# Periodic Prefix-Synchronized Codes: A Generating Function Approach

Navin Kashyap, *Member, IEEE,* and David L. Neuhoff, *Fellow, IEEE*

***Abstract*— A generating function method is developed in order to select synchronization markers that maximize the timing span of period-2 periodic prefix-synchronized (PPS) sync-timing coding with small delay. Sync-timing codes are used in situations where conventional data synchronization is required, and data time stamps or time indices are also needed. A PPS code is a sync-timing code in which each encoded block of data is preceded by a synchronization marker, with the markers preceding successive blocks forming a periodic sequence with some period $p$. Since only PPS codes with small periods can have good rates at small delays, and since codes with $p = 1$ are simply Gilbert's prefix-synchronized codes which have been studied previously in the literature, this paper focuses on $p = 2$ codes. The generating function method, which extends that used by Guibas and Odlyzko to analyze $p = 1$ codes, enables one to find PPS codes with the largest possible timing span among codes with a given delay and rate. It is found that at low delays such optimized PPS codes offer significant advantages over cascaded and natural marker PPS codes. They also compare favorably with embedded-index codes. Finally, for asymptotically large delays, it is shown that the best $p = 2$ PPS codes operate at approximately the same rate and delay, but twice the timing span, of the best $p = 1$ codes.**

***Index Terms*— generating functions, periodic prefix-synchronized codes, prefix-synchronized codes, Shannon capacity**

## I. INTRODUCTION

Periodic prefix-synchronized (PPS) codes were introduced in [5]–[8] as a family of *synchronization with timing codes*, or *sync-timing codes* for short. Sync-timing codes perform two tasks. The first is that of conventional synchronization [11],[12] namely, to encode data with bits in such a way that when decoding begins in the middle of the encoded bit stream or when the encoded bits are corrupted by insertion, deletion or substitution errors, the decoder synchronizes and begins producing correct data symbols as soon as possible. The second task, which distinguishes sync-timing codes, is to encode the data so that after synchronizing, the decoder produces an estimate of the time index of each decoded symbol. More specifically, it produces the true time index modulo some integer $T$, called the *timing span* of the sync-timing code. The performance of a sync-timing code is quantified by its *coding rate* $R$, which is the number of information symbols per encoded bit, its *resynchronization delay* $D$, which is the required number of correctly received bits until resynchronization is guaranteed to occur, and its timing span $T$. Large rate (small redundancy), small delay and large timing span are desired. The attainable $(R, D, T)$ triples form a capacity region, whose boundary is described by a function $T(r, d)$ giving the largest timing span attainable by sync-timing codes with rate at least $r$ and delay at most $d$. In our work, we tend to fix a constraint on the rate, *e.g.* $R = .9$, and examine how timing span $T$ grows as delay $D$ is permitted to increase.

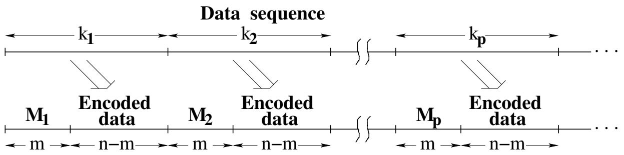

Fig. 1. A fragment of the encoded bitstream for PPS codes.

As in [5], to simplify discussion we assume that the data source produces binary symbols, *i.e.* bits. A $(p, m, n)$ PPS code is a binary block code characterized by a *period* $p$, *synchronizing markers* $M_1, \ldots, M_p$, which are distinct binary sequences each of length $m$, and codewords of length $n$. Specifically, corresponding to each marker $M_i$, there is a codebook $C_i$ containing $2^{k_i}$ binary sequences (codewords) of length $n$, all of which have $M_i$ as a prefix. Moreover, the codebooks are chosen so that when any codeword from $C_i$ is followed by any from $C_{i+1}$, no marker appears at any place except at the beginning of each codeword. Here, $C_{p+1}$ is to be interpreted as $C_1$. Encoding takes place by first parsing the sequence of data bits into blocks of length $K = \sum_{i=1}^{p} k_i$, and then in each block of length $K$, the first $k_1$ bits are encoded by a codeword from $C_1$, the next $k_2$ bits are encoded by one from $C_2$, and so on until the last $k_p$ bits are encoded by a codeword from $C_p$ (see Figure 1). Observe that the markers and codebooks are specifically designed to ensure that the markers are distinctly recognizable in any encoded sequence (no marker may appear in the code stream other than at the beginning of codewords), which allows the decoder to resynchronize in the event of errors. A PPS code with markers $M_1 = 000$, $M_2 = 111$, and codebooks $C_1 = \{0001100, 0001010\}$, $C_2 = \{1110011, 1110101\}$, is an example of a $(2, 3, 7)$ code.

The decoder operates as follows: it starts by looking for the first occurrence of a marker in the received binary sequence. If the first marker found is $M_i$, and this marker along with the $n - m$ subsequent received bits forms a codeword from $C_i$, then the decoder outputs the $k_i$ data bits encoded by this codeword. To these $k_i$ data bits, it assigns the time indices $k + 1, k + 2, \ldots, k + k_i$, where $k = \sum_{j=1}^{i-1} k_j$, defining $k_0$ to

Manuscript submitted November 2001, revised July 2005. This work was supported by NSF Grants NCR-9415754 and CCR-9815006, and was carried out while the first author was at the University of Michigan. Portions were published in the proceedings of the Data Compression Conference, Snowbird, Utah, March 1999.

Navin Kashyap is with the Department of Mathematics and Statistics at Queen's University, Kingston, ON K7L 3N6, Canada (email: nkashyap@mast.queensu.ca).

David L. Neuhoff is with the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor, MI 48109, U.S.A. (e-mail: neuhoff@eecs.umich.edu).

be 0. As result the index assigned to each decoded data bit is an estimate of the true time index modulo the integer $K$, which we call the *timing span* of the code. The decoder then begins another round of decoding by looking for the next marker. On the other hand, if the $n-m$ bits following the marker $M_i$ do not form a part of a codeword from $C_i$, then the decoder outputs nothing and renews its search for a marker, starting at the second bit of $M_i$. A more detailed description of the encoding and decoding of PPS codes can be found in [5].

The *rate* of the code is $R = K/N$, since each block of $K$ data bits is encoded into $N = pn$ coded bits, the *delay* of the code is $D = n$, because the decoder must wait for at most $n$ error-free bits to arrive until it sees a marker, and as mentioned earlier, the timing span is $T = K = NR = pDR$.

When $p = 1$, PPS codes reduce to prefix-synchronized codes, first studied by Gilbert [2] who analyzed their rate *vs.* blocklength. Further analysis was undertaken by Guibas and Odlyzko [3], and systematic encoding procedures were developed by Morita *et al* [10].

PPS codes are thus a generalization of Gilbert's prefix-synchronized codes. A generalization of Gilbert's codes in a different direction was considered recently by Wijngaarden and Morita [13]. Their *partial-prefix synchronizable codes* use a single distinguished marker that must appear at regular intervals in the encoded bitsream, but this marker is not required to appear as a contiguous subsequence of the encoded bitstream. The authors show that their codes are better than Gilbert's codes in terms of rate. It is very probable that the idea of allowing the markers to be non-contiguous strings can be used to improve the performance of PPS codes as well, but we do not pursue that idea in this paper.

The formula $T = pDR$ indicates that timing span for PPS codes increases with period $p$, provided that it remains possible to have codes with delay $D$ and rate $R$. Thus, they may offer substantially larger timing span than prefix-synchronized codes or partial-prefix synchronizable codes. Clearly, given constraints $r$ on rate and $d$ on delay, the timing span is maximized by the largest value $p$ for which there exists a period-$p$ PPS code with rate at least $r$ and delay at most $d$.

Previous work of the authors [5] established the following upper bound to the timing span attainable by PPS codes with rate at least $r$ and delay at most $d$:

$$T(r,d) \le 2^{d(1-r)+\log_2 d}. \tag{1}$$

It was also shown that for a rate constraint $r$ and large values of the delay constraint $d$, a structured family of PPS codes, called cascaded codes, can have timing span at least as large as $2^{d(1-r)+o(d)}$, where $o(d)$ denotes a quantity such that $o(d)/d \to 0$ as $d \to \infty$. Comparing this with the upper bound (1) reveals that cascaded codes have timing span increasing exponentially with delay, with asymptotically optimal exponent. Note that when delay is constrained to be small, the previous work shows how to compute the maximum timing span of cascaded codes for a given rate constraint, but it does not indicate how close these codes are to being optimal, in the sense of having the largest possible timing span among all PPS codes at that rate. This is the main topic of the present paper. In other words, we seek to optimize PPS codes with small delay and to compare the resulting performance to that of cascaded codes. We also compare the optimized PPS codes to embedded-index sync-timing codes, which is another family of sync-timing codes introduced in [5] that attains somewhat better performance than PPS codes, at the expense of greater encoding and decoding complexity. It must be pointed out that this paper is primarily concerned with determining the performance achievable by PPS codes, measured in terms of their rate, delay and timing span, and does not discuss the issue of the complexity of their implementation.

