Coding for the Optical Channel:
the Ghost-Pulse Constraint

Navin Kashyap, *Member, IEEE,* Paul H. Siegel, *Fellow, IEEE,* and Alexander Vardy, *Fellow, IEEE*

Abstract— We consider a number of constrained coding techniques that can be used to mitigate a nonlinear effect in the optical fiber channel that causes the formation of spurious pulses, called "ghost pulses." Specifically, if $b_1 b_2 \ldots b_n$ is a sequence of bits sent across an optical channel, such that $b_k = b_l = b_m = 1$ for some k, l, m (not necessarily all distinct) but $b_{k+l-m} = 0$, then the ghost-pulse effect causes b_{k+l-m} to change to 1, thereby creating an error. Such errors do not occur if the sequence of bits satisfies the following constraint: for all integers k, l, m such that $b_k = b_l = b_m = 1$, we have $b_{k+l-m} = 1$. We call this the binary ghost-pulse (BGP) constraint. We will show, however, that the BGP constraint has zero capacity, implying that sequences satisfying this constraint cannot carry much information. Consequently, we consider a more sophisticated coding scheme, which uses *ternary* sequences satisfying a certain ternary ghost-pulse (TGP) constraint. We further relax these constraints by ignoring interactions between symbols that are more than a certain distance t apart in the transmitted sequence. Analysis of the resulting BGP(t) and TGP(t) constraints shows that these have nonzero capacities, and furthermore, the TGP(t)-constrained codes can achieve rates that are significantly higher than those for the corresponding BGP(t) codes. We also discuss the design of encoders and decoders for coding into the BGP, BGP(t) and TGP(t) constraints.

Index Terms— Binary ghost-pulse (BGP) constraint, capacity of constrained systems, constrained encoding and decoding, optical communication, ternary ghost-pulse (TGP) constraint.

I. INTRODUCTION

High data-rate optical fiber communication presents several interesting challenges to a coding theorist. The diverse impairments peculiar to the optical channel necessitate the development of new coding schemes, capable of mitigating the effects of these impairments. One such impairment is the nonlinear effect known as intrachannel four-wave mixing (FWM)— see [9], [20], [23] and references therein. FWM results in strong inter-symbol interference between the symbols in a bitstream transmitted across the optical fiber. It is widely accepted [19], [20], [26] that at bit rates of 40 Gbps and beyond, FWM will play a major role in limiting the information-carrying capacity and the propagation distance of

a dispersion-managed optical communication system. In this paper, we consider a number of constrained coding techniques motivated by the intrachannel FWM effect.

A. Background on Ghost-Pulse Formation

In a typical optical fiber communication scenario, a train of light pulses, corresponding to a sequence of n bits, is sent across an optical fiber. Each bit in the sequence is allocated a time slot of duration T, and a binary one or zero is marked by the presence or absence of a pulse in that time-slot. The effect of intrachannel FWM is to transfer energy from triples of pulses in '1'-slots into certain '0'-slots, thereby creating spurious pulses known as *ghost pulses*. It has been observed that the interaction of pulses in the k-th, l-th, and m-th time-slots pumps energy into the $(k+l-m)$-th time-slot. If this slot did not originally contain a pulse — that is, if the $(k+l-m)$-th bit was a zero in the original n-bit sequence — then this transfer of energy creates a ghost pulse in this time-slot. This could cause the original zero to be read as a one (see Fig. **1**).

Since the overall energy is conserved, some of the pulses in the k-th, l-th, and m-th time-slots lose energy, resulting in a lowering of their amplitude (intensity). On the other hand, if the $(k+l-m)$-th slot already contained a pulse, then there is an exchange of energy between the pulses in the k-th, l-th, m-th, and $(k+l-m)$-th slots, leading to amplitude fluctuations. An analytic explanation of these phenomena can be derived using the nonlinear Schrödinger equation that describes pulse propagation in optical fibers — see [1], [2], [26].
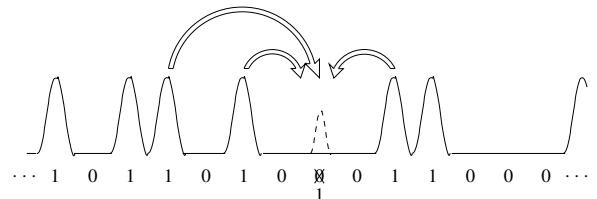

Fig. 1.   Model of ghost-pulse formation due to the interaction of three pulses

There are, in general, multiple (k, l, m) triples that result in the same integer $k+l-m$. Thus it is possible to have several pulse triples generating a ghost pulse at the same time-slot. Of course, in reality, the number of pulse triples involved in ghost-pulse formation at a certain time-slot is quite small. This is because, as one would expect from physical considerations, the interaction between pulses that are sufficiently far apart in the transmitted pulse train is weak. Indeed, for typical optical transmission parameters, pulses that are more than 10 to 12 time-slots apart do not contribute significantly to the formation of ghost pulses [3], [19]. In any case, when multiple pulse triples generate a ghost pulse at the same zero time-slot, the

resulting ghost pulse is the superposition of the ghost pulses formed by each of the pulse triples. The superposition of multiple ghost pulses may result in a stronger ghost pulse, or sometimes (due to destructive interference) in a weaker ghost pulse.

As shown in [4], [19], the phases of the original pulses play a vital role in determining which pulses lose energy and which gain energy in the course of the energy transfer induced by FWM. The phase of a ghost pulse created by a given pulse triple depends on the phases of all the pulses in the triple. Thus, in the case of a superposition of multiple triples, the relationship of the phase of the resulting ghost pulse to that of all the pulses involved in its creation can be quite complex. Indeed, even the amplitude of a ghost pulse depends on the phases of the pulse triples involved in its creation, since superposition of ghost pulses with opposing phases at the same time-slot will actually suppress ghost-pulse formation.

Physically, a ghost pulse is just another pulse of light. Thus it is possible that ghost-pulse formation may propagate: the interaction of a ghost pulse with actual pulses or other ghost pulses may lead to the creation of even more ghost pulses.

Finally, it should be noted that FWM is primarily a problem with long-haul and ultra long-haul optical communication systems, operating at 40 Gbps. This is so because the amplitude of ghost pulses grows linearly with propagation distance. A long-haul system consists of many periods of alternating spans of conventional and dispersion-compensating fiber. This causes quasi-periodic broadening and compression of the information-bearing pulses. For typical transmission parameters, ghost pulse amplitude reaches significant proportions over several periods of the dispersion map that is typically 50–100 km long. The simulations reported in the literature [19], [23], [26] were carried out over links of length 500 km to 5000 km.

B. Related Modulation Techniques

The optics literature has seen the emergence of several simple modulation schemes [4], [7], [16], [19] aimed at reducing the impact of FWM. Most of these schemes are based on the fact that FWM is a phase-sensitive effect and, therefore, can be controlled by modulating the phase of the pulses being transmitted. The one exception is the modulation scheme of [16], which proposes to use unequally spaced pulses at the expense of sacrificing spectral efficiency.

Coding — that is, introduction of redundancy in the transmitted bits as a means of controlling errors — has not been given much consideration as an approach to mitigating the FWM effect. To the best of our knowledge, the only previous work in this area has been reported by Vasic, Rao, Djordjevic, Kostuk, and Gabitov in [24]. In that paper, the authors use sequences satisfying a certain maximum-transition run (MTR) constraint to counter the impact of FWM. In the language of constrained coding, a binary sequence x is said to satisfy an MTR(j) constraint if every run of ones in x has length at most j (cf. [22]). In the modulation scheme of [24], a block code of rate 0.8, consisting of 256 binary codewords of length 10 satisfying the MTR(2) constraint, is used for transmission. Simulation results show significant ghost-pulse reduction due to the use of this coding scheme. The authors of [24] conclude

that "it is possible to successfully tackle the detrimental effects of FWM in 40-Gb/s systems using simple coding techniques."

In this paper, we undertake a systematic study of a number of coding schemes that combine constrained coding and phase modulation. Our study focuses purely on the coding-theoretic aspects (e.g. rate, encoding/decoding) of these schemes — we make no claims regarding their effectiveness in suppressing ghost pulses. In particular, we do not address the question of how well constrained coding schemes are suited to tackle the problem of eliminating ghost pulses in real-world optical systems. Such questions can only be answered via experimentation and/or extensive simulations of the fiber-optic channel, which is beyond the scope of this work.

C. Binary Ghost-Pulse Constraint

To formulate a well-defined coding problem, we model the formation of (primary) ghost pulses as follows. Let $b_1 b_2 \ldots b_n$, with $b_i \in \{0, 1\}$, be the binary sequence corresponding to the train of pulses sent across the fiber optic medium. If for some integers k, l, and m (not necessarily all distinct), we have

$$b_k = b_l = b_m = 1 \quad \text{while} \quad b_{k+l-m} = 0 \qquad (1)$$

then the formation of a primary ghost pulse converts the zero in time-slot $k+l-m$ to a one. Note that if we can encode the transmitted binary sequence in such a way that (1) never occurs, we will eliminate all (higher-order) ghost pulses caused by ghost-pulse propagation (discussed in Section I-A), as well. For example, a sequence containing at most one 1, or the all-ones sequence, or a sequence of alternating zeros and ones all satisfy this condition. In general, we say that a binary sequence $c_1 c_2 \ldots c_n$ satisfies the *binary ghost-pulse (BGP) constraint* if for all integers k, l, m such that $c_k = c_l = c_m = 1$ and $0 \leqslant k+l-m \leqslant n-1$, we also have $c_{k+l-m} = 1$. It is clear that transmitting a sequence that satisfies the BGP constraint will not allow ghost pulses to be created.

Let $f_{\text{BGP}}(n)$ be the number of binary sequences of length n that satisfy the BGP constraint. Then the asymptotic information rate (or the *capacity*, or the *entropy*) of the BGP constraint is defined (cf. [18], [21]) as follows:

$$H_{\text{BGP}} \stackrel{\text{def}}{=} \lim_{n \to \infty} \frac{1}{n} \log_2 f_{\text{BGP}}(n). \qquad (2)$$

Of course, we would like H_{BGP} to be as close to 1 as possible, so that coding into the BGP constraint adds little redundancy to the information being encoded. However, as we will show in Section III, a finite-length binary sequence satisfies the BGP constraint if and only if the ones in the sequence are uniformly spaced — that is, the positions of the ones form an arithmetic progression. It follows that there are $O(n^2)$ binary sequences of length n that satisfy the BGP constraint, and $H_{\text{BGP}} = 0$. Hence, we need to investigate alternative approaches to dealing with the ghost-pulse problem.

