k subsets, which gives us a much more convenient formula for enumerating the number of codewords. Moreover, derivation of the encoding and decoding algorithms has become straightforward.

III. Maximal Prefix-Synchronized Codes

In this section we investigate a recursive structure of $\mathcal{G}_{P_G}^{(k+m)}$ from which a coding scheme for $\mathcal{G}_{P_G}^{(k+m)}$ is directly deduced. The exact analysis on the difference between $\mathcal{M}_{P_G}^{(k+m)}$ and $\mathcal{G}_{P_G}^{(k+m)}$ is also deduced using the recursive structure. Before developing the theory, we introduce a useful definition of the correlation between two sequences [5], [13] using a slightly different notation.

Definition 1: For two sequences X and Y of length $|X|$ and $|Y|$, respectively, the *correlation* of X over Y, denoted by $X \circ Y$, is a binary sequence $B = b_1 b_2 \cdots b_{|X|}$ of the same length as X. Let $s = \max(|X| - |Y|, 0)$. Each element b_i with $1 \leq i \leq |X|$ is defined by

$$b_i = \begin{cases} \gamma(x_i \cdots x_{|Y|+i-1}, y_1 \cdots y_{|Y|}), & 1 \leq i \leq s \\ \gamma(x_i \cdots x_{|X|}, y_1 \cdots y_{|X|-i+1}), & s < i \leq |X| \end{cases}$$

where $\gamma(Z_1, Z_2)$ is 1 if two sequences Z_1 and Z_2 are identical, and 0 otherwise.

For example, if $X = 1021$ and $Y = 10102$, then $X \circ Y = 0001$ and $Y \circ X = 00100$. Note that in general $X \circ Y \neq Y \circ X$. The correlation $X \circ X$ is called the *autocorrelation* of X. We will denote a sequence of s consecutive symbols $b \in \mathcal{A}_\alpha$ by b^s. Then, for a self-uncorrelated sequence $X, X \circ X = 10\cdots 0$ holds.

Let the concatenation of two sequences X and Y be denoted by XY. In terms of correlation, we can represent the necessary and sufficient condition that a sequence PY of length $n = k+m$ is a codeword of $\mathcal{G}_P^{(k+m)}$ by

$$PYP \circ P = 10^{k+m-1}1(*)^{k-1} \tag{1}$$

where the character $*$ is used to denote an arbitrary symbol of $\mathcal{A}_\alpha$, and $(*)^s$ represents a sequence of $\mathcal{A}_\alpha^s$.

For a prefix P of length $k \geq 1$, let $\mathcal{F}_P^{(m)}$ denote the set of sequences of length m such that no P appears in any position as a string of k consecutive symbols. Therefore, $\mathcal{F}_P^{(m)}$ is defined by

$$\mathcal{F}_P^{(m)} = \begin{cases} \mathcal{A}_\alpha^m, & m < k \\ \mathcal{A}_\alpha^m \backslash \{P\}, & m = k \\ \{X \in \mathcal{A}_\alpha^m | X \circ P = 0^{m-k+1}(*)^{k-1}\}, & m > k. \end{cases} \tag{2}$$

The following lemma can be easily derived from the definition of $\mathcal{F}_P^{(m)}$, and is useful to obtain the structure of $\mathcal{G}_P^{(k+m)}$.

Lemma 1: Let $P \in \mathcal{A}_\alpha^k$. For every $Q \in \mathcal{A}_\alpha^s$ with $1 \le s \le \min(k-1, m)$, $\mathcal{F}_P^{(m)}$ contains at least one sequence with prefix Q, and at least one sequence with suffix Q.

Proof: If $m < k$, $\mathcal{F}_P^{(m)} = \mathcal{A}_\alpha^m$, and the correctness of the lemma immediately follows. If $m \ge k$, let $P = p_1 p_2 \cdots p_k$, and let $\overline{p}_i \in \mathcal{A}_\alpha \backslash \{p_i\}$. It is obvious that for every $Q \in \mathcal{A}_\alpha^s$ the sequences $Q(\overline{p}_k)^{m-s}$ and $(\overline{p}_1)^{m-s}Q$ are elements of $\mathcal{F}_P^{(m)}$ if $1 \le s \le \min(k-1, m)$, and therefore the lemma holds. □

For a sequence P and a set of sequences $\mathcal{S}$, let us denote the set of concatenations of P and all the sequences in $\mathcal{S}$ by $P\mathcal{S}$, that is, $P\mathcal{S} = \{PW | W \in \mathcal{S}\}$. The null string, denoted by ϕ, is introduced to represent a string of length 0, for which $\phi X = X\phi = X$ holds. In this context, $P\mathcal{F}_P^{(0)} = \{P\}$ and $\mathcal{F}_P^{(0)} = \{\phi\}$ with cardinality 1. For any $m < 0, P\mathcal{F}_P^{(m)}$ is empty.

Theorem 1: For any PS-code $\mathcal{G}_P^{(k+m)}$ with prefix $P \in \mathcal{A}_\alpha^k$

$$\mathcal{G}_P^{(k+m)} \subseteq P\mathcal{F}_P^{(m)}. \tag{3}$$

Equality always holds if P is self-uncorrelated. Moreover, if (3) holds with equality for any $m \ge k-1$, then $P \circ P = 10^{k-1}$.

Proof: For $m \le k$, (3) holds according to (2). Since any sequence $PY \in \mathcal{G}_P^{(k+m)}$ satisfies

$$PYP \circ P = 10^{k-1+m}1(*)^{k-1}, \quad \text{for } m > k$$

$Y \circ P = 0^{m-k+1}(*)^{k-1}$ holds. Thus $Y \in \mathcal{F}_P^{(m)}$.

If P is self-uncorrelated,

$$PYP \circ P = 10^{m+k-1}10^{k-1}$$

holds. Since Y is any sequence in $\mathcal{F}_P^{(m)}, P\mathcal{F}_P^{(m)}$ is a PS-code and it is the largest one among all PS-codes of length $k+m$ and with prefix P of length k. That is, $P\mathcal{F}_P^{(m)} = \mathcal{G}_P^{(k+m)}$.

Now we assume that

$$P\mathcal{F}_P^{(m)} = \mathcal{G}_P^{(k+m)}, \quad \text{for } m \ge k-1.$$

Then

$$PWP \circ P = 10^{m+k-1}1(*)^{k-1}$$

for any $W \in \mathcal{F}_P^{(m)}$. Hence, for $1 \le i \le k-1$, no subblocks $L_i = p_{k-i+1} \cdots p_k w_1 \cdots w_{k-i}$ of PWP equal P. Thus the first i symbols of P must be different from the last i symbols of P, which shows $P \circ P = 10^{k-1}$. □

The key equation to construct the encoding/decoding algorithms for $\mathcal{G}_{P_G}^{(k+m)}$ is presented in the following theorem.