To optimize PPS codes with small delay, one must consider codes with small periods, because they are the only ones with the potential for small delay. To clarify the situation, let us fix a rate constraint $r$; let $D(p,r)$ denote the least delay of PPS codes with period $p$ and rate at least $r$; and let us assume that, as usually happens, $D(p+1,r) > D(p,r)$ for small values of $p$ and typical values of $r$. Defining $T_{PPS}(r,d)$ to be the largest possible timing span of a PPS code with rate at least $r$ and delay at most $d$, we see that when $0 < d < D(1,r)$, there are no PPS codes with delay at most $d$, and hence $T_{PPS}(r,d) = 0$. When $D(1,r) \le d < D(2,r)$, the only PPS codes with delay at most $d$ are those with period one, *i.e.* prefix-synchronized codes, and so $T_{PPS}(r,d) \approx rd$. When $D(2,r) \le d < D(3,r)$, there are PPS codes with periods one and two, and the latter give the largest timing span, implying $T_{PPS}(r,d) \approx 2rd$, and so on for larger values of $p$. In general, to find the largest possible timing span attainable with a small delay constraint $d$, one needs to find $D(p,r)$ for small values of $p$. This, in turn, is accomplished by finding the function $R(p,n)$, defined to be the largest rate of PPS codes with period $p$ and delay $D = n$. Then

$$D(p,r) = \min\{n : R(p,n) \ge r\} \tag{2}$$

and

$$T_{PPS}(r,d) \approx prd, \text{ for } D(p,r) \le d < D(p+1,r) . \tag{3}$$

With the above as motivation, in this paper we focus primarily on PPS codes with period $p = 2$. We need to find $R(2,n)$, and to do this, in Section II, we use a generating function method developed by Guibas and Odlyzko [3], [4] to analyze the rate $\widehat{R}(M_1, M_2, n)$, which is the largest rate attainable by a PPS code with period 2, codeword length $n$ and markers $M_1, M_2$.

The results of the generating function analysis of period-2 PPS codes are given in Section III, where we show the rates attainable for various choices of $m$, $n$ and marker pairs, and where we compare the rates, delays and timing spans attainable with those for other types of codes. For example, the least delays of various sync-timing codes with period $p = 2$ and rate $r = 0.9$ are shown in Table I. It is interesting to see from this table that optimal PPS codes are as good as prefix-synchronized embedded-index codes [5]. While not as good as comma-free embedded-index codes, they are considerably simpler to decode. They are, however, better than cascaded codes. Indeed, the reduction of the least delay from 120 for cascaded codes to 90 for optimal period-2 PPS codes means that for delay constraints between 90 and 119, optimal codes permit a doubling of the timing span (they can operate with

TABLE I

LEAST DELAYS OF SYNC-TIMING CODES WITH PERIOD 2 AND RATE AT LEAST 0.9.

| PPS Codes | | embedded-index codes | |
|---|---|---|---|
| cascaded | optimal | prefix-sync | comma-free |
| 120 | 90 | 90 | 80 |

$p = 2$ whereas cascaded codes are infeasible and so prefix-synchronized codes ($p = 1$) would have to be used instead).

It was shown by Gilbert [2] that when $m$ is fixed, for all sufficiently large $n$, the all-ones marker $1^m$ yields a prefix-synchronized code with the largest rate among all $(1, m, n)$ codes. It is natural, then, to seek the best choice of markers for PPS codes with period $p = 2$. Examination of the pairs of markers found to yield the largest rates suggests the conjecture that for a given marker length $m$ and all sufficiently large $n$, rate $\widehat{R}(M_1, M_2, n)$ is maximized by the marker pairs $(0^m, 1^m)$ and $(\langle 01 \rangle_m, \langle 10 \rangle_m)$, where the first pair contains the two constant sequences ($m$ 1's and $m$ 0's) and the second pair contains the two complementary sequences with alternating 0's and 1's. This conjecture is essentially proved in [9], where it is shown that given $m$, for all sufficiently large $n$, one or both of the aforementioned marker pairs maximizes rate. Indeed, we show here that for most values of $n$, $\widehat{R}(0^m, 1^m, n) = \widehat{R}(\langle 01 \rangle_m, \langle 10 \rangle_m, n)$.

For prefix-synchronized codes with codewords of length $n$ and no constraint on the marker length, it was conjectured by Gilbert [2] and proved by Guibas and Odlyzko [3] that the marker that maximizes rate has the form $1^m 0$ for some value of $m$. For period-2 PPS codes, while the general form of the best pair of markers is unknown, we provide in Table III a list of the best marker pairs for various values of $n$.

Asymptotic results for period-2 PPS codes are discussed in Section IV, and comparisons are drawn with the previously-known asymptotic results for period-1 codes, *i.e.* prefix-synchronized codes. The generating function approach of Guibas and Odlyzko [3],[4] shows that as codeword length $n$ tends to infinity, the rates of prefix-synchronized codes with marker $M$ converge to the Shannon capacity, $H(M)$, of the constrained system that forbids $M$. ($H(M)$ is defined to be $\lim_{n \to \infty} n^{-1} \log_2 q_M(n)$, where $q_M(n)$ denotes the number of sequences of length $n$ that do not contain $M$.) The above facts, along with the well-known fact [1] that $H(1^m) = \log_2 \rho_m$, $\rho_m$ being the largest real root of the polynomial $z^m - z^{m-1} - \ldots - 1$, show that $\log_2 \rho_m$ is the limit of the largest rate of prefix-synchronized codes with markers of length $m$.

For codes with period 2, the situation is more complex. While it is true that for arbitrary markers $M_1, M_2$, the rate $\widehat{R}(M_1, M_2, n)$ does approach a limit as $n$ increases, this limit need not always be equal to the Shannon capacity, $H(M_1, M_2)$, of the constrained system that forbids $M_1, M_2$. The exact relationship, derived in [9] and summarized in Section IV, does however show that for the best marker pairs, $(0^m, 1^m)$ and $(\langle 01 \rangle_m, \langle 10 \rangle_m)$, the limiting rate equals the corresponding Shannon capacity. This, along with the fact proved in [9] that $H(0^m, 1^m) = H(\langle 01 \rangle_m, \langle 10 \rangle_m) = \log_2 \rho_{m-1}$, implies that the maximum rate of a $(2, m, n)$ PPS code converges, as $n$ goes to infinity, to a limit, which is $\log_2 \rho_{m-1}$. In Theorem 3, we use the generating function of Section II to directly show that both $\widehat{R}(0^m, 1^m, n)$ and $\widehat{R}(\langle 01 \rangle_m, \langle 10 \rangle_m, n)$ converge towards $\log_2 \rho_{m-1}$ as $n$ tends to infinity. The convergence results for the maximum rates of period-1 and period-2 codes imply that for sufficiently large delays, there exist PPS codes with period 2 that operate at roughly the same rates and delays as the best period-1 codes. Thus, the timing spans provided by these period-2 codes are roughly double that provided by the best period-1 codes.

While we restrict ourselves to the study of PPS codes with period 2 in this paper, we do provide the necessary tools for using the generating function approach to study PPS codes with larger periods. In Appendix A, we present a system of linear equations that can be solved to obtain a generating function for the number of sequences satisfying the requirements for inclusion in a codebook of a period-$p$ code with a given set of markers.

## II. GENERATING FUNCTIONS

### A. Gilbert's Prefix-Synchronized Codes

As mentioned earlier, for $p = 1$, PPS codes are nothing but the prefix-synchronized codes introduced by Gilbert in 1960 [2]. We shall refer to them as Gilbert's prefix-synchronized codes or GPS codes. Given a binary sequence $M$ of length $m$, and an integer $n > m$, let $g_M(n)$ be the number of binary sequences that could be used as codewords in a GPS code with codeword length $n$ and marker $M$. Thus, defining $R(1, m, n)$ to be the maximum rate of a GPS code with marker length $m$ and codeword length $n$, we see that

$$R(1, m, n) = \max_M \frac{\lfloor \log_2 g_M(n) \rfloor}{n} \tag{4}$$

the maximum being taken over all markers $M$ of length $m$. As mentioned earlier, Gilbert showed that if $m$ is fixed, then for all sufficiently large $n$, $g_M(n)$ is maximized by choosing $M$ to be $1^m$, the all-ones sequence of length $m$. Therefore, for any $m$, if $n$ is sufficiently large, $R(1, m, n) = \lfloor \log_2 g_M(n) \rfloor / n$ with $M = 1^m$.