One approach that we consider is based on the intuition that the interaction between pulses that are sufficiently far apart in the transmitted pulse train is weak. As noted in Section I-A, in a typical optical communication scenario, pulses that are more than 10–12 time-slots apart do not contribute significantly to the formation of ghost pulses (cf. [3], [19]).

Disregarding the interaction between ones that are separated by more than some fixed distance t, we say that a binary sequence $c_1 c_2 \ldots c_n$ satisfies *the BGP(t) constraint* if for all integers k, l, m (not necessarily distinct) such that

$$c_k = c_l = c_m = 1, \tag{3}$$

$$0 \leqslant k + l - m \leqslant n - 1, \tag{4}$$

and

$$\max\big\{|k - l|, |l - m|, |m - k|\big\} \leqslant t \tag{5}$$

we also have $c_{k+l-m} = 1$. The capacity $H_2(t)$ of the BGP(t) constraint can be defined as in (2). (We will provide formal definitions for the capacities of all such constraints in the next section.) In Section III, we will show that $H_2(t)$ is positive for all t. However, we also show in Section III that $H_2(t)$ lies in the range $0.21 - 0.25$, when $t \in \{10, 11, 12\}$. This makes the BGP(t) constraint somewhat unattractive as the basis for a coding scheme. Nevertheless, we briefly discuss in Section III the design of finite-state encoders that take an unconstrained binary sequence as input and produce a BGP(t)-constrained sequence as output.

D. Ternary Ghost-Pulse Constraint

Another approach that has been suggested [4], [7], [19] to mitigate the formation of ghost pulses is to apply, at the transmitter end, a phase shift of π to some of the pulses. We can effectively think of this phase-modulation technique as converting a binary sequence $b_1 b_2 \ldots b_n$, with $b_i \in \{0, 1\}$, into a ternary sequence $c_1 c_2 \ldots c_n$, where $c_i \in \{-1, 0, 1\}$, such that $b_i = |c_i|$ for all i. One reason behind this phase-modulation approach is that, as explained in Section I-A, superposition of the contributions due to multiple pulse triples will result in suppression of ghost-pulse formation if their interference is destructive. Thus, knowledge of the relationship between the phase of a ghost pulse and the phases of the pulses involved in its creation makes it possible to manipulate the phase of the transmitted pulses in a way that encourages destructive interference. Such phase modulation schemes are very effective at eliminating some of the stronger ghost pulses (cf. [4], [19]). However, as observed in [4], it is impossible to achieve destructive interference in several consecutive zero-slots. Moreover, these schemes do not mitigate the "side ghosts" that arise due to energy leakage from the one-slots into adjacent zero-slots. Therefore, another approach is to modulate the phase of the transmitted pulses with the aim of achieving energy redistribution among the one-slots, thereby preventing energy leakage into adjacent zero-slots. Overall, building upon the work of [4], it appears reasonable to try preventing situations in which pulses in time-slots k, l, and m all have the same phase, while the slot at time $k + l - m$ is empty (zero).

Thus we say that a ternary sequence $c_1 c_2 \ldots c_n$ satisfies *the ternary ghost-pulse (TGP) constraint* if for all integers k, l, m (not necessarily distinct) such that $0 \leqslant k + l - m \leqslant n - 1$, and

$$c_k = c_l = c_m = +1 \quad \text{or} \quad c_k = c_l = c_m = -1 \tag{6}$$

we also have $c_{k+l-m} \neq 0$. Let $\mathcal{T}_3$ be the set of all finite-length ternary sequences that satisfy the TGP constraint. To transmit a finite-length binary data sequence, we encode it as a sequence

from $\mathcal{T}_3$. Based on the discussion above, we shall assume, as a first-order approximation, that sequences in $\mathcal{T}_3$ are effective in mitigating ghost-pulse formation, so the transmitted sequence can be recovered without error at the receiver end.

However, there is a catch. Most long-haul optical communication systems use direct-detection optical receivers, which can only detect the intensity (amplitude) of the optical signal at the channel output, not its phase. Thus if the transmitted ternary sequence was $c_1 c_2 \ldots c_n$, then the receiver only sees the sequence $|c_1|, |c_2|, \ldots, |c_n|$. In other words, the receiver cannot distinguish a $+1$ from a -1. As a result, we cannot use two sequences in $\mathcal{T}_3$ that differ only in phase (sign) to encode two different binary data sequences.

We thus have a rather unusual coding problem: even though the sequence being transmitted is ternary, the alphabet used for encoding information is effectively binary. In general, discrete channels for which the output alphabet is *smaller* than the input alphabet are rarely encountered in information theory. In fact, to the best of our knowledge, a situation where the alphabet over which the constraint is defined is different from the information-bearing alphabet has not been previously studied in the constrained coding literature.

In order to describe the procedure for encoding a binary data sequence using TGP-constrained sequences, we define the set

$$\mathcal{B}_3 \overset{\text{def}}{=} \big\{ |c_1|, |c_2|, \ldots, |c_n| \ : \ c_1 c_2 \ldots c_n \in \mathcal{T}_3 \big\}. \tag{7}$$

This is the set of all finite-length *binary* sequences that can be converted to a sequence in $\mathcal{T}_3$ by changing certain 1's to -1's. To transmit a binary data sequence $a_1 a_2 \ldots a_N$, we first encode it as a sequence $b_1 b_2 \ldots b_n \in \mathcal{B}_3$, which is then converted to a corresponding sequence $c_1 c_2 \ldots c_n \in \mathcal{T}_3$ at the input to an optical channel. At the channel output, the receiver detects the sequence $b_1 b_2 \ldots b_n$, which can be uniquely decoded to recover the original binary sequence $a_1 a_2 \ldots a_N$.

The capacity H_{TGP} of the TGP constraint can be now defined in a manner analogous to (2). Let $f_{\text{TGP}}(n)$ denote the number of sequences of length n in the set $\mathcal{B}_3$. Then

$$H_{\text{TGP}} \overset{\text{def}}{=} \lim_{n \to \infty} \frac{1}{n} \log_2 f_{\text{TGP}}(n). \tag{8}$$

The analysis of the TGP constraint appears to be a much more difficult problem than analysis of the BGP constraint. However, we conjecture that $H_{\text{TGP}} = H_{\text{BGP}} = 0$. Strong evidence in support of this conjecture is given in [15] (see Section IV-A).

Consequently, we consider the weaker TGP(t) constraint obtained, similarly to the BGP(t) constraint, by ignoring interactions between nonzero symbols that are more than distance t apart. Define the set $\mathcal{T}_{3;t}$ by adjoining the extra condition (5) to (6). The capacity $H_3(t)$ of the TGP(t) constraint can be then defined as in (8), but with respect to the set $\mathcal{T}_{3;t}$ rather than $\mathcal{T}_3$. One can reasonably expect that as t increases, $H_3(t)$ decreases, converging upon H_{TGP} in the limit as $t \to \infty$. Indeed, we will prove in the next section that

$$H_{\text{TGP}} = \lim_{t \to \infty} H_3(t) = \inf_{t \geqslant 1} H_3(t). \tag{9}$$

This provides a means of computing increasingly tight upper bounds on the capacity H_{TGP} which, as we mentioned earlier,

is not easy to compute directly. Furthermore, we will show in Section IV that

$$H_3(1) \;=\; 1 \quad \text{and} \quad H_3(2) \;\approx\; 0.96. \tag{10}$$

These values are significantly larger than the corresponding values for the BGP(t) constraint, namely

$$H_2(1) \;\approx\; 0.69 \quad \text{and} \quad H_2(2) \;\approx\; 0.55. \tag{11}$$

Moreover, it appears from (10) and (11) that $H_3(t)$ decreases much slower with t than $H_2(t)$, since $H_2(1) - H_2(2) \approx 0.14$ while $H_3(1) - H_3(2) \approx 0.04$. Assuming that this trend continues for larger values of t, coding schemes based on TGP(t)-constrained sequences can be a reasonably efficient means of mitigating the ghost-pulse effect in optical communications.

Unfortunately, the techniques we use in Section IV to determine $H_3(1)$ and $H_3(2)$ do not easily generalize to the computation of $H_3(t)$ for arbitrary t. Thus we have been unable to verify whether the aforementioned trend continues for larger values of t. In Section IV-D, we describe a general method for computing $H_3(t)$; however, this method is too computationally intensive to be implemented in practice. Nevertheless, we do discuss (also in Section IV) the design of finite-state encoders for coding schemes involving TGP(t)-constrained sequences.

Remark. Before concluding this introductory section, we note that it is possible to design other coding schemes that combine constrained coding with phase modulation in order to achieve ghost-pulse suppression. For example, we can conceivably add phase modulation to the constrained coding scheme of [24], thereby gaining some improvement in performance. In this paper, however, we have chosen to focus solely on the BGP and TGP constraints. The unusual nature of these constraints requires the development of non-standard tools for their analysis, which may be of independent interest to coding theorists.

II. DEFINITIONS AND PRELIMINARY RESULTS

In this section, we formally define the various types of ghost-pulse constraints that we shall be interested in. We also give precise definitions for the corresponding capacities, and establish several useful relationships between them.

Let $\mathbb{Z}$ and $\mathbb{Z}^+$ denote the set of integers and the set of positive integers, respectively. Given $n, n' \in \mathbb{Z}$, we write

$$[n] \;\overset{\text{def}}{=}\; \{\, i \in \mathbb{Z} \;:\; 1 \leqslant i \leqslant n \,\}$$
$$[n, n'] \;\overset{\text{def}}{=}\; \{\, i \in \mathbb{Z} \;:\; n \leqslant i \leqslant n' \,\}.$$

Note that both $[n]$ and $[n, n']$ could be empty. Let $\mathcal{A}_2 = \{0, 1\}$ and let $\mathcal{A}_3 = \{-1, 0, 1\}$. These are the relevant alphabets for the binary and the ternary ghost-pulse constraints, respectively. However, rather than giving definitions for the binary case and the ternary case separately, we find it more convenient to define the ghost-pulse constraints over a generic q-ary alphabet. Thus, given an integer $q \geqslant 2$, let $\mathcal{A}_q$ denote a fixed set of q letters, one of which is a distinguished letter 0. Although this is not required in what follows, a good way to think of $\mathcal{A}_q$ is as the set of distinct $q - 1$th roots of unity, augmented by zero. For $n \in \mathbb{Z}^+$, let $\mathcal{A}_q^n$ denote the set of sequences of length

n over $\mathcal{A}_q$. Given $\boldsymbol{x} = (x_1 x_2 \ldots x_n) \in \mathcal{A}_q^n$, the *support of* $\boldsymbol{x}$ is defined as $\mathrm{supp}(\boldsymbol{x}) = \{i \in [n] : x_i \neq 0\}$.