Theorem 2: For $\alpha \ge 2$ and $m \ge 1$

$$\mathcal{F}_{P_G}^{(m)} = \{1^m\} \cup \bigcup_{i=2}^{k-1} \{1^{i-1}0\mathcal{F}_{P_G}^{(m-i)}\} \cup \bigcup_{\substack{a \in \mathcal{A}_\alpha \\ a \ne 1}} \{a\mathcal{F}_{P_G}^{(m-1)}\} \tag{4}$$

where P_G is of the form $1^{k-1}0$.

Proof: According to the definition, $1^m \in \mathcal{F}_{P_G}^{(m)}$. Any other sequence $W \in \mathcal{F}_{P_G}^{(m)}$ starts at one of

$$\{1^{i-1}0 | 2 \le i < k\} \cup (\mathcal{A}_\alpha \backslash \{1\}).$$

If $W = 1^{i-1}0V$, then $V \in \mathcal{F}_{P_G}^{(m-i)}$ since

$$1^{i-1}0V \circ P_G = 0^{m-k+1}(*)^{k-1}$$

implies that

$$V \circ P_G = 0^{m-i-k+1}(*)^{k-1}.$$

Similarly, it is shown that if $W \in \mathcal{F}_{P_G}^{(m)}$ is represented as aV with $a \in \mathcal{A}_\alpha \backslash \{1\}$, then $V \in \mathcal{F}_{P_G}^{(m-1)}$.

Conversely, the first i components of $1^{i-1}0V \circ P_G$ are all 0 for any $V \in \mathcal{A}_\alpha^{m-i}$ since $1^{i-1}0 \circ 1^{k-1} = 0^i$ where $i < k$. Moreover, the last $m-i$ components of $1^{i-1}0V \circ P$ equal $0^{m-i-k+1}(*)^{k-1}$ for $V \in \mathcal{F}_{P_G}^{(m-i)}$ from (2). Therefore, we obtain that

$$1^{i-1}0V \in \mathcal{F}_{P_G}^{(m)}, \quad \text{for } V \in \mathcal{F}_{P_G}^{(m-i)}.$$

In the same way, we can show that

$$aV \in \mathcal{F}_{P_G}^{(m)}, \quad \text{for } V \in \mathcal{F}_{P_G}^{(m-1)}.$$

□

Remark 1: Let $\overline{P_G}$ be the negation of P_G, that is, $\overline{P_G} = 0^{k-1}1$. Then we obtain

$$\mathcal{F}_{\overline{P_G}}^{(m)} = \{0^m\} \cup \bigcup_{i=2}^{k-1} \{0^{i-1}1\mathcal{F}_{\overline{P_G}}^{(m-i)}\} \cup \bigcup_{\substack{a \in \mathcal{A}_\alpha \\ a \ne 0}} \{a\mathcal{F}_{\overline{P_G}}^{(m-1)}\}.$$

Let $\widetilde{P_G}$ be the reverse order of P_G, that is, $\widetilde{P_G} = 01^{k-1}$. Then we obtain

$$\mathcal{F}_{\widetilde{P_G}}^{(m)} = \{1^m\} \cup \bigcup_{i=2}^{k-1} \{\mathcal{F}_{\widetilde{P_G}}^{(m-i)}01^{i-1}\} \cup \bigcup_{\substack{a \in \mathcal{A}_\alpha \\ a \ne 1}} \{\mathcal{F}_{\widetilde{P_G}}^{(m-1)}a\}.$$

For a prefix $P_G = 1^{k-1}0$, we denote the cardinality of $\mathcal{G}_{P_G}^{(k+m)}$ by $G_{k,m}$. Note that $G_{k,m}$ is written as

$$G_{k,m} = |P_G\mathcal{F}_{P_G}^{(m)}| = |\mathcal{F}_{P_G}^{(m)}|.$$

Theorem 3: For a given $k \ge 1$, a sequence $G_{k,0}, G_{k,1}, G_{k,2}, \cdots$ satisfies the following recursion:

$$G_{k,m} = \begin{cases} \alpha^m, & m < k \\ (\alpha-1)G_{k,m-1} + \sum_{i=2}^{k-1} G_{k,m-i} + 1, & m \ge k. \end{cases} \tag{5}$$

Proof: If $m < k, G_{k,m}$ equals α^m according to (2). Since the sets on the right-hand side of the formula in Theorem 2 are distinct, we obtain

$$\begin{aligned} |\mathcal{F}_{P_G}^{(m)}| &= 1 + \sum_{i=2}^{k-1} |1^{i-1}0\mathcal{F}_{P_G}^{(m-i)}| + \sum_{\substack{a \in \mathcal{A}_\alpha \\ a \ne 1}} |a\mathcal{F}_{P_G}^{(m-1)}| \\ &= (\alpha-1)|\mathcal{F}_{P_G}^{(m-1)}| + \sum_{i=2}^{k-1} |\mathcal{F}_{P_G}^{(m-i)}| + 1. \end{aligned}$$

Equation (5) follows by replacing $|\mathcal{F}_{P_G}^{(m-i)}|$ with $G_{k,m-i}$. □

Remark 2: Mandelbaum's code $\mathcal{M}_{P_G}^{(k+m)}$ is a binary PS-code defined by $\mathcal{M}_{P_G}^{(k+m)} = 1^{k-1}0\mathcal{F}_{1^{k-1}}^{(m)}$. It is also subdivided into $k-1$ subsets as follows:

$$\mathcal{M}_{P_G}^{(k+m)} = \bigcup_{i=1}^{k-1} \left\{1^{i-1}0\mathcal{F}_{1^{k-1}}^{(m-i)}\right\}. \tag{6}$$

Since the derivation of (6) is similar to Theorem 2, we omit it. Moreover, let $M_{k,m}$ be the cardinality of $\mathcal{M}_{P_G}^{(k+m)}$. Then, we obtain that

$$M_{k,m} = \begin{cases} 2^m, & m \le k-2 \\ M_{k,m-1} + \cdots + M_{k,m-k+1}, & m \ge k-1. \end{cases} \tag{7}$$

By comparing (5) in case of $\alpha = 2$ with (7), we immediately know that $G_{k,m} > M_{k,m}$.