The work of Gilbert, and subsequently, that of Guibas and Odlyzko [3] contained methods for evaluating $g_M(n)$, for any choice of $M$, using generating functions. In particular, it follows from Guibas and Odlyzko's results that for $M = 1^m$ and $n > m$, $g_M(n)$ is the coefficient of $z^{-n}$ in the expansion of

$$G_M(z) = \frac{(2 - z) z^{m-1}}{z^m - z^{m-1} - \ldots - z - 1} \tag{5}$$

as $G_M(z) = \sum_{n=0}^{\infty} v_n z^{-n}$, *i.e.*, $v_n = g_M(n)$ for $n > m$.

For small values of $m$ and $n > m$, one can find $R(1, m, n)$ by using the generating function method to compute $g_M(n)$ for all length-$m$ markers $M$, and then applying (4). Plots of $R(1, m, n)$, for $m = 2, 3$ are shown in Figure 2 (along with other plots to be discussed later). The non-monotonicity of $R(1, m, n)$ is a consequence of the fact that the cardinality of the codebook of a GPS code is constrained to be a power of two, which manifests itself in the floor function used in the expression for $R(1, m, n)$ in (4).

The generating function in (5) can be used to infer the asymptotic limit of $R(1,m,n)$. It is a well-known fact (see *e.g.* [2],[15]) that for $m \geq 2$, the polynomial $z^m - z^{m-1} - \ldots - z - 1$ has $m-1$ zeros within the unit circle on the complex plane, and the remaining zero, $\rho_m$, is real and lies in the interval (1,2). Thus, the largest pole[1] of $G_M(z)$ is simple, and hence it follows from the theory of generating functions (*cf.* [14], Chapter 5) that $g_M(n) = \alpha \ (\rho_m)^n (1 + o(1))$. As a result, the following theorem is an immediate consequence of the fact that for all sufficiently large $n$, $R(1,m,n) = \lfloor \log_2 g_M(n) \rfloor / n$ with $M = 1^m$.

*Theorem 1:* For $m \geq 1$,

$$\lim_{n\to\infty} R(1,m,n) = \log_2 \rho_m$$

where $\rho_m$ is the zero of the polynomial $z^m - z^{m-1} - \ldots - z - 1$ that is largest in magnitude.

The fact that $R(1,m,n)$ converges to $\log_2 \rho_m$ as $n$ increases is illustrated in Figure 2, where $\log_2 \rho_m$ is plotted as an asymptote to $R(1,m,n)$ for $m = 2,3$.

In [15], it is also shown that $\rho_m$ increases with $m$, and that $2(1 - 2^{-m}) < \rho_m < 2$, from which it follows that $\lim_{m\to\infty} \rho_m = 2$. Hence, $\lim_{n\to\infty} R(1,m,n)$ is an increasing function of $m$, whose limit as $m \to \infty$ is 1. Thus, GPS codes asymptotically introduce no redundancy.

Closely related to the rate of a prefix-synchronized code with marker $M$ is the notion of the Shannon capacity of the *constrained system* of binary sequences that forbids $M$, which is defined to be

$$H(M) = \lim_{n\to\infty} \frac{\log_2 q_M(n)}{n}$$

where $q_M(n)$ is the number of binary sequences that do not contain $M$ as a contiguous subsequence. Guibas and Odlyzko derived generating functions for both $q_M(n)$ and $g_M(n)$ [3],[4], and it is easily seen that for any given $M$, both generating functions have the same set of poles. It then follows from the theory of generating functions that for any $M$,

$$\lim_{n\to\infty} \frac{\log_2 g_M(n)}{n} = \lim_{n\to\infty} \frac{\log_2 q_M(n)}{n} = H(M) \qquad (6)$$

This fact, though seemingly obvious, does not actually extend to PPS codes with period 2, as we shall see in Section IV.

## B. PPS Codes with Period 2

Using a result of Guibas and Odlyzko in [4], we derive a generating function for the number of sequences that are allowable as codewords in a period-2 PPS code with a given pair of markers $M_1, M_2$. This generating function will be used in the remainder of the paper to analyze the performance of period-2 PPS codes analogous to the analysis of GPS codes summarized above.

The expression we derive for the generating function uses the notion of the *correlation* $A \circ B$ between a pair of binary strings $A$ and $B$ [3], [4]. This is itself a binary sequence of the same length as $A$, whose $i$th bit (from the left) is determined as follows: place $B$ under $A$ in such a way that the first bit of $B$ lies under the $i$th bit of $A$; if the segments that overlap are identical, then the $i$th bit of $A \circ B$ is 1, else it is 0. Note that the leading bit of $A \circ B$ can be 1 if and only if $A$ is a prefix of $B$, or $B$ is a prefix of $A$. From this, it follows that if $A$ and $B$ have the same length, then the first bit of $A \circ B$ is a 1 if and only if $A = B$.

*Example*: Let $A = 110001$, $B = 1000$. Then, $A \circ B = 010001$, $B \circ A = 0000$, $A \circ A = 100001$, and $B \circ B = 1000$.

If $A \circ B = (c_{n-1} c_{n-2} \ldots c_0)$ is the correlation between two sequences $A$ and $B$, then we define the corresponding *correlation polynomial*

$$\phi_{AB}(z) = \sum_{i=0}^{n-1} c_i z^i \qquad (7)$$

For $A$ and $B$ of the previous example, $\phi_{AB}(z) = z^4 + 1$, $\phi_{BA}(z) = 0$, $\phi_{AA}(z) = z^5 + 1$, and $\phi_{BB}(z) = z^3$.

Now, given distinct binary sequences $A$ and $B$ and an integer $k > 0$, we define $f_{AB}(k)$ to be the number of binary sequences of length $k$ that begin with $A$ and end with $B$, but do not contain $A$ or $B$ anywhere else. Note that if $C_1$ and $C_2$ are the two codebooks of a $(2,m,n)$ code with markers $M_1$ and $M_2$, then $|C_1| \leq f_{M_1 M_2}(m+n)$, $|C_2| \leq f_{M_2 M_1}(m+n)$, where $|C_i|$ denotes the cardinality of $C_i$. Thus, if $\widehat{R}(M_1, M_2, n)$ denotes the maximum rate achievable by a $(2,m,n)$ code with markers $M_1$ and $M_2$, then

$$\begin{aligned} &\widehat{R}(M_1, M_2, n) \\ &\quad = \frac{\lfloor \log_2 f_{M_1 M_2}(m+n) \rfloor + \lfloor \log_2 f_{M_2 M_1}(m+n) \rfloor}{2n} \end{aligned} \qquad (8)$$

We define $\widehat{R}(M_1, M_2, n)$ to be 0 if there exists no $(2,m,n)$ code with markers $M_1$ and $M_2$.

The following theorem presents a generating function for $f_{AB}(k)$ in the case when the sequences $A$ and $B$ have the same length:

*Theorem 2:* Let $A$ and $B$ be distinct binary sequences of length $m$. For $k > m$, $f_{AB}(k)$ is the coefficient of $z^{-k}$ in the expansion of

$$F_{AB}(z) = \frac{1}{z} \; \frac{(z-2)\phi_{AB} + 1}{(z-2)D_{AB} + R_{AB}} \qquad (9)$$

as $F_{AB}(z) = \sum_{k=0}^{\infty} v_k z^{-k}$, where $D_{AB} = \phi_{AA}\phi_{BB} - \phi_{AB}\phi_{BA}$ and $R_{AB} = \phi_{AA} + \phi_{BB} - \phi_{AB} - \phi_{BA}$, the $\phi$'s being correlation polynomials as defined in (7).

*Proof*: Let $X$ be the sequence obtained by deleting the first bit of $A$. In other words, if $A = (a_1 a_2 \ldots a_m)$, then $X = (a_2 \ldots a_m)$. For $T \in \{A, B\}$ and $k > 0$, define $h_{XT}(k)$ to be the number of binary sequences of length $k$ that begin with $X$, end with $T$, and do not contain $A$ or $B$ as a substring except for the single instance of $T$ at the end. Also, define

[1] We need the largest pole here, instead of the smallest pole, because $G_M(z)$ is defined as a power series in $z^{-1}$.