Definition 1. *A sequence $\boldsymbol{x} \in \mathcal{A}_q^n$ satisfies the q-ary ghost-pulse (q-GP) constraint if for all $k, l, m \in \mathrm{supp}(\boldsymbol{x})$ such that*

$$x_k \;=\; x_l \;=\; x_m$$

either $k + l - m \in \mathrm{supp}(\boldsymbol{x})$ or $k + l - m \notin [n]$. Note that the integers k, l, m above are not necessarily distinct.

For $n \in \mathbb{Z}^+$, let $\mathcal{T}_q(n)$ be the set of sequences of length n over $\mathcal{A}_q$ that satisfy the q-GP constraint. Further define

$$\mathcal{T}_q \;\overset{\text{def}}{=}\; \bigcup_{n=1}^{\infty} \mathcal{T}_q(n). \tag{12}$$

This is the set of all *finite-length* sequences satisfying the q-GP constraint. Let $\xi : \mathcal{A}_q \to \mathcal{A}_2$ be the "absolute value" function, defined by

$$\xi(x) \;\overset{\text{def}}{=}\; \begin{cases} 0 & x = 0 \\ 1 & x \neq 0. \end{cases} \tag{13}$$

For all $n \in \mathbb{Z}^+$, we extend this "absolute value" function componentwise to a function $\xi : \mathcal{A}_q^n \to \mathcal{A}_2^n$ via

$$\xi(x_1 x_2 \ldots x_n) \;\overset{\text{def}}{=}\; \big(\xi(x_1), \xi(x_2), \ldots, \xi(x_n)\big). \tag{14}$$

Given such a function, we further define for all $n \in \mathbb{Z}^+$ the sets $\mathcal{B}_q(n) \subset \mathcal{A}_2^n$ as follows

$$\mathcal{B}_q(n) \;=\; \xi\big(\mathcal{T}_q(n)\big) \;\overset{\text{def}}{=}\; \big\{\, \xi(\boldsymbol{x}) \;:\; \boldsymbol{x} \in \mathcal{T}_q(n) \,\big\}. \tag{15}$$

Finally, we set $\mathcal{B}_q = \xi(\mathcal{T}_q) = \bigcup_{n=1}^{\infty} \mathcal{B}_q(n)$. Thus, if $\mathcal{A}_q \backslash \{0\}$ is indeed a set of complex roots of unity, then $\mathcal{B}_q(n)$, respectively $\mathcal{B}_q$, consists of those *binary* sequences that can be transformed into a sequence in $\mathcal{T}_q(n)$, respectively $\mathcal{T}_q$, by means of appropriate phase shifts. In particular, our definition of $\mathcal{B}_3$ based upon (15) coincides with the earlier definition in (7).

Definition 2. *For all integers $q \geqslant 2$, the capacity of the q-ary ghost-pulse constraint is defined by*

$$H_q \;\overset{\text{def}}{=}\; \lim_{n \to \infty} \frac{\log_2 |\mathcal{B}_q(n)|}{n}. \tag{16}$$

It should be immediately clear from the discussion above that $H_2 = H_{\text{BGP}}$ and $H_3 = H_{\text{TGP}}$, as defined in (2) and (8) respectively. The following proposition shows that all these capacities are, indeed, well-defined.

Proposition 1. *The limit below exists for all $q \geqslant 2$, and moreover*

$$\lim_{n \to \infty} \frac{\log_2 |\mathcal{B}_q(n)|}{n} \;=\; \inf_{n \geqslant 1} \frac{\log_2 |\mathcal{B}_q(n)|}{n}. \tag{17}$$

Proof. This follows from the standard argument for shift spaces (see e.g. [18, pp. 103–104]), which we briefly reproduce here for completeness. Use the following test for convergence from elementary calculus: if $a_1, a_2, \ldots$ is a sequence of non-negative numbers such that $a_{n+n'} \leqslant a_n + a_{n'}$ for all $n, n' \geqslant 1$, then $\lim_{n \to \infty} a_n / n$ exists and equals $\inf_{n \geqslant 1} a_n / n$. Apply this

test to the sequence defined by $a_n = \log_2 |\mathcal{B}_q(n)|$. We need to show that $a_{n+n'} \leqslant a_n + a_{n'}$ or, equivalently, that

$$|\mathcal{B}_q(n+n')| \leqslant |\mathcal{B}_q(n)| \, |\mathcal{B}_q(n')| . \quad (18)$$

But this easily follows from the observation that if a sequence $\boldsymbol{y} \in \mathcal{A}_q^{n+n'}$ satisfies the q-GP constraint, then every contiguous subsequence of $\boldsymbol{y}$ also satisfies the q-GP constraint. Hence if $(x_1 x_1 \ldots x_{n+n'}) \in \mathcal{B}_q(n+n')$, then $(x_1 x_2 \ldots x_n) \in \mathcal{B}_q(n)$ and $(x_{n+1} x_{n+2} \ldots x_{n+n'}) \in \mathcal{B}_q(n')$, which implies (18). ∎

We will show in Section III-A that $H_2 = 0$, and our analysis in Section IV-A will lead us to conjecture that $H_3 = 0$ as well. In fact, we believe that $H_q = 0$ for all q. This is so because the q-GP constraint has unbounded memory. For large n, the value of a sequence $\boldsymbol{x} \in \mathcal{T}_q(n)$ at a given position $i \in [n]$ depends on the values of $\boldsymbol{x}$ at (essentially) all other positions. To obtain nonzero capacities, we relax the q-GP constraint by bounding its effective memory, as made precise in the next definition. As explained in Section I, it makes physical sense to do so.

Definition 3. *Let $t \in \mathbb{Z}^+$ be fixed. A sequence $\boldsymbol{x} \in \mathcal{A}_q^n$ satisfies the q-GP(t) constraint if for all $k, l, m \in \mathrm{supp}(\boldsymbol{x})$ such that*

$$x_k = x_l = x_m \quad \text{and} \quad \max\{|k{-}l|, |l{-}m|, |m{-}k|\} \leqslant t$$

either $k + l - m \in \mathrm{supp}(\boldsymbol{x})$ or $k + l - m \notin [n]$. As before, the integers k, l, m above need not be all distinct.

For $n, t \in \mathbb{Z}^+$, we let $\mathcal{T}_{q;t}(n)$ denote the set of sequences of length n over $\mathcal{A}_q$ satisfying the q-GP(t) constraint, and define $\mathcal{T}_{q;t} = \bigcup_{n=1}^{\infty} \mathcal{T}_{q;t}(n)$ as in (12). With the help of the function $\xi : \mathcal{A}_q^n \to \mathcal{A}_2^n$ given by (13) and (14), we define

$$\mathcal{B}_{q;t}(n) = \xi\big(\mathcal{T}_{q;t}(n)\big) \stackrel{\text{def}}{=} \big\{ \xi(\boldsymbol{x}) \, : \, \boldsymbol{x} \in \mathcal{T}_{q;t}(n) \big\} \quad (19)$$

and write $\mathcal{B}_{q;t} = \xi(\mathcal{T}_{q;t}) = \bigcup_{n=1}^{\infty} \mathcal{B}_{q;t}(n)$. We can now define the capacity of the q-GP(t) constraint as follows.

Definition 4. *For all integers $q \geqslant 2$ and $t \geqslant 1$, the capacity of the q-GP(t) constraint is defined by*

$$H_q(t) \stackrel{\text{def}}{=} \lim_{n \to \infty} \frac{\log_2 |\mathcal{B}_{q;t}(n)|}{n}. \quad (20)$$

Exactly the same argument that we used in the proof of Proposition 1 can be now used to show that the limit in (20) exists, and in fact

$$H_q(t) = \inf_{n \geqslant 1} \frac{\log_2 |\mathcal{B}_{q;t}(n)|}{n}. \quad (21)$$

Observe that, for all fixed q, the sequence $H_q(1), H_q(2), \ldots$ is a nonincreasing sequence of nonnegative numbers. This is so because $\mathcal{B}_{q;t+1}(n) \subseteq \mathcal{B}_{q;t}(n)$ for all $n \in \mathbb{Z}^+$ and all $t \in \mathbb{Z}^+$, as is evident from Definition 3. Therefore $\lim_{t \to \infty} H_q(t)$ exists, and equals $\inf_{t \geqslant 1} H_q(t)$. The following proposition shows that this limit is also equal to H_q, as defined in (16).

Proposition 2. *For all integers $q \geqslant 2$,*

$$H_q = \lim_{t \to \infty} H_q(t) = \inf_{t \geqslant 1} H_q(t). \quad (22)$$

Proof. Let $\alpha_q \stackrel{\text{def}}{=} \inf_{t \geqslant 1} H_q(t)$. We have already shown that $\lim_{t \to \infty} H_q(t) = \alpha_q$, so it remains to prove that $H_q = \alpha_q$. It follows immediately from Definition 1 and Definition 3 that

$\mathcal{B}_{q;t}(n) \supseteq \mathcal{B}_q(n)$ for all $n, t \in \mathbb{Z}^+$. Hence $|\mathcal{B}_{q;t}(n)| \geqslant |\mathcal{B}_q(n)|$ and $H_q(t) \geqslant H_q$ for all $t \in \mathbb{Z}^+$. Letting $t \to \infty$, we conclude that $\alpha_q \geqslant H_q$. For the reverse inequality, first fix an $n \in \mathbb{Z}^+$ and observe that $\mathcal{B}_q(n) = \mathcal{B}_{q;n}(n)$. Therefore

$$\frac{\log_2 |\mathcal{B}_q(n)|}{n} = \frac{\log_2 |\mathcal{B}_{q;n}(n)|}{n} \geqslant \inf_{m \geqslant 1} \frac{\log_2 |\mathcal{B}_{q;n}(m)|}{m}.$$

Note that the right-hand side above is precisely $H_q(n)$ in view of (21), and $H_q(n) \geqslant \alpha_q$ by the definition of α_q. If follows that $\log_2 |\mathcal{B}_q(n)|/n \geqslant \alpha_q$ for all $n \in \mathbb{Z}^+$, and therefore $H_q \geqslant \alpha_q$. This completes the proof of the proposition. ∎

Observe that our claim in (9) follows as a special case (for $q = 3$) from Proposition 2. Thus, as discussed in Section I-D, Proposition 2 provides a means of computing increasingly tight upper bounds on H_q. In particular, this proposition implies that $H_{\text{TGP}} = H_3$ can be determined by studying the asymptotics of the sequence $H_3(1), H_3(2), \ldots$. In Section IV-D, we show that there is indeed an algorithm that can be used to compute $H_3(t)$ for any given t. Unfortunately, this algorithm is too computationally intensive to be useful in practice.