The exact difference in size between $\mathcal{M}_{P_G}^{(k+m)}$ and $\mathcal{G}_{P_G}^{(k+m)}$ is analyzed using the generating functions for $G_{k,m}$ and $M_{k,m}$, which are defined as follows:

$$G_k(z) = \sum_{m=0}^{\infty} G_{k,m} z^m \tag{8}$$

$$M_k(z) = \sum_{m=0}^{\infty} M_{k,m} z^m. \tag{9}$$

Then, using the recursions of (5) and (7), $G_k(z)$ and $M_k(z)$ can be written as

$$G_k(z) = \frac{1}{1-2z+z^k} \tag{10}$$

$$M_k(z) = \frac{1-z^{k-1}}{1-2z+z^k}. \tag{11}$$

Since $M_k(z) = (1-z^{k-1})G_k(z)$, we obtain

$$M_{k,m} = G_{k,m} - G_{k,m-k+1}, \qquad m \ge 0 \tag{12}$$

where $G_{k,i} = 0, i < 0$. A variation of (12) is given by

$$G_{k,m} = G_{k,m-1} + M_{k,m-1}, \qquad m \ge 1. \tag{13}$$

To obtain (13), we modify (5) as follows:

$$\begin{aligned} G_{k,m} &= G_{k,m-1} + \cdots + G_{k,m-k+1} + 1 \\ &= G_{k,m-1} + (G_{k,m-1} - G_{k,m-k}) \\ &= G_{k,m-1} + M_{k,m-1} \end{aligned}$$

where we use (12) to obtain the last equality.

Next, we will deduce approximated expressions for $G_{k,m}$ and $M_{k,m}$, which indicate the asymptotic behavior of the code size. Since their derivations are similar to those in [4], [7], we only give the results and the intermediate steps are omitted. Let r_k be the real root but 1 of the equation $x = 1/(2-x^{k-1})$. Then, for large m, we obtain

$$G_{k,m} \cong -\frac{1}{k-1} - \frac{r_k}{2kr_k - k - 1} r_k^{-(m+1)} \tag{14}$$

$$M_{k,m} \cong -\frac{1-r_k}{2kr_k - k - 1} r_k^{-(m+1)}. \tag{15}$$

As an example, approximations of $G_{4,m}$ and $M_{4,m}$ are shown in Table I. Since the first term of the right-hand side of (14) is a fractional number for any integer $k > 2$ and the second term increases as m goes to infinity, we state that

TABLE I
APPROXIMATION OF $G_{k,m}$ AND $M_{k,m}$ ($k = 4$)

m	$G_{4,m}$	(14)	$M_{4,m}$	(15)
1	2	1.9927	2	2.0291
2	4	4.0848	4	3.8480
3	8	7.9328	7	7.0775
4	15	15.010	13	13.018
5	28	28.028	24	23.943
6	52	51.971	44	44.038
7	96	96.009	81	80.999
8	177	177.01	149	148.98
9	326	326.00	274	274.02
10	600	600.01	504	504.00

$$M_{k,m} \cong \frac{1-r_k}{r_k} G_{k,m}, \quad \text{for } m \gg 1. \tag{16}$$

Thus the difference in redundancy between $\mathcal{M}_{P_G}^{(k+m)}$ and $\mathcal{G}_{P_G}^{(k+m)}$ is approximately given by $\log_2 r_k/(1-r_k)$. Note that this difference is determined only by the prefix length k and does not depend on the constrained sequence length m. For example, for k equal to 4, 6, and 8, the difference is equal to 0.253, 0.050, and 0.011, respectively.

IV. Coding Algorithms for MPS-Codes of Prefix P_G

In this section, we present the encoding and decoding algorithm for a class of PS-codes $\mathcal{G}_{P_G}^{(k+m)}$ of prefix $P_G = 1^{k-1}0$ for arbitrary $k \ge 1$ and $m \ge 1$. Note that this class contains the class of binary Gilbert's PS-codes $\mathcal{G}^{(n)}$. The algorithms will be extended for any self-uncorrelated prefix in Section IV. For the sake of simplicity, we will only discuss the binary alphabet case in this paper. The extension to nonbinary alphabet, however, can be easily obtained using the same arguments that have been developed here.

A. Encoding Algorithm

Theorem 2 shows that $\mathcal{F}_{P_G}^{(m)}$ can be subdivided into $k + \alpha - 2$ distinct subsets. By recursively applying this theorem to each subset except the singleton set (consisting of only one element), we know that $\mathcal{F}_{P_G}^{(m)}$ can be represented as a collection of $G_{k,m}$ singleton sets. We assume that input data is represented as a stream of binary block sequences, each of which corresponds to a number x with $0 \le x < G_{k,m}$. For a given m and y, with $0 \le y < 2^m$, let $\beta_m(y)$ be an m-bit binary sequence $\beta_m(y) = b_1 b_2 \cdots b_m$ such that

$$y = \sum_{i=1}^{m} b_{m+1-i} 2^{i-1}.$$

Conversely, for each binary sequence W of length m, let $\beta_m^{-1}(W)$ be a number y such that $\beta_m(y) = W$.

The main task of the encoding algorithm is to find a singleton set corresponding to an input x with $0 \le x < G_{k,m}$. The encoding algorithm consists of two parts: `EncodePSC` (k, m, x) and `CodePSC`(k, m, x). `EncodePSC`(k, m, x) calls

$\texttt{CodePSC}(k,m,x)$ to get sequence $\Omega_{k,m}(x)$ corresponding to a number x, $0 \le x < G_{k,m}$ and then returns the concatenation of P_G and $\Omega_{k,m}(x)$. The task of $\texttt{CodePSC}(k,m,x)$ is to construct $\Omega_{k,m}(x)$ with recursive calls

```
X = EncodePSC(k, m, x)
        Return X = P_G CodePSC(k, m, x).
                                    (End of EncodePSC)

Y = CodePSC(k, m, x)
  begin
1   if (m >= k) then begin
2     t := 1; y := x;
3     while (y >= G_{k,m-t}) do begin
4       y := y - G_{k,m-t};
5       t := t + 1;
      end;
6     if (t = k) then return (Y = 1^m)
7     else return (Y = 1^{t-1}0 CodePSC(k, m - t, y))
8   end else return (Y = β_m(x))
  end
                                    (End of CodePSC)
```

Example: Let us consider the encoding procedure of $\mathcal{G}_{P_G}^{(10)}$ with prefix $P_G = 1110$ where $m = 6$ and $k = 4$.

$\texttt{EncodePSC}(4,6,x)$ converts a number x from 0 to 51 into a codeword in $\mathcal{G}_{1110}^{(10)}$. For instance, tracing the encoding procedure for $x = 17$, we obtain the codeword $1110\Omega_{4,6}(17) = 1110010010$. Similarly, we obtain $\Omega_{4,6}(3) = 000011$ and $\Omega_{4,6}(42) = 101111$.