$h_X(k)$ to be the number of binary sequence of length $k$ that begin with $X$ and do not contain $A$ or $B$ as a substring. For completeness, we define $h_{XT}(k) = h_X(k) = 0$ for $k = 0$. Let

$$H_{XT}(z) = \sum_{k=0}^{\infty} h_{XT}(k) z^{-k}, \quad H_X(z) = \sum_{k=0}^{\infty} h_X(k) z^{-k}$$

It is clear from the definitions that for any $k > m$, $f_{AT}(k) = h_{XT}(k-1)$. Thus, defining $F_{AT}(z) = \sum_{k=1}^{\infty} f_{AT}(k) z^{-k}$, we find that

$$F_{AT}(z) = z^{-1} H_{XT}(z) \tag{10}$$

Applying Theorem 2.1 of [4] to the sequences $X$, $A$ and $B$, we find that the generating functions $H_{XA}$, $H_{XB}$ and $H_X$ satisfy the following system of linear equations[2]:

$$\begin{aligned} (z-2)H_X + zH_{XA} + zH_{XB} &= z^{2-m} \\ -H_X + z\phi_{AA}H_{XA} + z\phi_{BA}H_{XB} &= z^{2-m}\phi_{XA} \\ -H_X + z\phi_{AB}H_{XA} + z\phi_{BB}H_{XB} &= z^{2-m}\phi_{XB} \end{aligned}$$

the $\phi$'s being correlation polynomials as defined in (7).

Eliminating $H_X$ from the above set of equations, we obtain

$$\begin{aligned} \gamma_{AA}H_{XA} + \gamma_{BA}H_{XB} &= z^{1-m}\gamma_{XA} \\ \gamma_{AB}H_{XA} + \gamma_{BB}H_{XB} &= z^{1-m}\gamma_{XB} \end{aligned}$$

where $\gamma_{**} = (z-2)\phi_{**} + 1$. Solving for $H_{XB}$ and plugging into (10), we obtain

$$F_{AB}(z) = \frac{z^{-m}(\gamma_{AA}\gamma_{XB} - \gamma_{AB}\gamma_{XA})}{\gamma_{AA}\gamma_{BB} - \gamma_{AB}\gamma_{BA}}$$

The theorem now follows by simplifying the expression for $F_{AB}(z)$ above using the facts that

$$\phi_{XA}(z) = \phi_{AA}(z) - z^{m-1}, \quad \phi_{XB}(z) = \phi_{AB}(z).$$

■

## III. Performance of Period-2 PPS Codes

For small values of $m$, it is practical to use Theorem 2 to exactly evaluate $R(2, m, n)$, which is the maximum rate achievable by a $(2, m, n)$ PPS code, for any given integer $n > m$. To be more specific, since

$$\begin{aligned} &R(2, m, n) \\ &\quad = \max\{\widehat{R}(M_1, M_2, n) : M_1, M_2 \in \{0,1\}^m, M_1 \neq M_2\} \end{aligned}$$

we can use (8) and (9) to maximize $\widehat{R}(M_1, M_2, n)$ over all pairs of markers $M_1$ and $M_2$ of length $m$.

In fact, we can reduce the number of marker pairs to be considered by partitioning the set of all pairs of length-$m$ sequences into equivalence classes as follows: two pairs of length-$m$ sequences $(M_1, M_2)$ and $(\tilde{M}_1, \tilde{M}_2)$ are assigned to the same equivalence class if $\{M_1 \circ M_1, M_2 \circ M_2\} = \{\tilde{M}_1 \circ \tilde{M}_1, \tilde{M}_2 \circ \tilde{M}_2\}$ and $\{M_1 \circ M_2, M_2 \circ M_1\} = \{\tilde{M}_1 \circ \tilde{M}_2, \tilde{M}_2 \circ \tilde{M}_1\}$. It easily follows from (8) and (9) that given an integer $n > m$, all marker pairs $(M_1, M_2)$ within the same equivalence class yield the same $\widehat{R}(M_1, M_2, n)$. While the notation $(M_1, M_2)$ is used to denote a pair of markers, it must be emphasized that the actual order of the markers within the pair is irrelevant, i.e., the pairs $(M_1, M_2)$ and $(M_2, M_1)$ are considered to be one and the same.

When $m = 2$, there are exactly three such equivalence classes: $\{(00, 11)\}$, $\{(01, 10)\}$ and $\{(00, 01), (11, 10), (00, 10), (11, 01)\}$. Now, for all odd $n > 2$, there exists no $(2, 2, n)$ code with markers from the first class because there is no binary sequence of odd length that begins with 00 and ends with 11, but does not contain 00 or 11 elsewhere. So, with $M_1 = 00$, $M_2 = 11$, we have $f_{M_1 M_2}(k) = f_{M_2 M_1}(k) = 0$, for all odd $k$. But for all even $k > 2$, $f_{M_1 M_2}(k) = f_{M_2 M_1}(k) = 1$, because sequences of the form $001010\ldots1011$ (resp. $110101\ldots0100$) are the only even-length sequences that can be counted by $f_{M_1 M_2}(k)$ (resp. $f_{M_2 M_1}(k)$). Therefore, we find that $\widehat{R}(00, 11, n) = 0$ for all $n$. Next, with markers $M_1$ and $M_2$ from the second equivalence class, the only sequences that can be counted by $f_{M_1 M_2}(k)$ and $f_{M_2 M_1}(k)$ are $01^{k-2}0$ and $10^{k-2}1$, respectively, so that $f_{M_1 M_2}(k) = f_{M_2 M_1}(k) = 1$ for all $k > 2$, which means that $\widehat{R}(01, 10, n) = 0$ for all $n$. Finally, it can be easily verified that for any $n > 2$, there does not exist a $(2, 2, n)$ PPS code for a marker pair chosen from the third equivalence class. Thus, we see that $R(2, 2, n) = 0$ for all $n > 2$.

When $m = 3$, it turns out that there are eleven equivalence classes. Table II lists $\widehat{R}(M_1, M_2, n)$ values for a representative marker pair $(M_1, M_2)$ from seven of these classes, for a few values of $n$. The remaining four equivalence classes are represented by the marker pairs (000,001), (001,011), (001,101) and (001,110). It can easily be shown using Theorem 2 that $\lim_{n\to\infty} \widehat{R}(M_1, M_2, n) = 0$ for any marker pair $(M_1, M_2)$ belonging to these four equivalence classes, and so they can be ignored. Thus, we can find $R(2, 3, n)$ by maximizing $\widehat{R}(M_1, M_2, n)$ over the seven equivalence classes listed in Table II.

Observe that the data in Table II indicates that for $L \geq 10$, $R(2, 3, 3+L)$ is achieved by codes with marker pairs from the equivalence classes represented by $(000, 010)$, $(000, 111)$ and $(010, 101)$. Similarly, from numerical data for the $m = 4$ case, it appears that the marker pairs $(0000, 1111)$ and $(0101, 1010)$ achieve $R(2, 4, n)$ for $n \geq 19$. Also, data for $m = 5$ suggests that for $n$ large enough, $R(2, 5, n)$ is achieved by the pairs $(00000, 11111)$ and $(01010, 10101)$.

Based on these observations, it is reasonable to make the conjecture, mentioned in the introduction, that given an integer $m \geq 3$, for all sufficiently large $n$, $R(2, m, n)$ is achieved by the marker pairs $(0^m, 1^m)$ and $(\langle 01\rangle_m, \langle 10\rangle_m)$, the former being the pair consisting of the all-zeros and all-ones sequences, and the latter being the pair consisting of the two complementary sequences with alternating 0's and 1's. It is easily verified that these pairs are the only members of their respective equivalence classes. This conjecture has essentially been proved in [9], where it is shown that for $m \geq 2$, at least one of the marker pairs $(0^m, 1^m)$ and $(\langle 01\rangle_m, \langle 10\rangle_m)$ achieves $R(2, m, n)$ for all sufficiently large $n$. In fact, a simple argument outlined in Appendix B shows that |f[illegible](k) − f[illegible](k)| ≤ 1 for all $k$, and hence due

[2] There is a slight error in the statement of Theorem 2.1 in [4]: in the right-hand sides of all but the first equation in (2.2), the expression $z^{1-|H|}$ should actually be $z^{1-|X|}$.