III. THE BINARY GHOST-PULSE CONSTRAINTS

Following the terminology of Section I, we shall refer to the q-GP constraints with $q = 2$ as the *binary ghost-pulse* (BGP) constraints. Such constraints can be completely analyzed, and the purpose of this section is to present this analysis.

A. The BGP Constraint with Unbounded Memory

Note that Definitions 1 and 3 become somewhat redundant in the binary case. For a binary sequence $\boldsymbol{x} = (x_1 x_2 \ldots x_n)$, *any* $k, l, m \in \mathrm{supp}(\boldsymbol{x})$ satisfy $x_k = x_l = x_m$. Thus the BGP constraint is simply the requirement that for all $k, l, m \in \mathrm{supp}(\boldsymbol{x})$, either $k + l - m \in \mathrm{supp}(\boldsymbol{x})$ or $k + l - m \notin [n]$. The following theorem makes use of this observation to show that sequences that satisfy the BGP constraint are precisely those whose support set forms an arithmetic progression.

Theorem 3. *For all $n \in \mathbb{Z}^+$, a sequence $\boldsymbol{x} = (x_1 x_2 \ldots x_n) \in \mathcal{A}_2^n$ satisfies the BGP constraint iff there exist $a, d \in [0, n]$ such that*

$$\mathrm{supp}(\boldsymbol{x}) = (a + d\mathbb{Z}) \cap [n]. \quad (23)$$

Proof. ($\Leftarrow$) Suppose that $\boldsymbol{x}$ satisfies (23), and consider any $k_1, k_2, k_3 \in \mathrm{supp}(\boldsymbol{x})$. Then $k_i = a + d j_i$ for some $j_i \in \mathbb{Z}$. Set $j = j_1 + j_2 - j_3$. Then $k_1 + k_2 - k_3 = a + jd \in (a + d\mathbb{Z})$. Thus, either $k_1 + k_2 - k_3 \in \mathrm{supp}(\boldsymbol{x})$ or $k_1 + k_2 - k_3 \notin [n]$.

($\Rightarrow$) Suppose that $\boldsymbol{x} = (x_1 x_1 \ldots x_n)$ satisfies the BGP constraint. If $\mathrm{supp}(\boldsymbol{x}) = \varnothing$, then we can take $a = d = 0$ in (23). If $|\mathrm{supp}(\boldsymbol{x})| = 1$, then we can take a to be the unique integer in $\mathrm{supp}(\boldsymbol{x})$ and set $d = 0$. Hence, it remains to consider the case where $|\mathrm{supp}(\boldsymbol{x})| \geqslant 2$. For this case, set

$$d = \min\{ |k - m| \, : \, k, m \in \mathrm{supp}(\boldsymbol{x}), \; k \neq m \}, \quad (24)$$

and then take a to be any integer with $a, a + d \in \mathrm{supp}(\boldsymbol{x})$. To prove that $\boldsymbol{x}$ satisfies (23) with this choice of a and d, we will first show that $(a + d\mathbb{Z}) \cap [n] \subset \mathrm{supp}(\boldsymbol{x})$, and then prove that every element of $\mathrm{supp}(\boldsymbol{x})$ must also be in $a + d\mathbb{Z}$.

Claim 1: $(a + d\mathbb{Z}) \cap [n] \subset \operatorname{supp}(\boldsymbol{x})$. In order to establish this claim, suppose that

$$\{ a + \ell d \ : \ i \leqslant \ell \leqslant j \} \subset \operatorname{supp}(\boldsymbol{x}) \tag{25}$$

for some $i \leqslant 0$ and $j \geqslant 1$. By our choice of a and d, we know that (25) certainly holds for $i = 0$ and $j = 1$. Observe that

$$a + (i{-}1)d \ = \ (a + id) + (a + id) - (a + (i{+}1)d)$$

and

$$a + (j{+}1)d \ = \ (a + jd) + (a + jd) - (a + (j{-}1)d).$$

Hence, if $\boldsymbol{x}$ satisfies the BGP constraint, then $a + (i{-}1)d$ and $a + (j{+}1)d$ belong to $\operatorname{supp}(\boldsymbol{x})$, provided only that these positions are in $[n]$. In other words, we can grow the arithmetic progression on the left-hand side of (25) in both directions, as long as it fits inside $[n]$, and the claim follows.
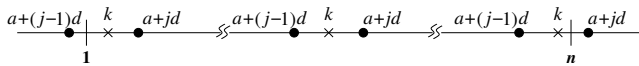

Fig. 2. Various possibilities for the choice of $k \in \operatorname{supp}(\boldsymbol{x})$ with $k \notin (a + d\mathbb{Z})$

Claim 2: $\operatorname{supp}(\boldsymbol{x}) \subset (a + d\mathbb{Z})$. Assume to the contrary that there is a $k \in \operatorname{supp}(\boldsymbol{x})$ with $k \notin (a + d\mathbb{Z})$. Then we must have

$$a + (j{-}1)d \ < \ k \ < \ a + jd \tag{26}$$

for some $j \in \mathbb{Z}$ such that at least one of $a + (j{-}1)d$ and $a + jd$ lies in $[n]$ (cf. Fig. 2). Without loss of generality (w.l.o.g.), suppose that $(a + jd) \in [n]$. Then $(a + jd) \in \operatorname{supp}(x)$ in view of Claim 1. But the difference between $a + jd$ and k is strictly less than d by (26), which contradicts the definition of d in (24).

By Claim 1 and Claim 2, we have $\operatorname{supp}(\boldsymbol{x}) = (a + d\mathbb{Z}) \cap [n]$, which completes the proof of the theorem. ∎

Corollary 4. *There are at most $(n + 1)^2$ sequences in $\mathcal{A}_2^n$ that satisfy the BGP constraint, and therefore*

$$H_{\text{BGP}} \ \overset{\text{def}}{=} \ \lim_{n \to \infty} \frac{\log_2 |\mathcal{B}_2(n)|}{n} \ \leqslant \ \lim_{n \to \infty} \frac{\log_2 (n+1)^2}{n} \ = \ 0.$$

Proof. There are $(n + 1)^2$ different ways of selecting the integers a and d from $[0, n]$. By Theorem 3, every sequence in $\mathcal{B}_2(n)$ is uniquely determined by one such choice. ∎

In fact, using Theorem 3 as a starting point, a more careful analysis of the possible choices for a and d shows that

$$|\mathcal{B}_2(n)| \ = \ \begin{cases} \frac{1}{4}(n+2)(n+2) & n \text{ even} \\ \frac{1}{4}(n+1)(n+3) & n \text{ odd.} \end{cases} \tag{27}$$

We leave the proof of this expression as a straightforward, but tedious, combinatorial exercise for the reader.

B. The BGP(t) Constraints

We next take on the analysis of the BGP(t) constraint, for arbitrary $t \in \mathbb{Z}^+$. We will show that the BGP(t) constraint is closely related to the well-known (t, ∞) constraint. A binary sequence $\boldsymbol{x}$ is said to *satisfy the (t, ∞) constraint* if there are at least t zeros between any two ones in $\boldsymbol{x}$. We use $\mathcal{S}_{t,\infty}(n)$

TABLE I
CAPACITY OF THE BGP(t) CONSTRAINT FOR $t = 1, 2, \ldots, 20$

t	$H_2(t)$	t	$H_2(t)$	t	$H_2(t)$	t	$H_2(t)$
1	0.6942	6	0.3282	11	0.2301	16	0.1813
2	0.5515	7	0.3011	12	0.2180	17	0.1742
3	0.4650	8	0.2788	13	0.2073	18	0.1678
4	0.4057	9	0.2600	14	0.1977	19	0.1618
5	0.3620	10	0.2440	15	0.1891	20	0.1564

to denote the set of all (t, ∞)-constrained binary sequences of length n. Such sequences have been extensively studied in the constrained coding literature [12]–[14], [18], [21]. The next theorem shows that the set $\mathcal{B}_{2;t}(n)$ of all sequences in $\mathcal{A}_2^n$ that satisfy the BGP(t) constraint is not much larger than $\mathcal{S}_{t,\infty}(n)$.

Theorem 5. *Let $\mathcal{Q}_t(n)$ denote the set of all sequences $\boldsymbol{x} \in \mathcal{A}_2^n$ such that $\operatorname{supp}(\boldsymbol{x}) = (a + d\mathbb{Z}) \cap [n]$ for some a and d in $[0, t]$. Then for all $n, t \in \mathbb{Z}^+$, we have*

$$\mathcal{B}_{2;t}(n) \ = \ \mathcal{S}_{t,\infty}(n) \cup \mathcal{Q}_t(n). \tag{28}$$

Proof. It is easy to see from (the proof of) Theorem 3 that $\mathcal{Q}_t(n) \subset \mathcal{B}_{2;t}(n)$. Note that if $\boldsymbol{x} \in \mathcal{S}_{t,\infty}(n)$, then (5) cannot be satisfied by any $k, l, m \in \operatorname{supp}(\boldsymbol{x})$. Hence, by Definition 3, all $\boldsymbol{x} \in \mathcal{S}_{t,\infty}(n)$ also belong to $\mathcal{B}_{2;t}(n)$. It follows that

$$\left(\mathcal{S}_{t,\infty}(n) \cup \mathcal{Q}_t(n) \right) \ \subseteq \ \mathcal{B}_{2;t}(n). \tag{29}$$

To establish the inclusion in the other direction, it would suffice to show that

$$\left(\mathcal{B}_{2;t}(n) \setminus \mathcal{S}_{t,\infty}(n) \right) \ \subseteq \ \mathcal{Q}_t(n). \tag{30}$$

Thus consider an $\boldsymbol{x} \in \left(\mathcal{B}_{2;t}(n) \setminus \mathcal{S}_{t,\infty}(n) \right)$. Since $\boldsymbol{x} \notin \mathcal{S}_{t,\infty}(n)$, there exist distinct $k, m \in \operatorname{supp}(\boldsymbol{x})$ with $|k - m| \leqslant t$. Define

$$d \ \overset{\text{def}}{=} \ \min \{ \ |k - m| \ : \ k, m \in \operatorname{supp}(\boldsymbol{x}), \ k \neq m \} \tag{31}$$

as in (24), and note that $1 \leqslant d \leqslant t$. As in Theorem 3, let a' be any integer with $a', a' + d \in \operatorname{supp}(\boldsymbol{x})$. Then exactly the same argument we used in the proof of Theorem 3 shows that

$$\operatorname{supp}(\boldsymbol{x}) = (a' + d\mathbb{Z}) \cap [n]. \tag{32}$$

Finally, set $a = a' \bmod d$. Since $d \leqslant t$ in (31), we obviously have $a \in [0, t{-}1]$. But $a' + d\mathbb{Z} = a + d\mathbb{Z}$, so (32) implies that $\operatorname{supp}(\boldsymbol{x}) = (a + d\mathbb{Z}) \cap [n]$. Thus $\boldsymbol{x} \in \mathcal{Q}_t(n)$, as desired. ∎

Let $C(t, \infty)$ denote the capacity of the (t, ∞) constraint, given by $C(t, \infty) = \lim_{n \to \infty} \log_2 |\mathcal{S}_{t,\infty}(n)| / n$. It is well known (see e.g. [12, p. 88]) that $C(t, \infty) = \log_2 \rho_t$, where ρ_t is the largest-magnitude root of the polynomial $z^{t+1} - z^t - 1$. It is also known that this root is always real, irrational [5], and lies in the open interval $(1, 2)$. Thus $0 < C(t, \infty) < 1$.