As shown in this example, the value of $\Omega_{k,m}(x)$ is recursively determined during the encoding process. Let

$$L_{k,m}[i] = \begin{cases} 0, & i = 1 \\ \displaystyle\sum_{j=1}^{i-1} G_{k,m-j}, & 1 < i < k. \end{cases}$$

Let us denote the set of integers $\{0,1,\cdots,G_{k,m}-1\}$ by $\mathcal{I}_{k,m}$ for $m \ge 1$. Then, we divide $\mathcal{I}_{k,m}$ into k distinct sets $\mathcal{I}_{k,m}[i]$ with $1 \le i \le k$, which are defined by

$$\mathcal{I}_{k,m}[i] = \begin{cases} \{j \mid L_{k,m}[i] \le j < L_{k,m}[i+1]\}, & 1 \le i < k \\ \{G_{k,m} - 1\} & i = k. \end{cases} \tag{17}$$

Theorem 4: $\Omega_{k,m}$ is a one-to-one mapping from $\mathcal{I}_{k,n}$ onto $\mathcal{F}_{P_G}^{(m)}$.

Proof: If $m < k$, then $\Omega_{k,m}(x)$ is the m-bit binary representation of x with $0 \le x < G_{k,m} = 2^m$, and $\mathcal{F}_{P_G}^{(m)} = \mathcal{A}_2^m$. Therefore, the theorem holds. For $m \ge k$, we use induction. We assume that $\Omega_{k,m-1}$ is a one-to-one mapping from $\mathcal{I}_{k,m-1}$ onto $\mathcal{F}_{P_G}^{(m-1)}$. Then, we prove that $\Omega_{k,m}$ is a one-to-one mapping from $\mathcal{I}_{k,m}$ onto $\mathcal{F}_{P_G}^{(m)}$.

First, we will show $\Omega_{k,m}$ maps $\mathcal{I}_{k,m}$ into $\mathcal{F}_{P_G}^{(m)}$. Suppose that $x \in \mathcal{I}_{k,m}[k]$, that is, $x = G_{k,m} - 1$. Then, the while-loop at step 3 in $\texttt{CodePSC}(k,m,x)$ is repeated $k-1$ times since

$$G_{k,m} - 1 = G_{k,m-1} + G_{k,m-2} + \cdots + G_{k,m-k+1}.$$

At the kth repetition of step 3, $y = 0, t = k$, and $G_{k,m-k} > 0$ for $m \ge k$. Therefore, the sequence 1^m is returned (step 6). It implies that $\Omega_{k,m}$ maps $G_{k,m} - 1$ into 1^m. Next, suppose that $x \in \mathcal{I}_{k,m}[i]\, i < k$. Then

$$\sum_{j=1}^{i-1} G_{k,m-j} \le x < \sum_{j=1}^{i} G_{k,m-j}. \tag{18}$$

The while-loop at step 3 is repeated $i-1$ times until the ith repetition, when

$$y = x - \sum_{j=1}^{i-1} G_{k,m-j} < G_{k,m-i}$$

and $t = i$. In step 6, since $i < k$ holds, $\texttt{CodePSC}(k, m-i, y)$ is called in step 7. Hence, if $x \in \mathcal{I}_{k,m}[i] (i < k)$, we can write

$$\Omega_{k,m}(x) = 1^{i-1}0\Omega_{k,m-i}(y) \tag{19}$$

where

$$y = x - \sum_{j=1}^{i-1} G_{k,m-j}.$$

From the assumption of induction, $\Omega_{k,m-i}(z) \in \mathcal{F}_{P_G}^{(m-i)}$ holds for $z \in \mathcal{I}_{k,m}[m-i]$ and $1 \le i \le m$. Since $0 \le y < G_{k,m-i}, y$ must be in $\mathcal{I}_{k,m-i}$. Therefore, $\Omega_{k,m-i}(y) \in \mathcal{F}_{P_G}^{(m-i)}$. Equation (19) shows that P_G does not appear in $1^{i-1}0\Omega_{k,m-i}(y)$ if $1 \le i \le k-1$. Thus

$$\Omega_{k,m}(x) \in \mathcal{F}_{P_G}^{(m)}. \tag{20}$$

Since (20) holds for any $x \in \mathcal{I}_{k,m}[i]$ and $1 \le i \le k$, we have $\Omega_{k,m}(\mathcal{I}_{k,m}) \subset \mathcal{F}_{P_G}^{(m)}$.

Now, we will show that $\Omega_{k,m}(x)$ is one-to-one. If $x \in \mathcal{I}_{k,m}[i]$ and $y \in \mathcal{I}_{k,m}[j]\ (i \ne j)$, the sequences corresponding to those numbers have distinct prefixes. Therefore, $\Omega_{k,m}(x) \ne \Omega_{k,m}(y)$. In case that x and y belong to the same $\mathcal{I}_{k,m}[i]$, the corresponding sequences can be represented as

$$\Omega_{k,m}(x) = 1^{i-1}0\Omega_{k,m-i}(x')$$
$$\Omega_{k,m}(y) = 1^{i-1}0\Omega_{k,m-i}(y')$$

respectively, where

$$x' = x - \sum_{j=1}^{i-1} G_{k,m-j}$$

and

$$y' = y - \sum_{j=1}^{i-1} G_{k,m-j}.$$

From the assumption of induction, $\Omega_{k,m-i}(x') \ne \Omega_{k,m-i}(y')$. Hence, $\Omega_{k,m}(x) \ne \Omega_{k,m}(y)$ holds. □

The time complexity of the encoding algorithm is evaluated as the number of comparisons of possibly large numbers y and $G_{k,m-t}$ at step 3 and the number of recursive calls at step 7. The sum of the numbers of comparisons and recursive calls is upper-bounded by m. Hence, the time complexity is $O(m)$. At most m values of $G_{k,i}$ $(1 \le i \le m)$ must be stored in memory to invoke $\texttt{EncodePSC}(k,m,x)$. Since $G_{k,i}$ can be represented by at most m bits, the total amount of memory is $O(m^2)$.

B. Decoding Algorithm

Suppose that the decoder receives a series of codewords in $\mathcal{G}_{P_G}^{(k+m)} = P_G\mathcal{F}_{P_G}^{(m)}$. After finding prefix P_G followed by an m-bit block $W = (w_1, w_2, \cdots, w_m)$ from the received sequence, the decoder converts $W \in \mathcal{F}_{P_G}^{(m)}$ into a number x where $0 \le x < G_{k,m}$. The following decoding algorithm returns a unique number for any $W \in \mathcal{F}_{P_G}^{(m)}$:

$x = \mathtt{DecodePSC}(k, m, W)$
begin
if $(m \ge k)$ then begin
if there exists $1 \le i < k$ such that $W = 1^{i-1}0V$ then
return
$$\left(x = \sum_{j=1}^{i-1} G_{k,m-j} + \mathtt{DecodePSC}(k, m-i, V)\right)$$
else return $(x = G_{k,m} - 1)$
end else return $(x = \beta_m^{-1}(W))$
end

(End of DecodePSC)

Let $\Xi_{k,m}(W)$ denote the returned value of $\mathtt{DecodePSC}(k, m, W)$. Then, $\Xi_{k,m}$ maps $\mathcal{F}_{P_G}^{(m)}$ to the set of integers.