TABLE II
$\widehat{R}(M_1, M_2, 3+L)$ VALUES FOR VARIOUS $(M_1, M_2)$ PAIRS. BOLD VALUES ARE LARGEST IN THEIR RESPECTIVE ROWS.

| $L$ | $(000, 010)$ | $(000, 011)$ | $(000, 101)$ | $(000, 111)$ | $(001, 010)$ | $(001, 100)$ | $(010, 101)$ |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | **0.1250** | 0 |
| 2 | 0 | **0.1000** | 0 | 0 | **0.1000** | **0.1000** | 0 |
| 3 | **0.1667** | 0.0833 | **0.1667** | **0.1667** | 0.0833 | **0.1667** | **0.1667** |
| 4 | 0.1429 | **0.2143** | 0.1429 | 0.1429 | **0.2143** | **0.2143** | 0.1429 |
| 5 | 0.1250 | 0.1875 | **0.2500** | **0.2500** | 0.1875 | 0.1875 | **0.2500** |
| 6 | 0.2222 | 0.2222 | 0.2222 | 0.2222 | **0.2778** | 0.2222 | 0.2222 |
| 7 | **0.3000** | 0.2500 | **0.3000** | **0.3000** | 0.2500 | 0.2500 | **0.3000** |
| 8 | 0.2727 | 0.2273 | 0.2727 | **0.3636** | 0.3182 | 0.2273 | **0.3636** |
| 9 | **0.3333** | 0.2500 | **0.3333** | **0.3333** | 0.2917 | 0.2500 | **0.3333** |
| 10 | **0.3846** | 0.2692 | 0.3077 | **0.3846** | 0.3462 | 0.2692 | **0.3846** |
| 11 | **0.4286** | 0.2857 | 0.3571 | **0.4286** | 0.3214 | 0.2500 | **0.4286** |
| 12 | **0.4000** | 0.3000 | 0.3333 | **0.4000** | 0.3667 | 0.2667 | **0.4000** |
| 20 | **0.5217** | 0.3261 | 0.4348 | **0.5217** | 0.4130 | 0.3043 | **0.5217** |
| 50 | **0.6226** | 0.3774 | 0.4906 | **0.6226** | 0.5000 | 0.3208 | **0.6226** |
| 100 | **0.6505** | 0.3883 | 0.5243 | **0.6505** | 0.5194 | 0.3350 | **0.6505** |
| 500 | **0.6859** | 0.4016 | 0.5467 | **0.6859** | 0.5457 | 0.3449 | **0.6859** |
| 1000 | **0.6899** | 0.4038 | 0.5484 | **0.6899** | 0.5489 | 0.3460 | **0.6899** |

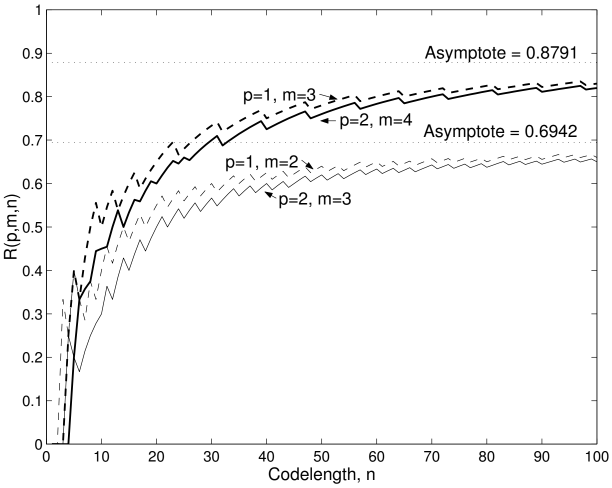

Fig. 2. The maximum rate $R(p, m, n)$ achievable by $(p, m, n)$ PPS codes, for various values of $p$ and $m$, as a function of $n$.

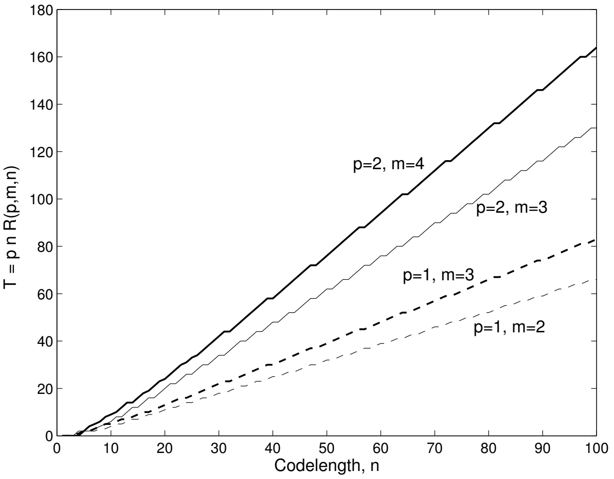

Fig. 3. The timing spans of $(p, m, n)$ PPS codes operating at rate $R(p, m, n)$, for various values of $p$ and $m$, as a function of $n$.

to the floor function used in defining $\widehat{R}(A, B, n)$, for nearly all (if not all) values of $n$, $\widehat{R}(0^m, 1^m, n) = \widehat{R}(\langle 01\rangle_m, \langle 10\rangle_m, n)$.

Based on rate calculations like those used to generate Table II, we plot $R(2, 3, n)$ and $R(2, 4, n)$ for $1 \leq n \leq 100$ in Figure 2. We shall show in the next section that as $n$ increases, $R(2, m, n)$ and $R(1, m-1, n)$ converge towards the same limit. Accordingly, for comparison, $R(1, 2, n)$ and $R(1, 3, n)$ are also shown in the figure, along with their limits from Theorem 1. As illustrated by the figure, for $m = 3, 4$, $R(2, m, n)$ and $R(1, m-1, n)$ approach each other rapidly, which means that for small to moderate values of $n$, PPS codes with period 2 essentially attain the promised doubling of timing span (recall that $T = pDR$). This is illustrated in Figure 3 which plots $T = pnR(p, m, n)$ for the values of $p$, $m$ and $n$ in Figure 2.

Fixing $p$ and $n$, and maximizing $R(p, m, n)$ over $m$ yields the $R(p, n)$ curves shown in Figure 4 for $p = 1, 2$. Note that for $p = 2$, this maximization is equivalent to maximizing $\widehat{R}(M_1, M_2, n)$ over pairs of markers $(M_1, M_2)$ of the same length. For $p = 1$, as mentioned in the introduction, it was conjectured by Gilbert [2] and subsequently proved (at least for all sufficiently large $n$) by Guibas and Odlyzko [4] that the GPS code using a marker of the form $1^m0$, for a suitable choice of $m$, attains the maximal rate of $R(1, n)$. For the $p = 2$ case, the general form of the best pair of markers is unknown, but we do provide a list of the best marker pairs for various values of $n$ in Table III. While for each value of $n$, there may be several marker pairs that are optimal, the table only lists one representative pair of markers. Note that because we allow the length of the markers to vary, there is no guarantee that the best markers are those demonstrated to be optimal for a fixed value of $m$. Indeed, the marker pairs $(0^m, 1^m)$ and $(\langle 01\rangle_m, \langle 10\rangle_m)$ never appear in the table. Although these pairs may be optimal for a few values of $n$, they are only rarely so. Nevertheless, due to their simplicity they are an interesting pair of markers to consider. Accordingly, Table III also lists for each $n$, the value of $m$ that maximizes $\widehat{R}(0^m, 1^m, n)$, and Figure 4 also plots the rates $\max_m \widehat{R}(0^m, 1^m, n)$. One can see from the figure that the optimal marker pairs improve over the simple $(0^m, 1^m)$ pair by a substantial amount for

TABLE III
MARKER PAIRS THAT ACHIEVE $R(2,n)$, AND MARKER LENGTHS $m$ THAT ACHIEVE $\max_m \widehat{R}(0^m, 1^m, n)$

| $n$ | best marker pair | argmax$_m \widehat{R}(0^m, 1^m, n)$ |
|---|---|---|
| 4 | (001,110) | 2,3 |
| 5 | (001,011) | 2,3,4 |
| 6 | (0001,1110) | 3 |
| 7 | (0001,0111) | 3,4 |
| 8 | (0001,1001) | 3,4 |
| 9 | (0001,1110) | 3,4,5 |
| 10 | (0001,0111) | 3,4,5 |
| 11 | (0001,0111) | 3,4 |
| 12 | (0001,1110) | 4 |
| 13 | (0011,0101) | 4 |
| 14 | (00001,10001) | 4 |
| 15 | (00001,11010) | 4,5 |
| 16 | (00011,00101) | 4,5 |
| 17 | (00011,00101) | 4,5 |
| 18 | (00001,00101) | 4,5 |
| 19 | (00000,10001) | 4,5 |
| 20 | (00001,00100) | 4,5 |
| 25 | (000001,010011) | 5 |
| 30 | (000001,000101) | 4,5,6 |
| 35 | (000000,100001) | 5,6 |
| 40 | (000001,001001) | 5,6 |
| 45 | (000001,010110) | 5,6 |
| 50 | (0000001,1000001) | 6 |
| 55 | (0000001,0000101) | 5,6,7 |
| 60 | (0001001,0100111) | 5,6,7 |
| 65 | (000001,001100) | 6,7 |
| 70 | (0000001,0001000) | 6,7 |
| 75 | (0000001,0001000) | 6,7 |
| 80 | (0000001,0001000) | 6,7 |
| 85 | (0000001,0010001) | 6,7 |
| 90 | (0000001,0100110) | 6,7 |
| 95 | (00000001,01000011) | 7 |
| 100 | (0000001,0000100) | 6,7,8 |