Corollary 6. *Let ρ_t denote the largest-magnitude root of the polynomial $z^{t+1} - z^t - 1$. Then for all $t \in \mathbb{Z}^+$, the capacity of the BGP(t) constraint is given by*

$$H_2(t) \ = \ C(t, \infty) \ = \ \log_2 \rho_t. \tag{33}$$

Proof. This follows immediately from Theorem 5. By (28), we have $|\mathcal{S}_{t,\infty}(n)| \leqslant |\mathcal{B}_{2;t}(n)| \leqslant |\mathcal{S}_{t,\infty}(n)| + |\mathcal{Q}_t(n)|$. Note

that $|\mathcal{Q}_t(n)| \leqslant (t+1)^2$, as there are $(t+1)^2$ different ways of choosing $a, d \in [0, t]$. The corollary now follows from (20). ∎

It is well known [12, p. 89] (and obvious) that ρ_t decreases as t increases. Moreover $\lim_{t \to \infty} \log_2 \rho_t = 0$, which by Lemma 2 provides an independent confirmation of Corollary 4.

For reference, we list in Table I the value of $H_2(t) = \log_2 \rho_t$, rounded to four decimal places, for all $t = 1, 2, \ldots, 20$. As can be seen from this table, $H_2(t)$ is less than 0.25 for all $t \geqslant 10$. This means that codes consisting of sequences that satisfy the BGP or the BGP(t) constraints are not particularly efficient means of mitigating the ghost-pulse problem.

C. Coding Into the BGP Constraints

Nevertheless, it may still be of interest to suggest methods for encoding an arbitrary binary sequence into a sequence satisfying the BGP or the BGP(t) constraints.

For the BGP constraint, Theorem 3 and (27) give a precise enumeration of all the sequences in $\mathcal{B}_2(n)$. Thus unconstrained binary data can be mapped into BGP-constrained sequences using an enumerative coding technique [8].

In principle, enumerative coding can be also used to code into the BGP(t) constraints. However, this requires precise enumeration of the sequences in $\mathcal{B}_{2;t}(n)$ for each $n \in \mathbb{Z}^+$. Unfortunately, Theorem 5 does not yield a simple formula for computing $|\mathcal{B}_{2;t}(n)|$ as a function of n and t. Thus, enumerative coding would be unnecessarily complex in this case.

We can code into the BGP(t) constraint with significantly lower complexity if we are willing to suffer a marginal loss in coding rate. When n is sufficiently large, we can ignore the contribution of $\mathcal{Q}_t(n)$ to $\mathcal{B}_{2;t}(n)$ for all practical purposes. Observe that when t is fixed, $|\mathcal{Q}_t(n)|$ is bounded by the constant $(t+1)^2$ while $|\mathcal{S}_{t,\infty}(n)|$ grows exponentially with n.
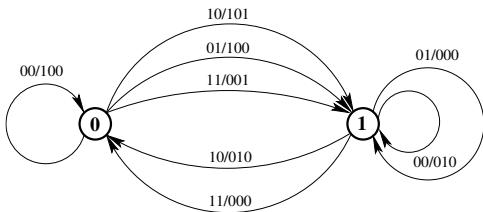

Fig. 3. A rate 2:3 sliding-block decodable encoder for the $(1, \infty)$ constraint

Coding into the (t, ∞) constraint is a very well-studied subject [13], [14], [18, Chapter 5], [21]. For all positive integers p and q with $p/q < C(t, \infty)$, there is a rate $p:q$ finite-state encoder for the (t, ∞) constraint, meaning a finite-state machine that generates an output block of q bits for every input block of p bits, and converts unconstrained binary sequences into sequences that satisfy the (t, ∞) constraint. For example, the graph in Fig. 3 is a rate 2:3 two-state encoder for the $(1, \infty)$ constraint. Such rate $p:q$ encoders can, in fact, be designed so that the constrained sequences they generate are amenable to decoding with a *sliding-block decoder* [21, Theorem 3.35]. For example, the encoder in Fig. 3 is indeed sliding-block decodable: a description of the corresponding sliding-block decoder can be derived from [18, Example 5.5.5].

It is well known [5] that the capacity $C(t, \infty)$ is *irrational* for all $t \geqslant 1$. Thus the design and the implementation of rate $p:q$ encoders necessarily becomes more cumbersome as the rate p/q approaches capacity. Consequently, in situations where variable-rate encoding and state-dependent decoding are acceptable, the constrained coding technique of [6], [17], known as "bit-stuffing," is an attractive alternative. The bit-stuffing encoder comprises two components. The first is an invertible distribution transformer that converts a sequence of i.i.d. equiprobable information bits into a sequence of i.i.d. biased bits, with the probability of a zero given by a prescribed value p. The second component inserts (stuffs) a string of t consecutive zeros following every one in this biased sequence. The decoder simply discards the string of t zeros that follows each one, and then applies the inverse of the distribution transformer. It can be shown [6] that, if the parameter p is optimized, the average rate of the bit-stuffing encoder equals the capacity $C(t, \infty)$.

IV. The Ternary Ghost-Pulse Constraints

It happens to be much harder to analyze the TGP and TGP(t) constraints than their binary counterparts BGP and BGP(t). Nevertheless, we will attempt to do so in this section.

A. The TGP Constraint with Unbounded Memory

In order to gain some understanding of the structure of finite-length TGP-constrained binary sequences, we extend the definition of the TGP constraint in a natural way to *bi-infinite sequences* — that is, sequences indexed by the set of integers $\mathbb{Z}$.

Definition 5. *A bi-infinite sequence* $\boldsymbol{x} = \{x_j\}_{j \in \mathbb{Z}}$ *over the ternary alphabet* $\mathcal{A}_3 = \{-1, 0, 1\}$ *is said to satisfy the TGP constraint if for all* $k, l, m \in \mathbb{Z}$ *such that* $x_k, x_l,$ *and* x_m *are equal and nonzero, we also have* $x_{k+l-m} \neq 0$.

Let $\mathcal{T}_3^*$ denote the set of all bi-infinite ternary sequences satisfying the TGP constraint, and let $\mathcal{B}_3^* = \xi(\mathcal{T}_3^*)$ denote the set of all *binary* bi-infinite sequences that can be converted to a sequence in $\mathcal{T}_3^*$ by changing some of their 1's to -1's. Using results from a branch of mathematics known as Ramsey theory [10], we have shown in [15] that any $\boldsymbol{y} \in \mathcal{B}_3^*$ is *almost periodic*: it differs from a periodic sequence in at most two positions. Based on this and other results, we conjecture that the capacity $H_{\text{TGP}} = H_3$ of the TGP constraint is zero.

TABLE II
VALUES OF $\mathcal{B}_3(n)$ FOR $n = 1, 2, \ldots, 32$

t	$|\mathcal{B}_3(n)|$	t	$|\mathcal{B}_3(n)|$	t	$|\mathcal{B}_3(n)|$	t	$|\mathcal{B}_3(n)|$
1	2	9	240	17	2591	25	11497
2	4	10	358	18	3245	26	13427
3	8	11	501	19	3977	27	15521
4	16	12	705	20	4881	28	17952
5	32	13	937	21	5850	29	20498
6	60	14	1248	22	7026	30	23449
7	100	15	1609	23	8313	31	26590
8	162	16	2078	24	9860	32	30193

In Table II, we list the number of sequences in $\mathcal{B}_3(n)$ for all $n = 1, 2, \ldots, 32$. All the values in Table II have been found
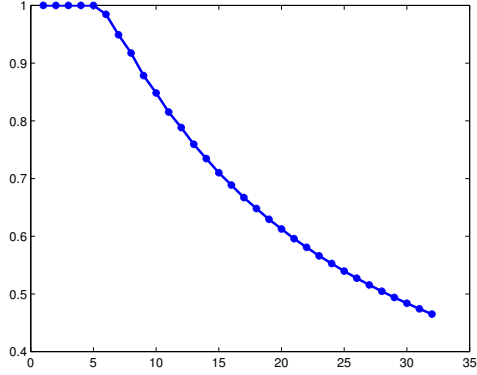
Fig. 4. Plot of $\log_2 |\mathcal{B}_3(n)|/n$ as a function of n, for $n = 1, 2, \ldots, 32$

by exhaustive computer search. We then used these values to plot $\log_2 |\mathcal{B}_3(n)|/n$ as a function of n in Fig. 4. As can be seen from this plot, the value of $\log_2 |\mathcal{B}_3(n)|/n$ decreases steadily as n increases, lending some further credence to our conjecture that $H_{\text{TGP}} = \lim_{n \to \infty} \log_2 |\mathcal{B}_3(n)|/n = 0$.