Theorem 5: $\Xi_{k,m}$ is the inverse mapping of $\Omega_{k,m}$.

Proof: If $m < k$, then

$$\Xi_{k,m}(W) = \mathtt{DecodePSC}(k, m, W)$$

is a number x such that W equals the m-bit binary representation of x. Given $x, \mathtt{EncodePSC}(k, m, x)$ returns the m-bit binary representation of $x (0 \le x < 2^m)$. Hence, $\Xi_{k,m}(\Omega_{k,m}(x)) = x$. Now, suppose that

$$\Xi_{k,m-1}(\Omega_{k,m-1}(x)) = x$$

holds for $m \ge k$ and $x \in \mathcal{I}_{k,m-1}$. We will show that the assumption also holds when $m - 1$ is replaced by m. First we consider the case $x = G_{k,m} - 1$. As shown in the proof of Theorem 4, $\Omega_{k,m}(G_{k,m} - 1) = 1^m$ holds. Moreover, $\mathtt{DecodePSC}(k, m, 1^m)$ returns $G_{k,m} - 1$. Hence,

$$\Xi_{k,m}(\Omega_{k,m}(G_{k,m} - 1)) = G_{k,m} - 1$$

holds. Assume that $x \in \mathcal{I}_{k,m}[i]$ for $1 \le i < k$. Then, there exists a value r such that

$$x = \sum_{j=1}^{i-1} G_{k,m-j} + r$$

for which $0 \le r < G_{k,m-i})$. Moreover, according to (19), $\Omega_{k,m}(x)$ can be written as

$$\Omega_{k,m}(x) = 1^{i-1}0\Omega_{k,m-i}(r).$$

With $W = \Omega_{k,m}(x)$ and $V = \Omega_{k,m-i}(r), \Xi_{k,m}(W)$ can be written as

$$\Xi_{k,m}(W) = \sum_{j=1}^{i-1} G_{k,m-j} + \Xi_{k,m-i}(V).$$

From the assumption of induction, $\Xi_{k,m-i}(V) = r$. Therefore

$$\begin{aligned}\Xi_{k,m}(\Omega_{k,m}(x)) &= \sum_{j=1}^{i-1} G_{k,m-j} + \Xi_{k,m-i}(\Omega_{k,m-i}(r)) \\ &= \sum_{j=1}^{i-1} G_{k,m-j} + r \\ &= x.\end{aligned}$$

The proof is complete. □

Although DecodePSC needs the same amount of memory for storing the values of $G_{k,i}$ as EncodePSC does, the decoder is much faster than the encoder since no comparisons of two large integers are required in the decoding process.

V. Construction of MPS Codes with Arbitrary Self-Uncorrelated Prefixes

In practical situations, one might want to use another prefix than $P_G = 1^{k-1}0$. Although we can easily obtain the encoding and decoding algorithms for the negation of P_G or the reversed P_G, it seems to be hard to obtain a recursive relation on the partitions even for any self-uncorrelated prefix other than P_G. We will present the encoding and decoding algorithms for $\mathcal{G}_Q^{(k+m)}$ with a self-uncorrelated prefix Q. As in the previous section, we will only consider the binary alphabet case for the sake of simplicity. The algorithm can be easily extended to nonbinary alphabets.

We will describe a mapping Φ_Q to transform each sequence in $\mathcal{F}_{P_G}^{(m)}$ into another $\mathcal{F}_Q^{(m)}$ where Q is any prefix but P_G. If Q is self-uncorrelated, that is, $Q \circ Q = 10^{k-1}$, then it is shown that $\Phi_Q(\mathcal{F}_{P_G}^{(m)}) = \mathcal{F}_Q^{(m)}$. As a byproduct of this result, we obtain another proof of the statement that $\mathcal{G}_Q^{(k+m)}$ with a self-uncorrelated prefix Q has the same size as $\mathcal{G}_{P_G}^{(k+m)}$ [5].

The main idea of the mapping is to uniquely transform a sequence $X = \Omega_{k,m}(x)$ obtained from $\mathtt{EncodePSC}(k, m, x)$, to another sequence in $\mathcal{F}_Q^{(m)}$ where Q has the same length as P_G. Scanning X from the left to the right, check pattern X for the occurrence of Q. If we find Q as a subsequence of X, this subsequence is replaced by P_G. Let us denote the sequence obtained after the transformation by $\tau_{Q \to P_G}(X)$. After the conversion, no Q is supposed to appear in anywhere in $\tau_{Q \to P_G}(X)$, which means that the obtained sequence would be in $\mathcal{F}_Q^{(m)}$. Unfortunately, $\tau_{Q \to P_G}(X) \in \mathcal{F}_Q^{(m)}$ does not always hold, since the replacement of Q with P_G may cause Q to occur at a position which has been scanned before. For example, for $P_G = 1110$ and $Q = 1011$, let us convert the sequence $W = 0101011001 \in \mathcal{F}_{P_G}^{(10)}$. When we scan this sequence, Q is found at the fourth position, and W is converted to 0101110001. However, another Q now appears at the second position. Hence, replacing Q by P_G might result in scanning the sequence again and again. Fortunately, if we replace P_G by $\overline{P_G}$, then no Q appears at the position of W scanned before. In fact, W is converted to 0100001001 which belongs to $\mathcal{F}_Q^{(10)}$. In general, there exists a one-path scanning from left to right to uniquely transform $\mathcal{F}_{P_G}^{(m)}$ to $\mathcal{F}_Q^{(m)}$ or $\mathcal{F}_{\overline{P_G}}^{(m)}$ to $\mathcal{F}_Q^{(m)}$. In the rest of this section, we will show this method for correctly transforming $\mathcal{F}_{P_G}^{(m)}$ into $\mathcal{F}_Q^{(m)}$. The following lemma shows a sufficient condition for the existence of a "one-pass" transformation.

Lemma 2: Let S and T be distinct sequences of length k. Let X be a sequence in $\mathcal{F}_S^{(m)}$. If $T \circ S = 0^k$, then $\tau_{T\to S}(X) \in \mathcal{F}_T^{(m)}$.