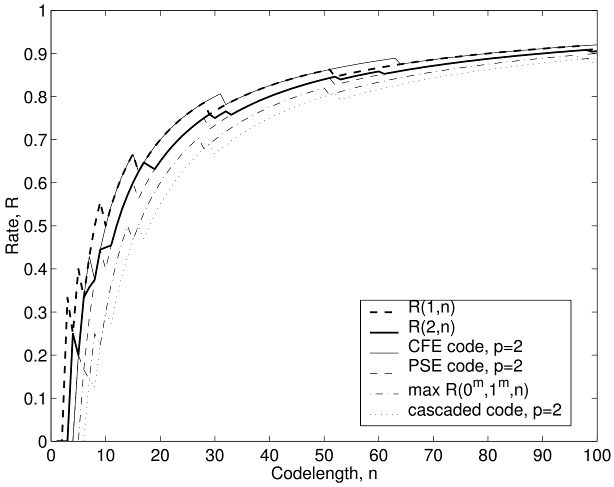

Fig. 4. Plots comparing the maximal rates of period-1 PPS codes ($R(1,n)$) and the following period-2 codes: optimal PPS ($R(2,n)$), comma-free embedded-index (CFE), prefix-synchronized embedded-index (PSE), period-2 cascaded, and PPS codes with marker pair $(0^m, 1^m)$ for an optimal choice of $m$.

values of $n$ less than $\sim$ 35, after which the improvement becomes less pronounced. For comparison purposes, we also plot the maximum rates achievable at each $n$ by cascaded codes, comma-free and prefix-synchronized embedded-index codes with $p = 2$ (see [5] for descriptions of these codes).

Finally, to put everything into perspective, Figure 5 plots a delay and timing span point $(D,T)$ for a number of sync-timing code classes. Specifically, for each class, $D$ is the least delay for which codes of that class have rate at least 0.9 (at most 10% redundancy), and $T = 0.9pD$ is the resulting timing span, as the codes represented in the figure actually turn out to have rates equal to 0.9 at their respective least delays. Accordingly, all codes shown with period $p$ lie on a straight line (also shown) passing through the origin with slope $0.9p$. For $p = 1$, we show only one point (asterisk) for the least delay GPS code. For $p = 2$, minimal delay codes are shown for the following classes: PPS codes, cascaded codes, and embedded-index codes – both prefix-synchronized and comma-free. Table I in Section I gives the least delay values for these period-2 codes. For $p = 3, 4$, Figure 5 shows the same points as for $p = 2$, except that we have not found the least delay PPS codes. For each category of codes, one can also attain, approximately, any $(D,T)$ value on the straight line emerging to the right of its point. We may conclude from this figure, that for delays below 80, there are no sync-timing codes with rates at least 0.9 and the maximal timing span is 0. For delays between 80 and 89, GPS codes ($p = 1$) and comma-free embedded-index (CFE) codes with $p = 2, 3$ are the only options. For these delays, CFE codes provide timing spans that are two or three times that of GPS codes, but at the expense of much higher encoding and decoding complexity. For delays between 90 and 99, both optimal period-2 PPS codes and prefix-synchronized embedded-index codes permit a doubling of timing span over GPS codes, and these codes are only marginally more complex than GPS codes. We also see that in comparison to cascaded codes with $p = 2$, optimal PPS codes permit the doubling to take place at $D = 90$ as opposed to $D = 120$. Thus optimizing period-2 PPS codes can make a notable difference if D is tightly constrained, whereas the simpler cascaded codes will be sufficient if the delay is not so tightly constrained. Finally, the points shown for $p = 3, 4$ indicate the delays at which codes with periods 3 and 4 can replace period-2 codes.

## IV. ASYMPTOTICS OF PERIOD-2 PPS CODES

As mentioned in the previous section, it has been proved in [9] that for $m \geq 2$, at least one of the marker pairs $(0^m, 1^m)$ and $(\langle 01 \rangle_m, \langle 10 \rangle_m)$ achieves $R(2,m,n)$ for all sufficiently large $n$. This, along with the fact proved in Appendix B that $|f_{\langle 01 \rangle_m \langle 10 \rangle_m}(k) - f_{0^m 1^m}(k)| \leq 1$ for all $k$, leads us to the asymptotic result for $R(2,m,n)$ stated in the next theorem, which is the period-2 analogue of Theorem 1. Although a proof of this result can be found in [9], for the sake of completeness, we include a (somewhat different) proof here.

*Theorem 3:* For $m \geq 2$,

$$\lim_{n \to \infty} R(2,m,n) = \log_2 \rho_{m-1}$$

where $\rho_{m-1}$ is the zero of the polynomial $z^{m-1} - \ldots - z - 1$ that is largest in magnitude.

*Proof*: We have already seen previously that $R(2,2,n) = 0$ for all $n > 2$, and so we need only consider the case when $m \geq 3$ (so that we have $\rho_{m-1} > 1$). Since at least one

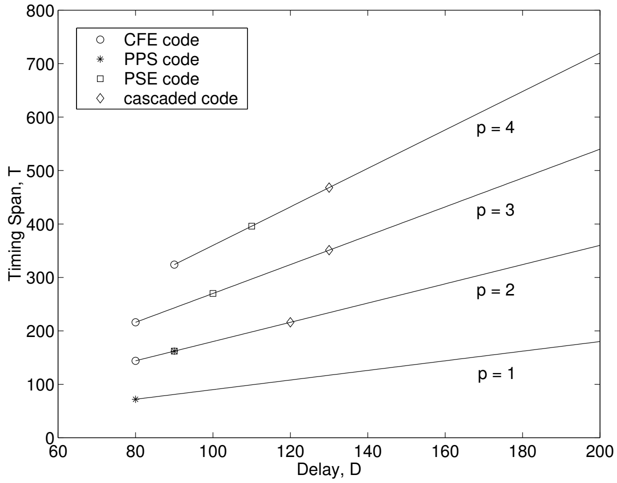

Fig. 5. Timing span vs. delay for various classes of sync-timing codes. The circles plot the timing spans of period-$p$ ($p = 2, 3, 4$) CFE codes at the least delays at which such codes exist with rate $R \geq 0.9$; the asterisks, squares and diamonds mark similar points for PPS codes, PSE codes and cascaded codes, respectively. Any point on the solid lines can be approximately attained by some code with $R \geq 0.9$.

of the marker pairs $(0^m, 1^m)$ and $(\langle 01\rangle_m, \langle 10\rangle_m)$ achieves $R(2, m, n)$ for all sufficiently large $n$, it is sufficient to show that $\lim\limits_{n\to\infty} \widehat{R}(0^m, 1^m, n) = \lim\limits_{n\to\infty} \widehat{R}(\langle 01\rangle_m, \langle 10\rangle_m, n) = \log_2 \rho_{m-1}$.

We note first that it is an easy consequence of Theorem 2 that $F_{\langle 01\rangle_m\langle 10\rangle_m}(z) = F_{\langle 10\rangle_m\langle 01\rangle_m}(z)$ and $F_{0^m1^m}(z) = F_{1^m0^m}(z)$, and hence that $f_{\langle 01\rangle_m\langle 10\rangle_m}(k) = f_{\langle 10\rangle_m\langle 01\rangle_m}(k)$ and $f_{0^m1^m}(k) = f_{1^m0^m}(k)$ for all $k$. By (8), this implies that $\widehat{R}(\langle 01\rangle_m, \langle 10\rangle_m, n) = \lfloor \log_2 f_{\langle 01\rangle_m\langle 10\rangle_m}(m + n)\rfloor/n$, and $\widehat{R}(0^m, 1^m, n) = \lfloor \log_2 f_{0^m1^m}(m + n)\rfloor/n$. But now, since $|f_{\langle 01\rangle_m\langle 10\rangle_m}(k) - f_{0^m1^m}(k)| \leq 1$, we see that $\lim\limits_{n\to\infty} \widehat{R}(\langle 01\rangle_m, \langle 10\rangle_m, n) = \lim\limits_{n\to\infty} \widehat{R}(0^m, 1^m, n)$. Hence, it suffices to show that $\lim_{n\to\infty} \widehat{R}(0^m, 1^m, n) = \log_2 \rho_{m-1}$.