B. The TGP(1) Constraint

For the degenerate case $t = 1$, things remain simple. It is easy to show that the set $\mathcal{B}_{3;1}(n)$ of all the binary sequences that satisfy the TGP(1) constraint is, in fact, the entire space $\mathcal{A}_2^n$. This is based upon the following simple observation. A ternary sequence $(x_1 x_2 \ldots x_n)$ is in $\mathcal{T}_{3;1}(n)$ if and only if the following holds: for all $k \in [n]$ such that

$$x_k = x_{k+1} = +1 \quad \text{or} \quad x_k = x_{k+1} = -1 \quad (34)$$

we have $x_{k-1} \neq 0$ if $(k-1) \in [n]$ and $x_{k+2} \neq 0$ if $(k+2) \in [n]$. On the other hand, it is easy to allocate signs to any binary sequence in such a way that (34) never holds. In what follows, we will often use $+$ and $-$ to denote $+1$ and -1, respectively.
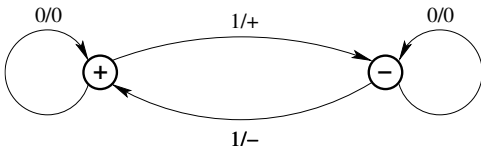

Fig. 5. Simple rate 1:1 two-state encoder for the TGP(1) constraint

Theorem 7. *For all $n \in \mathbb{Z}^+$, we have $\mathcal{B}_{3;1}(n) = \mathcal{A}_2^n$ and therefore the capacity of the TGP(1) constraint is $H_3(1) = 1$.*

Proof. Given any sequence $\boldsymbol{y} \in \mathcal{A}_2^n$, the following encoding rule converts $\boldsymbol{y}$ to a ternary sequence $\boldsymbol{x}$ satisfying the TGP(1) constraint: label the ones in $\boldsymbol{y}$ with alternating signs. More precisely, if we think of $\boldsymbol{y}$ as the input to the rate 1:1 encoder in Fig. 5, then $\boldsymbol{x}$ is the output of the encoder. To see that $\boldsymbol{x}$ indeed satisfies the TGP(1) constraint, note that the alternating signs rule guarantees that (34) never occurs. ∎

Observe that, in addition to its use in the proof of Theorem 7, the encoder of Fig. 5 gives a practical method by which an arbitrary finite-length binary sequence can be transformed into a ternary sequence satisfying the TGP(1) constraint.

C. The TGP(2) Constraint

For $t = 2$, things become much more interesting. Our main result for this case is the characterization of the set $\mathcal{B}_{3;2} = \xi(\mathcal{T}_{3;2})$ of all finite-length binary sequences that satisfy the TGP(2) constraint in terms of a small number of *forbidden blocks*.

To make this precise, let us first clarify our use of the term sub-block. We say that a sequence $(x'_1 x'_2 \ldots x'_m)$ is *a sub-block of the sequence* $(x_1 x_2 \ldots x_n)$ if there exists an $i \in [0, n-m]$ such that $(x'_1 x'_2 \ldots x'_m) = (x_{i+1} x_{i+2} \ldots x_{i+m})$. Now, let

$$\mathcal{F}(2) \stackrel{\text{def}}{=} \{(011100), (001110), (001111100)\} \quad (35)$$

and let $\mathcal{S}_{\mathcal{F}(2)}(n)$ be the set of all binary sequences of length n that do not contain any element of $\mathcal{F}(2)$ as a sub-block. Our main result in this subsection is the following theorem.

Theorem 8. *For all $n \in \mathbb{Z}^+$, we have*

$$\mathcal{B}_{3;2}(n) = \mathcal{S}_{\mathcal{F}(2)}(n). \quad (36)$$

We split the proof of (36) into two lemmas: one shows that $\mathcal{S}_{\mathcal{F}(2)}(n) \subseteq \mathcal{B}_{3;2}(n)$, the other establishes $\mathcal{B}_{3;2}(n) \subseteq \mathcal{S}_{\mathcal{F}(2)}(n)$. One of the two directions is easy, as the next lemma shows.

Lemma 9. *For all $n \in \mathbb{Z}^+$, we have*

$$\mathcal{B}_{3;2}(n) \subseteq \mathcal{S}_{\mathcal{F}(2)}(n). \quad (37)$$

Proof. We need to show that none of the sequences in $\mathcal{B}_{3;2}$ contains any of the three sequences in $\mathcal{F}(2)$ as a sub-block. Consider first the sequence $(011100) \in \mathcal{F}(2)$. The three ones in (011100) can be labeled in 2^3 different ways by $+/-$ to produce ternary sequences. However, noting that a ternary sequence $\boldsymbol{x}$ satisfies the TGP(2) constraint if and only if so does the sequence $-\boldsymbol{x}$, it is enough to consider the following four labelings of (011100):

$$\begin{array}{ll} (0+++00), & (0++-00) \\ (0+-+00), & (0+--00). \end{array} \quad (38)$$

It can be verified by direct inspection that none of the four sequences in (38) satisfies the TGP(2) constraint. Hence, none can be a sub-block of a sequence in $\mathcal{T}_{3;2}$, which implies that (011100) cannot be a sub-block of a sequence in $\mathcal{B}_{3;2}$. The other two forbidden blocks in $\mathcal{F}(2)$ can be disposed of in the same way. ∎

To establish inclusion in the opposite direction, we describe an encoding rule that takes an arbitrary sequence $\boldsymbol{y} \in \mathcal{S}_{\mathcal{F}(2)}(n)$ and assigns a $+/-$ labeling to the ones in $\boldsymbol{y}$ in such a way that the resulting ternary sequence satisfies the TGP(2) constraint. More precisely, we construct a function

$$\Psi : \bigcup_{n=1}^{\infty} \mathcal{A}_2^n \to \bigcup_{n=1}^{\infty} \mathcal{A}_3^n \quad (39)$$

such that $\xi(\Psi(\boldsymbol{y})) = \boldsymbol{y}$ for all $\boldsymbol{y}$ in the domain of Ψ and, furthermore, $\Psi(\boldsymbol{y}) \in \mathcal{T}_{3;2}(n)$ for all $\boldsymbol{y} \in \mathcal{S}_{\mathcal{F}(2)}(n)$. This function Ψ will be based upon the alternating signs idea of Theorem 7; however, a much more careful analysis is now required.

The first step in the construction of Ψ consists of decomposing a binary sequence $\boldsymbol{y}$ into its maximal runs. Henceforth, we use $\boldsymbol{0}^j$ and $\boldsymbol{1}^j$ to denote the all-zero and the all-one sequences

of length j, respectively. Any finite-length nonzero binary sequence $\boldsymbol{y}$ can be written uniquely in its maximal-run form:

$$\boldsymbol{y} = \left(\mathbf{0}^{a_0}\,\mathbf{1}^{b_1}\,\mathbf{0}^{a_1}\,\mathbf{1}^{b_2}\,\mathbf{0}^{a_2}\,\cdots\,\mathbf{0}^{a_{r-1}}\mathbf{1}^{b_r}\,\mathbf{0}^{a_r}\right) \quad (40)$$

for some $r \geqslant 1$, where $a_1, a_2, \ldots, a_{r-1}$ and $b_1, b_2, \ldots, b_r$ are positive integers while $a_0, a_r \geqslant 0$. Each of the r sub-blocks $\mathbf{1}^{b_i}$ of $\boldsymbol{y}$ is called a *maximal run of ones* in $\boldsymbol{y}$.

The next step is to convert maximal runs into sequences over the alphabet $\{+, -\}$. Specifically, we define the function $\psi : \bigcup_{j=1}^{\infty}\{\mathbf{1}^j\} \to \bigcup_{j=1}^{\infty}\{+, -\}^j$ as follows:

$$\psi(\mathbf{1}^1) = +, \quad \psi(\mathbf{1}^2) = +-, \quad \psi(\mathbf{1}^3) = +-+$$

$$\psi(\mathbf{1}^4) = +--+, \quad \psi(\mathbf{1}^5) = +-++- \quad (41)$$

$$\psi(\mathbf{1}^6) = +--++-, \quad \psi(\mathbf{1}^j) = +--\mathbf{1}^{j-6}--+$$

where the last expression above applies for all $j \geqslant 7$. Observe that $\psi(\mathbf{1}^j)$ is a sequence of length j, so that $\xi\left(\psi(\mathbf{1}^j)\right) = \mathbf{1}^j$. More importantly, $\psi(\mathbf{1}^j)$ satisfies the property described in the following lemma.

Lemma 10. *Let j be a positive integer other than 3 or 5, and let $\psi(\mathbf{1}^j) = (x_1 x_2 \ldots x_j)$. Then for all $k, l, m \in [j]$ such that*

$$x_k = x_l = x_m \quad \text{and} \quad \max\{|k-l|, |l-m|, |m-k|\} \leqslant 2 \quad (42)$$

we have $k + l - m \in [j]$ as well. Thus $x_{k+l-m} \neq 0$ and, more-over, if $(x_1 x_2 \ldots x_j)$ is a sub-block of a $TGP(2)$-constrained ternary sequence $\boldsymbol{x}$ of length n, then this sub-block does not impose constraints on any of the other $n - j$ positions in $\boldsymbol{x}$.

Proof. The fact that $k + l - m \in [j]$ whenever (42) is satis-fied follows by direct inspection from (41). ∎

Now let $\boldsymbol{y} \in \mathcal{A}_2^n$ be an arbitrary binary sequence of length n. If $\boldsymbol{y} = \mathbf{0}^n$ or $\boldsymbol{y} = \mathbf{1}^n$, we simply set $\Psi(\boldsymbol{y}) = \boldsymbol{y}$. Otherwise, we decompose $\boldsymbol{y}$ into its maximal runs as in (40), and set

$$\Psi(\boldsymbol{y}) = \left(\mathbf{0}^{a_0}\,\boldsymbol{x}_1\,\mathbf{0}^{a_1}\,\boldsymbol{x}_2\,\mathbf{0}^{a_2}\,\cdots\,\mathbf{0}^{a_{r-1}}\boldsymbol{x}_r\,\mathbf{0}^{a_r}\right) \quad (43)$$

where $\boldsymbol{x}_i \in \{+, -\}^{b_i}$ are defined by the following iterative procedure:

$$\boldsymbol{x}_1 = \begin{cases} ++- & \text{if } a_0 = 0 \text{ and } b_1 = 3 \\ \psi(\mathbf{1}^{b_1}) & \text{otherwise} \end{cases} \quad (44)$$

$$\boldsymbol{x}_i = \begin{cases} \psi(\mathbf{1}^{b_i}) & \text{if last symbol of } \boldsymbol{x}_{i-1} \text{ is } - \\ -\psi(\mathbf{1}^{b_i}) & \text{if last symbol of } \boldsymbol{x}_{i-1} \text{ is } + \end{cases} \quad (45)$$

for all $i = 2, 3, \ldots, r$, but with two exceptions. If $b_i = 5$ while $a_i \in \{0, 1\}$, we modify the expression for $\boldsymbol{x}_i$ as follows:

$$\boldsymbol{x}_i = \begin{cases} +--+- & \text{if last symbol of } \boldsymbol{x}_{i-1} \text{ is } - \\ -++-+ & \text{if last symbol of } \boldsymbol{x}_{i-1} \text{ is } + . \end{cases} \quad (46)$$

Finally, if $\boldsymbol{y}$ ends with 0111 (that is, if $a_r = 0$ and $b_r = 3$), then we also modify the expression for $\boldsymbol{x}_r$ as follows:

$$\boldsymbol{x}_r = \begin{cases} +-- & \text{if last symbol of } \boldsymbol{x}_{r-1} \text{ is } - \\ -++ & \text{if last symbol of } \boldsymbol{x}_{r-1} \text{ is } + . \end{cases} \quad (47)$$

Observe that (44)–(47) iteratively determine $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_r$ in such a way that the first symbol of $\boldsymbol{x}_i$ is always opposite in sign to the last symbol of $\boldsymbol{x}_{i-1}$, for all $i = 2, 3, \ldots, r$. This is

the appropriate generalization of the *alternating signs* rule of Theorem 7 for the case of the TGP(2) constraint.