Proof: Let us assume that T is found for the first time at the ith position of X. It means that X can be written as $X = VTW$ where V is a string of length $i-1$ such that $VT \circ T = 0^{i-1}1(*)^{k-1}$ and W is the remaining part of X. By replacing T with S at the ith position, X is converted to VSW.

To prove the Lemma, it is sufficient to show that $VS \circ T = 0^i(*)^{k-1}$. If another T is found at the jth ($j > i$) position, the prefix V' of length $j-1$ followed by T satisfies $V'T \circ T = 0^{j-1}1(*)^{k-1}$ and the situation is equivalent.

Suppose that the hth symbol of $VS \circ T$ is one, where $1 \le h < i-1$. Then, T can be written as $T = T_1T_2$ where T_1 is equal to a suffix of V and T_2 to a prefix of S. However, this contradicts $T \circ S = 0^k$. Therefore, we have $VS \circ T = 0^i(*)^{k-1}$. □

Lemma 3: For $Q \in \mathcal{A}_2^k$, if the last symbol of Q is zero and $Q \ne P_G$, then $Q \circ P_G = 0^k$. If it is one and $Q \ne \overline{P_G}$, then $Q \circ \overline{P_G} = 0^k$.

Proof: If $q_k = 0$, q_k is different from any symbol of P_G but the last one. Hence,

$$Q \circ P_G = 0^k, \quad \text{if } Q \ne P_G.$$

If $q_k = 1$, it is different from any symbol of $\overline{P_G}$ but the last one. Hence,

$$Q \circ \overline{P_G} = 0^k \quad \text{if } Q \ne \overline{P_G}.$$ □

Lemma 3 guarantees that if P_G or $\overline{P_G}$ is chosen appropriately depending on the "target" prefix Q, then the sufficient condition of Lemma 2 holds.

Using Lemmas 2 and 3, let us construct a mapping

$$\Phi_Q^{(m)}: \mathcal{F}_{P_G}^{(m)} \to \mathcal{F}_Q^{(m)}$$

as follows:

$$\Phi_Q^{(m)}(X) = \begin{cases} \tau_{Q\to P_G}(X), & \text{if } q_k = 0 \\ \overline{\tau_{\overline{Q}\to P_G}(X)}, & \text{otherwise.} \end{cases} \tag{21}$$

We note that $\Phi_Q^{(m)}(X) \in \mathcal{F}_Q^{(m)}$ holds for $X \in \mathcal{F}_{P_G}^{(m)}$.

Theorem 6: For any $Q \in \mathcal{A}_2^k, \Phi_Q^{(m)}$ is a one-to-one mapping from $\mathcal{F}_{P_G}^{(m)}$ to $\mathcal{F}_Q^{(m)}$ for $m \ge 1$.

Proof: We will show that

$$\Phi_Q^{(m)}(X) \ne \Phi_Q^{(m)}(Y) \quad \text{if } X \ne Y$$

for any pair $X, Y \in \mathcal{F}_{P_G}^{(m)}$. This statement will be proved using induction on m. First, let us consider the case that $q_k = 0$. If $m < k$, then the statement holds since $\Phi_Q^{(m)}(X) = X$ for $X \in \mathcal{A}_2^m$ and X belongs to both $\mathcal{F}_{P_G}^{(m)}$ and $\mathcal{F}_Q^{(m)}$ since $\mathcal{F}_{P_G}^{(m)} = \mathcal{F}_Q^{(m)} = \mathcal{A}_2^m$ from (2). Next, assume $m = k$. Then, if $X = Q$, then $\Phi_Q^{(k)}(Q) = P_G$. If $X \in \mathcal{F}_{P_G}^{(k)} \setminus \{Q\}$, then $\Phi_Q^{(k)}(X) = X$. Since $Q \in \mathcal{F}_{P_G}^{(k)}$ and $P_G \in \mathcal{F}_Q^{(k)}$, $\Phi_Q^{(k)}$ is one-to-one. Assuming that the statement holds for $m = t$, we will now prove that the statement holds for $m = t+1$. Let $X = x_1x_2\cdots x_{t+1}$ and $Y = y_1y_2\cdots y_{t+1}$ be in $\mathcal{F}_{P_G}^{(t+1)}$. We assume $X \ne Y$. Let W' denote the subsequence of W obtained by removing the first symbol. That is, $X' = (x_2, \cdots, x_{t+1})$ and $Y' = (y_2, \cdots, y_{t+1})$.

According to the values of the first k symbols of X and Y, we have four cases to consider:

i) $X_k \ne Q$ and $Y_k \ne Q$,
ii) $X_k = Q$ and $Y_k = Q$,
iii) $X_k = Q$ and $Y_k \ne Q$,
iv) $X_k \ne Q$ and $Y_k = Q$,

where $X_k = x_1x_2\cdots x_k$ and $Y_k = y_1y_2\cdots y_k$. For case i), we obtain

$$\Phi_Q^{(t+1)}(X) = x_1\Phi_Q^{(t)}(X') \tag{22}$$
$$\Phi_Q^{(t+1)}(Y) = y_1\Phi_Q^{(t)}(Y'). \tag{23}$$

If $x_1 \ne y_1$, the statement is obviously true. Otherwise, $X' \ne Y'$ must hold from the assumption. Then, by the inductive hypothesis, we have $\Phi_Q^{(t)}(X') \ne \Phi_Q^{(t)}(Y')$. Next, we consider case ii). By choosing appropriate sequences R and $S, (R \ne S)$ of length $t+1-k$, X and Y can be written as $X = QR$ and $Y = QS$, respectively. Then, we have

$$\Phi_Q^{(t+1)}(X) = 1\Phi_Q^{(t)}(1^{k-2}0R) \tag{24}$$
$$\Phi_Q^{(t+1)}(Y) = 1\Phi_Q^{(t)}(1^{k-2}0S). \tag{25}$$

By the hypothesis of induction, we obtain $\Phi_Q^{(t+1)}(X) \ne \Phi_Q^{(t+1)}(Y)$ since $R \ne S$. Because of the symmetry of both cases iii) and iv), it is sufficient to consider case iii). In case iii), X and Y can be written as $X = QT$ and $Y = Y_kU$. Then, we obtain

$$\Phi_Q^{(t+1)}(X) = 1\Phi_Q^{(t)}(1^{k-2}0T) \tag{26}$$
$$\Phi_Q^{(t+1)}(Y) = y_1\Phi_Q^{(t)}(Y_k'U). \tag{27}$$