An application of Theorem 2 to the marker pair $(0^m, 1^m)$ shows that

$$F_{0^m1^m}(z) = \frac{1}{z^2(z^{m-1} + \cdots + z + 1)(z^{m-1} - \ldots - z - 1)}$$

The polynomial $z^{m-1} + \cdots + z + 1$ is a factor of $z^m - 1$, and hence all its zeros lie on the unit circle $|z| = 1$. Moreover, since $z^{m-1} - \ldots - z - 1$ has exactly one zero, $\rho_{m-1}$, that lies outside the unit circle, we see that $\rho_{m-1}$ is the unique largest-magnitude pole of $F_{0^m1^m}(z)$, and it is a simple pole. Hence, by the reasoning used to derive Theorem 1, it follows that $\lim\limits_{n\to\infty} \widehat{R}(0^m, 1^m, n) = \lim\limits_{n\to\infty} \lfloor \log_2 f_{0^m1^m}(m + n)\rfloor/n = \log_2 \rho_{m-1}$. ■

One consequence of Theorems 1 and 3 is the fact that if we are willing to incur reasonably large delays, then we can get roughly double the amount of timing span, if not more, by using a period-2 PPS code instead of a period-1 code, with little or no loss in terms of rate or delay. This is because by the aforementioned theorems, for each $m \geq 3$, if $n$ is sufficiently large, then $R(1, m-1, n) \approx R(2, m, n)$. Now, the timing span of a $(2, m, n)$ code operating at rate $R(2, m, n)$ is $2nR(2, m, n)$, and hence for large enough $n$, this is approximately equal to $2nR(1, m-1, n)$, which is the timing span of a $(1, m-1, n)$ code operating at rate $R(1, m-1, n)$. This is illustrated for $m = 3, 4$ in Figure 3. It can be seen from this figure that even at reasonably small values of delay, the advantage gained by using period-2 codes instead of period-1 codes is significant.

Finally, we would like to mention an interesting relationship, analogous to (6), that exists between the limit, as $n$ goes to $\infty$, of the maximum rate, $\widehat{R}(A, B, n)$, of a $(2, m, n)$ PPS code with marker pair $(A, B)$, and the Shannon capacity, $H(A, B)$, of the constrained system of binary sequences that forbids $A$ and $B$. Formally, we define

$$H(A, B) = \lim_{n\to\infty} \frac{\log_2 q_{AB}(n)}{n}$$

where $q_{AB}(n)$ is the number of length-$n$ binary sequences that do not contain $A$ or $B$ as a contiguous subsequence.

Guibas and Odlyzko [4] showed that the generating function $Q_{AB}(z) = \sum_{n=0}^{\infty} q_{AB}(n) z^{-n}$ can be expressed as

$$Q_{AB}(z) = \frac{z\, D_{AB}}{(z - 2)D_{AB} + R_{AB}} \tag{11}$$

where $D_{AB}$ and $R_{AB}$ are as in the statement of Theorem 2. Now, if it could be shown that for any $A, B$, the generating functions $F_{AB}(z)$, $F_{BA}(z)$ and $Q_{AB}(z)$ have identical, unique largest-magnitude poles, then it would follow that $\lim_{n\to\infty} \widehat{R}(A, B, n)$ exists and equals $H(A, B)$. Comparing (9) and (11), we see that the denominators of $F_{AB}(z)$, $F_{BA}(z)$ and $Q_{AB}(z)$ are all the same. However, this is not even enough to guarantee that the largest-magnitude poles of $F_{AB}(z)$, $F_{BA}(z)$ and $Q_{AB}(z)$ are the same, since the largest-magnitude zero of the denominator polynomial may get cancelled out by a zero of the numerator polynomial in one or more of these generating functions. Indeed, this does happen for certain choices of $A, B$. For instance, it may be verified that when $A = 100$ and $B = 001$, the largest-magnitude pole of $F_{AB}(z)$ is 1, while that of $F_{BA}(z)$, as well as that of $Q_{AB}(z)$, is $\rho_2 \approx 1.618$. In this case, we have

$$\lim_{n\to\infty} \frac{\log_2 f_{AB}(n)}{n} = 0$$

but

$$\lim_{n\to\infty} \frac{\log_2 f_{BA}(n)}{n} = \lim_{n\to\infty} \frac{\log_2 q_{AB}(n)}{n} = \log_2 \rho_2 \approx 0.6942$$

Therefore, from (8), we see that $\lim_{n\to\infty} \widehat{R}(100, 001, n) = \frac{1}{2} H(100, 001)$.

However, as shown in [9], it is still true that $\lim_{n\to\infty} \widehat{R}(A, B, n)$ exists for all choices of $A, B$, and that in most cases, it equals $H(A, B)$, which is itself equal to the logarithm of the unique largest-magnitude pole of $Q_{AB}(z)$. In fact, the following statements are true for $m \geq 2$ (Proposition 20 in [9] states these for $m \geq 5$, but they can be directly verified for $m = 2, 3, 4$):

(a) If $\{A, B\}$ or $\{\overline{A}, \overline{B}\} = \{0^m, 0^{m-1}1\}$ or $\{0^m, 10^{m-1}\}$, then $\lim_{n\to\infty} \widehat{R}(A, B, n) = 0$, while $H(A, B) = \log_2 \rho_{m-1}$.

(b) If $\{A, B\}$ or $\{\overline{A}, \overline{B}\} = \{10^{m-1}, 0^{m-1}1\}$, then $\lim_{n\to\infty} \widehat{R}(A, B, n) = \frac{1}{2} H(A, B) = \frac{1}{2} \log_2 \rho$

(c) For all other pairs of distinct length-$m$ sequences $A, B$, $\lim_{n\to\infty} \widehat{R}(A, B, n) = H(A, B)$.

In statements (a) and (b), $\overline{A}, \overline{B}$ are the sequences obtained by complementing each bit of $A, B$. The above facts indicate that period-2 PPS codes exhibit more complex asymptotic behavior than period-1 codes, for which as we observed in Section II, the limiting code rate is always equal to the corresponding Shannon capacity.

## V. Concluding Remarks

Conceptually, it is possible to continue the above style of analysis to study the maximum rates of PPS codes with $p > 2$. Indeed, in Appendix A, we show how to derive generating functions for PPS codes with arbitrary $p$, similar to the one in Theorem 2. However, this kind of analysis is cumbersome, and the resultant expressions for generating functions are unwieldy even for $p = 3$. In fact, the difficulty of this is compounded by the fact that when $p > 2$, then even within a given set of markers, different orderings of the markers can yield different rates. This means that there are $(2^m)!/(2^m - p)!$ possible candidate orderings of markers, of which only a small proportion may be ruled out by inspection as being clearly sub-optimal. Naturally, a brute force search technique such as that described in Section III cannot be used for finding the maximal rate $R(p, m, n)$ in the general $(p, m, n)$ case.

It would be interesting to explore the relationships between $R(p, m, n)$ for various values of $p$, $m$ and $n$. For example, it seems intuitively clear, but as yet unproven, that for fixed $m$ and $n$, $R(p, m, n)$ should decrease with $p$, at least for sufficiently large values of $n$. Also, for arbitrary $p$ and $m$, the question of whether $\lim_{n\to\infty} R(p, m, n)$ exists remains unanswered, as does the question of how the limiting rate (if it exists) of a PPS code with a given set of $p$ length-$m$ markers is related to the corresponding Shannon capacity.

## Appendix A

It is possible, in principle, to derive a generating function for the number of binary sequences that can be included in a codebook of a PPS code with marker set $\mathcal{M} = \{M_1, M_2, \ldots, M_p\}$ containing a finite number of distinct length-$m$ binary markers. For $i, j \in \{1, 2, \ldots, p\}$ (not necessarily distinct) and $k > 0$, we define $f_{ij}(k)$ to be the number of binary sequences of length $k$ that begin with $M_i$, end with $M_j$, but do not contain any member of $\mathcal{M}$ elsewhere. We also define $f_i(k)$ to be the number of binary sequences of length $k$ that begin with $M_i$ and do not contain any member of $\mathcal{M}$ elsewhere. Let $F_{ij}(z)$ and $F_i(z)$ be the corresponding generating functions. We shall also find it convenient to define the polynomials $\gamma_{ij}(z) = (z-2)\phi_{ij}(z) + 1$, where $\phi_{ij}(z) = \phi_{M_i M_j}(z)$ is the correlation polynomial for the pair of markers $M_i$ and $M_j$. For the sake of simplicity, we shall henceforth drop the argument $z$ from the notation for the polynomials $\phi$ and $\gamma$, as well as for all the generating functions $F$. The following theorem provides a means of deriving an expression for $F_{ij}$.