Lemma 11. *The function Ψ defined by equations (40)–(47) has the following properties:*

    P1. *For all $\boldsymbol{y} \in \mathcal{A}_2^n$, we have $\xi\left(\Psi(\boldsymbol{y})\right) = \boldsymbol{y}$.*

    P2. *For all $\boldsymbol{y} \in \mathcal{S}_{\mathcal{F}(2)}(n)$, we have $\Psi(\boldsymbol{y}) \in \mathcal{T}_{3;2}(n)$.*

Proof. Property **P1** means that Ψ converts a given binary se-quence $\boldsymbol{y}$ to a ternary sequence solely by assigning $+/-$ labes to the ones in $\boldsymbol{y}$. This should be obvious from the fact that $\xi\left(\psi(\mathbf{1}^j)\right) = \mathbf{1}^j$ and our construction of Ψ in (43)–(47).

To establish property **P2**, consider an arbitrary $\boldsymbol{y} \in \mathcal{S}_{\mathcal{F}(2)}(n)$ and let $\boldsymbol{x} = (x_1 x_2 \ldots x_n)$ denote its image $\Psi(\boldsymbol{y})$ under Ψ. We need to show that $\boldsymbol{x}$ satisfies the TGP(2) constraint. Clearly, if $\boldsymbol{y} \in \{\mathbf{0}^n, \mathbf{1}^n\}$, then $\boldsymbol{x} = \boldsymbol{y}$ trivially satisfies the constraint. We therefore assume that $\boldsymbol{y} \notin \{\mathbf{0}^n, \mathbf{1}^n\}$, which implies that $\boldsymbol{x}$ is given by (43). Now, let $k, l, m \in \text{supp}(\boldsymbol{x})$ and suppose that

$$x_k = x_l = x_m \quad \text{and} \quad \max\{|k-l|, |l-m|, |m-k|\} \leqslant 2.$$

We will further assume w.l.o.g. that $k \leqslant l \leqslant m$. Clearly, either x_k and x_m come from the same sub-block $\boldsymbol{x}_i$ of $\boldsymbol{x}$ in (43), or they belong to distinct sub-blocks $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. This leads to two cases, which we consider next.

Case 1: x_k belongs to $\boldsymbol{x}_i$ while x_m belongs to $\boldsymbol{x}_j$, with $i \neq j$. Since distinct sub-blocks in (43) are separated by at least one zero, the only way that $|m - k| \leqslant 2$ can be satisfied is if x_k is the last symbol of $\boldsymbol{x}_i$ whereas x_m is the first symbol of $\boldsymbol{x}_{i+1}$. But then the alternating signs rule implemented in (44)–(47) guarantees that $x_k \neq x_m$. We have thus arrived at a contradiction. This implies that x_k and x_m (and, hence, also x_l) must belong to the same sub-block $\boldsymbol{x}_i$ of $\boldsymbol{x}$ in (43).

Case 2: x_k, x_l, x_m belong to the sub-block $\boldsymbol{x}_i$ of length j. First suppose that $j \notin \{3, 5\}$. Then (44)–(47) guarantee that $\boldsymbol{x}_i = \psi(\mathbf{1}^j)$ or $\boldsymbol{x}_i = -\psi(\mathbf{1}^j)$. For this case, Lemma 10 implies that x_{k+l-m}, x_{k+m-l}, and x_{l+m-k} also lie within $\boldsymbol{x}_i$. This, in turn, guarantees that they are all nonzero, which is in agreement with the TGP(2) constraint. We are thus left to deal with the situation where $j = 3$ or $j = 5$. This is precisely where the forbidden blocks in $\mathcal{F}(2)$ come into play.

Case 2.1: The sub-block $\boldsymbol{x}_i$ is of length $j = 5$.

The key point is that the binary sequence (001111100) never occurs as a sub-block of $\boldsymbol{y}$. Hence $\boldsymbol{x}_i$ never appears in the context $\cdots 00\,\boldsymbol{x}_i\,00\cdots$. Note that the only relevant context for the TGP(2) constraint consists of the two symbols immediately before $\boldsymbol{x}_i$ and the two symbols immediately after $\boldsymbol{x}_i$. The fact that (001111100) does not occur in $\boldsymbol{y}$ together with the encoding rules in (41)–(47) guarantee that $\boldsymbol{x}_i$ appears as follows in all of its possible contexts:

$$\begin{array}{ll} (+-++-00\cdots & (+-++-0+\cdots \\ (0+-++-00\cdots & (0+-++-0+\cdots \\ \cdots+0-+--+00\cdots & \cdots+0-+--+0-\cdots \\ \cdots-0+-++-00\cdots & \cdots-0+-++-0+\cdots \\ \cdots00+--+-0+\cdots & \cdots00-++-+0-\cdots \\ \cdots00+--+-) & \cdots00-++-+) \\ \cdots-0+--+-) & \cdots+0-++-+) \\ \cdots00+--+-0) & \cdots00-++-+0) \\ \cdots-0+--+-0) & \cdots+0-++-+0) \end{array} \quad (48)$$

where '(' and ')' signify the beginning and the end of the entire sequence $x = \Psi(y)$, respectively. It is now easy to verify by direct inspection that each of the 18 sequences in (48) satisfies the TGP(2) constraint.

Case 2.2: The sub-block x_i is of length $j = 3$. Similarly to the previous case, the fact that (011100) and (001110) do not occur in y together with the encoding rules in (41)–(47) guarantee that x_i appears as follows in all of its possible contexts:

$$
\begin{aligned}
&(++{-}00\cdots \quad (++{-}0+\cdots \quad (0+{-}+0{-}\cdots \\
&\cdots{-}0+{-}+0{-}\cdots \quad \cdots+0{-}+{-}0+\cdots \\
&\cdots00+{-}{-}) \quad \cdots+0{-}++) \quad \cdots{-}0+{-}+0) \\
&\cdots00{-}++) \quad \cdots{-}0+{-}{-}) \quad \cdots+0{-}+{-}0).
\end{aligned}
\tag{49}
$$

Again, it can be verified by direct inspection that each of the 11 sequences in (49) satisfies the TGP(2) constraint.

Since our analysis in Cases 1 and 2 is exhaustive, this establishes property **P2** and completes the proof of the lemma. ∎

Lemma 11 shows that every sequence $y \in \mathcal{S}_{\mathcal{F}(2)}(n)$ can be converted to a ternary sequence in $\mathcal{T}_{3;2}(n)$ by assigning $+/-$ labels to the ones in y. This implies that $\mathcal{S}_{\mathcal{F}(2)}(n) \subseteq \mathcal{B}_{3;2}(n)$, by the definition of $\mathcal{B}_{3;2}(n)$ in (19). Together with Lemma 9, this completes the proof of Theorem 8. The next corollary uses this result to determine the capacity of the TGP(2) constraint.

Corollary 12. *Let ρ denote the largest-magnitude root of the polynomial $z^{10} - 2z^9 + z^5 - z^4 + 2z^3 - z^2 - 2z + 1$. Then the capacity of the TGP(2) constraint is given by*

$$
H_3(2) = \log_2 \rho \approx 0.96048.
\tag{50}
$$

Proof. We will use the results of Wilf [25] and of Guibas and Odlyzko [11], which provide a much more efficient means to compute the capacity of a constraint from its set of forbidden blocks than the standard methods (briefly discussed at the end of this subsection). Let $g_0 = 1$, and for $n \in \mathbb{Z}^+$, define

$$
g_n \stackrel{\text{def}}{=} |\mathcal{B}_{3;2}(n)| = |\mathcal{S}_{\mathcal{F}(2)}(n)|.
$$

Further, define the generating function $G(z) = \sum_{n=0}^{\infty} g_n z^{-n}$. Using Theorem 1 of [11], we find that $G(z)$ is given by

$$
G(z) = \frac{z(z+1)(z^8 - z^7 + z^6 - z^5 + z^4 - z^2 + 2z - 1)}{z^{10} - 2z^9 + z^5 - z^4 + 2z^3 - z^2 - 2z + 1}.
$$

It can be easily verified (using, say, MATLAB or MATHEMATICA) that the largest-magnitude pole of $G(z)$ is the unique largest-magnitude root of its denominator polynomial. Moreover, this root ρ is real and simple. It now follows from the theory of generating functions due to Wilf [25, Chapter 5] that $g_n = \alpha \rho^n (1 + o(1))$ for some constant $\alpha > 0$. Consequently,

$$
H_3(2) = \lim_{n \to \infty} \frac{\log_2 |\mathcal{B}_{3;2}(n)|}{n} = \lim_{n \to \infty} \frac{\log_2 g_n}{n} = \log_2 \rho.
$$

Using the MATHEMATICA software package, we have found that $\rho \approx 1.94596$, and therefore $H_3(2) \approx 0.96048$. ∎

Observe that $H_3(t)$ is *much* larger than $H_2(t)$ for $t = 1, 2$, as can be seen by comparing Table I with Theorem 7 and Corollary 12. Furthermore, the drop from $H_3(1)$ to $H_3(2)$ is

significantly smaller than the drop from $H_2(1)$ to $H_2(2)$. As mentioned in Section I-D, if this trend continues for larger values of t, we can have reasonably efficient codes that, under the simplifying assumption of that section, mitigate the formation of ghost pulses in a typical optical communication scenario.