If $y_1 \ne 1$, then the statement is true. Otherwise, $Y_k' \ne 1^{k-2}0$, since Y is assumed to be in $\mathcal{F}_{P_G}^{(m)}$. Hence, $Y_k'U \ne 1^{k-2}0T$. By the inductive hypothesis, we have

$$\Phi_Q^{(t)}(1^{k-2}0T) \ne \Phi_Q^{(t)}(Y_k'U).$$

Now we consider the case that $q_k = 1$. Using the same argument as for $q_k = 0$, we take the negation of Q, and show $\tau_{\overline{Q}\to P_G}$ is a one-to-one mapping from $\mathcal{F}_{P_G}^{(m)}$ to $\mathcal{F}_{\overline{Q}}^{(m)}$. Note that

$$\mathcal{F}_{\overline{Q}}^{(m)} = \overline{\mathcal{F}_Q^{(m)}}$$

and the correspondence between X and $\overline{X}$ is one-to-one. Thus

$$\Phi_Q^{(m)} = \overline{\tau_{\overline{Q}\to P_G}}$$

is one-to-one if $q_k = 1$. This completes the proof. □

Using Theorem 6, we can construct the coding algorithms of a PS-code $\mathcal{C}_Q^{(k+m)}$ for a self-uncorrelated prefix Q by combining the algorithms for P_G and one of the transformations $\tau_{Q\rightarrow P_G}$ and $\overline{\tau_{\overline{Q}\rightarrow P_G}}$. The PS-code $\mathcal{C}_Q^{(k+m)}$ constructed in this way is a subset of $\mathcal{G}_Q^{(k+m)}$ because of the existence of a one-to-one mapping from $\mathcal{F}_{P_G}^{(m)}$ to $\mathcal{F}_Q^{(m)}$. In what follows, we will show that $\mathcal{C}_Q^{(k+m)}$ equals $\mathcal{G}_Q^{(k+m)}$.

Now let us assume that $q_k = 0$, and consider a conversion $\tau_{Q\leftarrow P_G}$ from $\mathcal{F}_Q^{(m)}$ to $\mathcal{F}_{P_G}^{(m)}$ as follows: by scanning $W \in \mathcal{F}_Q^{(m)}$ from right to left, look for pattern P_G in W. If P_G is found as a subsequence of W, then it will be replaced by Q. We continue the above operation until we reach the left-end of W. The obtained sequence is denoted by $\tau_{Q\leftarrow P_G}(W)$. This conversion does not produce P_G at any position in W which has been already scanned. Therefore, the obtained sequence after the conversion belongs to $\mathcal{F}_{P_G}^{(m)}$. That is, $\tau_{Q\leftarrow P_G}$ is a mapping from $\mathcal{F}_Q^{(m)}$ into $\mathcal{F}_{P_G}^{(m)}$. If $q_k = 1$, replace P_G by $\overline{P_G}$. Then we similarly obtain a conversion $\tau_{Q\leftarrow \overline{P_G}}$ which transforms $\mathcal{F}_Q^{(m)}$ to $\mathcal{F}_{\overline{P_G}}^{(m)}$. Now we define a mapping $\Psi_Q^{(m)}(X)$ as follows:

$$\Psi_Q^{(m)}(X) = \begin{cases} \tau_{Q\leftarrow P_G}(X), & \text{if } q_k = 0 \\ \overline{\tau_{Q\leftarrow \overline{P_G}}(X)}, & \text{if } q_k = 1. \end{cases}$$

From the definition, $\Psi_Q^{(m)}(X)$ is a mapping from $\mathcal{F}_Q^{(m)}$ to $\mathcal{F}_{P_G}^{(m)}$. Let us note that

$$\overline{\tau_{Q\leftarrow \overline{P_G}}(X)} = \tau_{\overline{Q}\leftarrow P_G}(\overline{X}). \tag{28}$$

Theorem 7: If Q is self-uncorrelated, then $\Psi_Q^{(m)}$ is a one-to-one mapping from $\mathcal{F}_Q^{(m)}$ to $\mathcal{F}_{P_G}^{(m)}$.

Proof: Using the same arguments as in Theorem 6, we show that $\Psi_Q^{(m)}$ is one-to-one for $m \leq k$. Next, assuming that the statement holds for $m = t$, we will now prove that it also holds for $m = t+1$ with the condition $q_k = 0$. Let two distinct sequences $X = (x_1, x_2, \cdots, x_{t+1})$ and $Y = (y_1, y_2, \cdots, y_{t+1})$ be in $\mathcal{F}_Q^{(t+1)}$. Using the same notation as in Theorem 6, X and Y can be written as x_1X' and y_1Y', respectively. Let us write

$$\Psi_Q^{(t)}(X') = M_X R_X \tag{29}$$

$$\Psi_Q^{(t)}(Y') = M_Y R_Y \tag{30}$$

where M_X and M_Y are sequences of length $k-1$, and R_X and R_Y are sequences of length $t+1-k$. Before the last step of the conversion $\tau_{Q\leftarrow P_G}$, sequences X and Y must have been converted into $x_1M_XR_X$ and $y_1M_YR_Y$, respectively. Then, $R_X \neq R_Y$ or $R_X = R_Y$ may occur. In the former case, it always holds that $\Psi_Q^{(t+1)}(X) \neq \Psi_Q^{(t+1)}(Y)$ since neither R_X nor R_Y changes when converting the last part of X and Y. Hence, the only case where $\Psi_Q^{(t+1)}(X) = \Psi_Q^{(t+1)}(Y)$ holds is the latter case.

Assume that $R_X = R_Y$. If $x_1M_X = y_1M_Y$, then $\Psi_Q^{(t)}(X') = \Psi_Q^{(t)}(Y')$ follows. From the inductive hypothesis, $X' = Y'$ holds. Hence, $X = Y$ holds since $x_1 = y_1$. This contradicts $X \neq Y$. Thus $x_1M_X \neq y_1M_Y$ always holds when $R_X = R_Y$.

Now, we assume that $x_1M_X \neq y_1M_Y$. Without loss of generality, it is sufficient to consider the following four cases:

i) $x_1M_X = P_G$ and $y_1M_Y = Q$.
ii) $x_1M_X = P_G$ and $y_1M_Y \neq Q$.
iii) $x_1M_X \neq P_G$ and $y_1M_Y = Q$.
iv) $x_1M_X \neq P_G$, $x_1M_X \neq Q$, $y_1M_Y \neq P_G$, and $y_1M_Y \neq Q$.

In cases iii) and iv), neither x_1M_X nor y_1M_Y change at the last step of conversion $\tau_{Q\leftarrow P_G}$. From the assumption, $x_1M_X \neq y_1M_Y$ holds. Hence, we have

$$\Psi_Q^{(t+1)}(X) \neq \Psi_Q^{(t+1)}(Y).$$

In case ii), x_1M_X is converted into Q, while y_1M_Y keeps the same values. Hence,

$$\Psi_Q^{(t+1)}(X) \neq \Psi_Q^{(t+1)}(Y)$$

also follows.