*Theorem A.1:* Let $\mathbf{F} = (F_{ij})$ be the $p \times p$ matrix whose $(i, j)$th entry is the generating function $F_{ij}$, and let $\Gamma = (\gamma_{ij})$ be the $p \times p$ matrix whose entries are the polynomials $\gamma_{ij}$. Then,

$$\mathbf{F} = z^{-m} I - \frac{z-2}{z}\,\Gamma^{-1}$$

where $I$ is the $p \times p$ identity matrix.

*Proof*: Generalizing the ideas contained in the proof of Theorem 2, it can be shown that for $i, j = 1, 2, \ldots, p$, the generating functions $F_i$ and $F_{ij}$ satisfy the following system of linear equations:

$$\begin{aligned} (z-2)F_i + z\sum_{j=1}^{p} F_{ij} &= z^{1-m} \\ -F_i + z\sum_{j=1}^{p} \phi_{jk} F_{ij} &= z^{1-m}\phi_{ik} - \delta_{ik}, \quad k = 1, 2, \ldots, p \end{aligned}$$

where $\delta_{ik} = 1$ if $i = k$, and 0 otherwise.

Eliminating $F_i$ from the above equations, we get

$$\sum_{j=1}^{p} \gamma_{jk} F_{ij} = z^{-m}\gamma_{ik} - \frac{z-2}{z}\,\delta_{ik}$$

for $k = 1, 2, \ldots, p$.

But this system of linear equations is more compactly expressed in matrix form as

$$\mathbf{F}\,\Gamma = z^{-m}\Gamma - \frac{z-2}{z}\, I$$

which, upon multiplying by $\Gamma^{-1}$, proves the theorem. ■

## Appendix B

In this appendix, we show that $|f_{\langle 01\rangle_m \langle 10\rangle_m}(k) - f_{0^m 1^m}(k)| \le 1$ for all $k$, and hence for nearly all (if not all) values of $n$, $\widehat{R}(0^m, 1^m, n) = \widehat{R}(\langle 01\rangle_m, \langle 10\rangle_m, n)$.

From Theorem 2, it follows that the generating function for $f_{0^m 1^m}(k)$ is given by

$$F_{0^m 1^m}(z) = \frac{1}{z^2(z^{m-1} + \cdots + z + 1)(z^{m-1} - \ldots - z - 1)}$$

and that for $f_{\langle 01\rangle_m \langle 10\rangle_m}(k)$ is given by

$$F_{\langle 01\rangle_m \langle 10\rangle_m}(z) = \frac{(z-2)\left(\sum_{j=1}^{\lfloor m/2\rfloor} z^{m-2j}\right)}{(z^2)(z^{m-1} - \ldots - z - 1)\left(\sum_{j=1}^{m} (-1)^{j-1} z^{m-j}\right)}$$

Note also that $F_{\langle 01\rangle_m \langle 10\rangle_m}(z) = F_{\langle 10\rangle_m \langle 01\rangle_m}(z)$ and $F_{0^m 1^m}(z) = F_{1^m 0^m}(z)$, which implies that $f_{\langle 01\rangle_m \langle 10\rangle_m}(k) = f_{\langle 10\rangle_m \langle 01\rangle_m}(k)$ and $f_{0^m 1^m}(k) = f_{1^m 0^m}(k)$ for all $k$.

Now, $F_{\langle 01\rangle_m \langle 10\rangle_m}(z) - F_{0^m 1^m}(z)$ is a generating function for $f_{\langle 01\rangle_m \langle 10\rangle_m}(k) - f_{0^m 1^m}(k)$. After some algebraic manipulations, we find that

$$F_{\langle 01\rangle_m \langle 10\rangle_m}(z) - F_{0^m 1^m}(z) = \begin{cases} \frac{1}{z(z^m - 1)} & \text{if } m \text{ is even} \\ \frac{z^{m-1} - 1}{z^{2m} - 1} & \text{if } m \text{ is odd} \end{cases}$$

But $z^{-1}(z^m - 1)^{-1} = \sum_{r=1}^{\infty} z^{-(rm+1)}$, and $(z^{m-1} - 1)/(z^{2m} - 1) = z^{-(m+1)} - z^{-2m} + z^{-(3m+1)} - z^{-4m} + \cdots$. This shows that when $m$ is even,

$$f_{\langle 01\rangle_m \langle 10\rangle_m}(k) - f_{0^m 1^m}(k) = \begin{cases} 1 & \text{if } k = m+1, 2m+1, 3m+1, \ldots \\ 0 & \text{otherwise} \end{cases}$$

and when $m$ is odd,

$$f_{\langle 01\rangle_m \langle 10\rangle_m}(k) - f_{0^m 1^m}(k) = \begin{cases} 1 & \text{if } k = m+1, 3m+1, 5m+1, \dots \\ -1 & \text{if } k = 2m, 4m, 6m, \dots \\ 0 & \text{otherwise} \end{cases}$$

Thus, we have shown that $|f_{\langle 01\rangle_m \langle 10\rangle_m}(k) - f_{0^m 1^m}(k)| \leq 1$. Now, note that since $f_{\langle 01\rangle_m \langle 10\rangle_m}(k) = f_{\langle 10\rangle_m \langle 01\rangle_m}(k)$, we have $\widehat{R}(\langle 01\rangle_m, \langle 10\rangle_m, n) = \lfloor \log_2 f_{\langle 01\rangle_m \langle 10\rangle_m}(m+n)\rfloor/n$, and similarly, we see that $\widehat{R}(0^m, 1^m, n) = \lfloor \log_2 f_{0^m 1^m}(m+n)\rfloor/n$. The floor functions in these expressions ensure that for nearly all (if not all) values of $n$, $R(0^m, 1^m, n) = R(\langle 01\rangle_m, \langle 10\rangle_m, n)$.

## REFERENCES

[1] J.J. Ashley and P.H. Siegel, "A note on the Shannon capacity of run-length-limited codes," *IEEE Trans. Inform. Theory*. vol. 33, pp. 601–605, July 1987.

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

[3] L.J. Guibas and A.M. Odlyzko, "Maximal prefix-synchronized codes," *SIAM J. Appl. Math.*, vol. 35, no. 2, pp. 401–418, Sept. 1978.

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

[5] N. Kashyap and D.L. Neuhoff, "Data synchronization with timing," *IEEE Trans. Inform. Theory*, vol. 47, no. 4, pp. 1444–1460, May 2001.

[6] N. Kashyap and D.L. Neuhoff, "Codes for data synchronization with timing," *Proc. Data Compression Conference*, Snowbird, Utah, pp. 443–452, Mar. 1999.

[7] N. Kashyap and D.L. Neuhoff, "Codes for data synchronization and timing," *Proc. 1999 IEEE Information Theory and Communications Workshop*, pp. 63–65, Kruger National Park, South Africa, June 1999.

[8] N. Kashyap, "Data Synchronization with timing", Ph.D. Dissertation, University of Michigan, 2001.

[9] N. Kashyap, "Maximizing the Shannon capacity of constrained systems with two constraints," *SIAM J. Discr. Math.*, vol. 17, no. 2, pp. 276-297, 2003.

[10] H. Morita, A.J. van Wijngaarden, and A.J. Han Vinck, "On the construction of maximal prefix-synchronized codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, pp. 2158–2166, Nov. 1996.

[11] R.A. Scholtz, "Frame synchronization techniques," *IEEE Trans. Commun.*, vol. 28, no. 8, pp. 1204–1212, Aug. 1980.

[12] J.J. Stiffler, *Theory of Synchronous Communications*, Prentice-Hall, Englewood Cliffs, NJ, 1971.

[13] A.J. van Wijngaarden and H. Morita, "Partial-Prefix Synchronizable Codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 5, pp. 1839–1848, July 2001.

[14] H.S. Wilf, *generatingfunctionology*, 2nd ed., Academic Press, San Diego, CA, 1994.

[15] D.A. Wolfram, "Solving generalized Fibonacci recurrences," *The Fibonacci Quarterly*, vol. 36.2, pp. 129–145, May 1998.