The only remaining case to be considered is case i). We aim to show that case i) never holds if $Y \in \mathcal{F}_Q^{(t+1)}$. Let us break Y' into two parts, $Y_M = (y_2, \cdots, y_k)$ and $Y_R = (y_{k+1}, \cdots, y_{t+1})$. Since $Y \in \mathcal{F}_Q^{(t+1)}, y_1Y_M$ is not equal to Q while $y_1M_Y = Q$, which implies that a P_G exists with overlapping Y_M and Y_R. That is, a suffix of Y_M equals a prefix of P_G and a suffix of P_G equals a prefix of Y_R. Let Y_M^t be the sequence obtained after converting the suffix of Y_M into the corresponding prefix of Q. And let Y_R^t be the sequence obtained after converting the prefix of Y_R into the corresponding suffix of Q. If $Y_M^tY_R^t$ contains no P_G, then it equals M_YR_Y. Hence, a suffix of M_Y equals a prefix of Q. Even if P_G is found again before the scan reaches the left end of $Y_M^tY_R^t$, a suffix of M_Y equals a prefix of Q after all. This contradicts the assumption that $Q \circ Q = 10^{k-1}$.

If $q_k = 1$ and Q is replaced by $\overline{Q}$, we can use the same arguments as for $q_k = 0$ to show that $\tau_{\overline{Q}\leftarrow P_G}$ is a one-to-one mapping from $\mathcal{F}_{\overline{Q}}^{(m)}$ to $\mathcal{F}_{P_G}^{(m)}$.

Since $\overline{q_k} = 0$, from the arguments developed above, we have that $\tau_{\overline{Q}\leftarrow P_G}$ is a one-to-one mapping from $\mathcal{F}_{\overline{Q}}^{(m)}$ to $\mathcal{F}_{P_G}^{(m)}$. Using (28) and

$$\mathcal{F}_{\overline{Q}}^{(m)} = \overline{\mathcal{F}_Q^{(m)}}$$

$\Psi_Q^{(n)}$ is proved to be a one-to-one mapping from $\mathcal{F}_Q^{(m)}$ to $\mathcal{F}_{P_G}^{(m)}$. □

Corollary 1: Let P_G be $1^{k-1}0$ and let Q be a sequence of length k. If $Q \circ Q = 10^{k-1}$, then

$$|\mathcal{F}_Q^{(m)}| = |\mathcal{F}_{P_G}^{(m)}|, \quad \text{for } m \geq 1.$$

Proof: According to Theorem 6, there exists a one-to-one mapping from $\mathcal{F}_{P_G}^{(m)}$ to $\mathcal{F}_Q^{(m)}$. Thus, the inequality $|\mathcal{F}_{P_G}^{(m)}| \leq |\mathcal{F}_Q^{(m)}|$ always holds. If $Q \circ Q = 10^{k-1}$, there exists a one-to-one mapping from $\mathcal{F}_Q^{(m)}$ to $\mathcal{F}_{P_G}^{(m)}$ according to Theorem 7. Hence, $\mathcal{F}_Q^{(m)}$ has the same size as $\mathcal{F}_{P_G}^{(m)}$. □

VI. Conclusion

Encoding and decoding algorithms for a class of MPS-codes have been presented. The key idea used in the algorithms is to partition recursively the set $\mathcal{F}_{P_G}^{(m)}$ of the constrained sequences of length m in which pattern $P_G = 1^{k-1}0$ does not appear, from which it is straightforward to obtain the algorithms.

Moreover, a method to transform $\mathcal{F}_{P_G}^{(m)}$ into $\mathcal{F}_{P}^{(m)}$ with a self-uncorrelated prefix P has been obtained. Based on this method, the algorithms have been extended to construct MPS codes with arbitrary self-uncorrelated prefixes. The obtained algorithms provide us a variety of options of selecting a prefix since the majority of prefixes used in practical applications of frame synchronization [14] are known to be self-uncorrelated.

The time complexity of the algorithms is proportional to the code length n since we can adopt one of the linear-time string matching algorithms [15] for the transformation.

References

[1] S. W. Golomb, B. Gordon, and L. R. Welch, "Comma-free codes," *Can. J. Math.*, vol. 10, pp. 202–209, 1958.

[2] W. L. Eastman, "On the construction of comma-free codes," *IEEE Trans. Inform. Theory*, vol. IT-11, pp. 263–266, 1965.

[3] R. A. Scholtz, "Maximal and variable word-length comma-free codes," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 300–306, 1969.

[4] E. N. Gilbert, "Synchronization of binary messages," *IRE Trans. Inform. Theory*, vol. IT-6, pp. 470–477, 1960.

[5] L. J. Guibas and A. M. Odlyzko, "Maximal prefix-synchronized codes," *SIAM J. Appl. Math.*, vol. 35, pp. 401–418, Sept. 1978.

[6] D. M. Mandelbaum, "Synchronization of codes by means of Kautz's Fibonacci encoding," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 281–285, Mar. 1972.

[7] W. H. Kautz, "Fibonacci codes for synchronization control," *IEEE Trans. Inform. Theory*, vol. IT-11, pp. 284–292, 1965.

[8] D. T. Tang and L. R. Bahl, "Block codes for a class of constrained noiseless channels," *Inform. Contr.*, vol. 17, pp. 436–461, 1970.

[9] G. F. M. Beenker and K. A. S. Immink, "A generalized method for encoding and decoding runlength-limited binary sequences," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 751–754, Sept. 1983.

[10] K. A. S. Immink and H. D. L. Hollmann, "Prefix-synchronized runlength-limited sequences," *IEEE J. Sel. Areas Commun.*, vol. 10, pp. 214–222, Jan. 1992.

[11] R. M. Capocelli, "Comments and additions to 'Robust transmission of unbounded strings using Fibonacci representations'," *IEEE Trans. Inform. Theory*, vol. 35, pp. 191–193, Jan. 1989.

[12] H. Yamamoto and H. Ochi, "A new asymptotically optimal code for the positive integers," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1420–1429, Sept. 1991.

[13] L. J. Guibas and A. M. Odlyzko, "String overlap, pattern matching and nontransitive games," *J. Comb. Theory*, vol. 30, ser. A, pp. 183–208, 1981.

[14] R. A. Scholtz, "Frame synchronization techniques," *IEEE Trans. Commun.*, vol. COM-28, pp. 1204–1213, 1980.

[15] G. A. Stephen, *String Searching Algorithm*, vol. 3 of *Lecture Notes Series on Computing*. Singapore: World Scientific, 1994.