To conclude our discussion of the TGP(2) constraint, we comment upon the design of encoders for converting arbitrary binary sequences into TGP(2)-constrained ternary sequences. The function Ψ constructed in (41)–(47) provides an explicit method of transforming sequences in $\mathcal{S}_{\mathcal{F}(2)}(n) = \mathcal{B}_{3;2}(n)$ into sequences in $\mathcal{T}_{3;2}(n)$. However, this function does not work for *arbitrary* binary sequences: if $y \notin \mathcal{S}_{\mathcal{F}(2)}(n)$, then $\Psi(y)$ is not necessarily in $\mathcal{T}_{3;2}(n)$. Thus, we still need to design an encoder that converts an arbitrary (unconstrained) binary sequence to a sequence in the constrained system

$$
\mathcal{S}_{\mathcal{F}(2)} \stackrel{\text{def}}{=} \bigcup_{n=1}^{\infty} \mathcal{S}_{\mathcal{F}(2)}(n).
$$

The theory of constrained coding provides a standard way to design such encoders, which we briefly outline in what follows. Let $\mathcal{G}$ be a finite, labeled, directed graph. We say that $\mathcal{G}$ is a *presentation of a constrained system* $\mathcal{S}$ if $\mathcal{S}$ is the set of all sequences obtained by reading the labels of all finite paths in $\mathcal{G}$. A presentation $\mathcal{G}$ of $\mathcal{S}$ is *deterministic* if at each vertex of $\mathcal{G}$, the outgoing edges are labeled distinctly. Given a deterministic presentation of $\mathcal{S}$ along with integers p and q such that p/q is less than or equal to the capacity of $\mathcal{S}$, there is a systematic algorithm [21, Section 4] for designing a rate $p{:}q$ finite-state encoder for $\mathcal{S}$ along with a corresponding decoder. Thus, to construct a finite-state encoder for our constrained system $\mathcal{S}_{\mathcal{F}(2)} = \mathcal{B}_{3;2}$, all we need to do is provide a deterministic presentation for $\mathcal{S}_{\mathcal{F}(2)}$. From this, the desired encoder can be generated via the algorithm mentioned above.
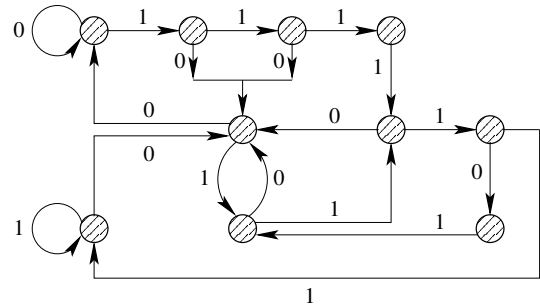

Fig. 6. A deterministic presentation of the constrained system $\mathcal{S}_{\mathcal{F}(2)} = \mathcal{B}_{3;2}$

It may be verified that the graph in Fig. 6 is a deterministic presentation of $\mathcal{S}_{\mathcal{F}(2)}$. Hence, it can be used as the starting point for the design of encoders that convert unconstrained binary sequences to sequences in $\mathcal{B}_{3;2}$ (and then, via the function Ψ in (41)–(47), to sequences in $\mathcal{T}_{3;2}$). In fact, the graph in Fig. 6 is the *minimal* deterministic presentation (also known as the *Shannon cover*) of $\mathcal{B}_{3;2}$, in the sense that it has the least number of vertices among all deterministic presentations of $\mathcal{B}_{3;2}$.

While on the subject of deterministic presentations, let us state the following well-known fact [21, Theorem 3.12], which will be needed in the next subsection. If $\mathcal{G}$ is a deterministic presentation of a given constrained system $\mathcal{S}$, then the capacity

of $\mathcal{S}$ is $\log_2 \lambda\big(A_\mathcal{G}\big)$, where $\lambda\big(A_\mathcal{G}\big)$ is the largest eigenvalue of the adjacency matrix of $\mathcal{G}$. Incidentally, this provides an alternative proof of Corollary 12, since the characteristic polynomial of the adjacency matrix of the graph in Fig. 6 is precisely $z^{10} - 2z^9 + z^5 - z^4 + 2z^3 - z^2 - 2z + 1$.

D. The TGP(t) Constraints for $t \geqslant 3$

It is clear that the painstaking analysis presented in the previous subsection cannot be easily extended to the TGP(t) constraint for an arbitrary $t \in \mathbb{Z}^+$. Instead, we suggest an alternative, systematic approach to tackle the general case, which can, in principle, be programmed into a computer.

The approach developed in this section has two main disadvantages. First, instead of computing $H_3(t)$ we end up with a slightly different quantity

$$H_3'(t) \stackrel{\text{def}}{=} \lim_{n \to \infty} \frac{\log_2 |\mathcal{B}_{3;t}'(n)|}{n} \qquad (51)$$

where $\mathcal{B}_{3;t}'(n)$ is the set of all binary sequences of length n that can be *extended to a bi-infinite sequence* without violating the TGP(t) constraint (more precise definition to follow shortly). This is not much of a problem, since $H_3'(t) \leqslant H_3(t)$ for all t and there are good reasons to believe that $H_3'(t) = H_3(t)$ for all t (see the remark below). The second problem is the computational complexity of the proposed approach. Unfortunately, this complexity is doubly-exponential in t. In fact, in order to compute $H_3'(t)$ one needs to construct a graph with at least $2^{\Omega(9^t)}$ vertices. Thus the proposed approach is not practical even for $t = 2$. Nevertheless, we believe that this approach has conceptual value, and sheds additional light on the underlying structure of the TGP(t) constraint.

The general idea behind our approach is to develop a procedure that, given a $t \in \mathbb{Z}^+$, generates a deterministic presentation $\mathcal{H}_{3;t}$ of the constrained system

$$\mathcal{B}_{3;t}' \stackrel{\text{def}}{=} \bigcup_{n=1}^{\infty} \mathcal{B}_{3;t}'(n). \qquad (52)$$

In developing our results, it would be much more convenient to deal with bi-infinite sequences. This eliminates the "edge effects" present at the beginning and end of a finite sequence, which could be quite bothersome (for example, much of the effort in describing the encoding rule Ψ of the previous subsection — see (44), (46), (47) — was devoted to such edge effects).

Recall that $\mathcal{T}_3^*$ was defined in Section IV-A as the set of bi-infinite ternary sequences satisfying the TGP constraint. We extend this definition in the natural way to the TGP(t) constraint.

Definition 6. *A bi-infinite sequence $\boldsymbol{x} = \{x_j\}_{j \in \mathbb{Z}}$ over the ternary alphabet $\mathcal{A}_3 = \{-1, 0, 1\}$ is said to satisfy the TGP(t) constraint if for all $k, l, m \in \mathbb{Z}$ such that*

$$\max\{|k-l|, |l-m|, |m-k|\} \leqslant t,$$

whenever x_k, x_l, x_m are equal and nonzero, then x_{k+l-m} is also nonzero. We let $\mathcal{T}_{3;t}^*$ denote the set of all bi-infinite ternary sequences satisfying the TGP(t) constraint, and let $\mathcal{B}_{3;t}^* = \xi\big(\mathcal{T}_{3;t}^*\big)$ denote the set of all bi-infinite binary sequences that can be converted to a sequence in $\mathcal{T}_{3;t}^*$ by negating some of their ones.

We now construct a deterministic presentation for $\mathcal{T}_{3;t}^*$. Given a $t \in \mathbb{Z}^+$, define a finite, labeled, directed graph $\mathcal{G}_{3;t}$, as follows. The set of vertices of $\mathcal{G}_{3;t}$ is the set of all

$$\boldsymbol{x} = \big(x_{-t}x_{-t+1}\ldots x_{-1}x_0 x_1 \ldots x_{2t-1}x_{2t}\big) \in \mathcal{A}_3^{3t+1}$$

that satisfy the following condition: for all $k, l, m \in [0, t]$ such that x_k, x_l, x_m are equal and nonzero, we also have $x_{k+l-m} \neq 0$. Note that the position indices k, l, m are restricted to the interval $[0, t]$ in the above condition. This implies that $\mathcal{G}_{3;t}$ has at least 3^{2t} vertices; for example all the sequences of the form

$$\big(x_{-t}x_{-t+1}\ldots x_{-1} \, 0 \, 0 \ldots 0 \, x_{t+1}x_{t+2}\ldots x_{2t}\big)$$

are vertices of $\mathcal{G}_{3;t}$. In fact, the order (number of vertices) of $\mathcal{G}_{3;t}$ is probably closer to 3^{3t} than to 3^{2t} (however, when t is small, the vertices of $\mathcal{G}_{3;t}$ can still be enumerated by exhaustive computer search). The edges of $\mathcal{G}_{3;t}$ are defined as follows. For each pair of vertices

$$\boldsymbol{x} = \big(x_{-t}x_{-t+1}\ldots x_{2t}\big) \quad \text{and} \quad \boldsymbol{x}' = \big(x_{-t}'x_{-t+1}'\ldots x_{2t}'\big)$$

where $\boldsymbol{x}$ and $\boldsymbol{x}'$ are not necessarily distinct, we draw a single directed edge from $\boldsymbol{x}$ to $\boldsymbol{x}'$ if and only if the last $3t$ symbols of $\boldsymbol{x}$ are equal to the first $3t$ symbols of $\boldsymbol{x}'$, that is if

$$\big(x_{-t+1}x_{-t+2}\ldots x_{2t}\big) = \big(x_{-t}'x_{-t+1}'\ldots x_{2t-1}'\big).$$

The label of this directed edge is the symbol x_{2t}'. This completes our construction of the graph $\mathcal{G}_{3;t}$.

Given a finite, labeled, directed graph $\mathcal{G}$, the *sofic shift of* $\mathcal{G}$ is the set of all bi-infinite sequences obtained by reading the labels of *bi-infinite* paths in $\mathcal{G}$. One of our main results in this subsection is the following theorem.

Theorem 13. *Let $\mathcal{X}_{3;t}$ denote the sofic shift of the graph $\mathcal{G}_{3;t}$. Then, for all $t \in \mathbb{Z}^+$, we have*

$$\mathcal{X}_{3;t} = \mathcal{T}_{3;t}^*. \qquad (53)$$

Proof. We first show that $\mathcal{T}_{3;t}^* \subseteq \mathcal{X}_{3;t}$. Consider any element $\boldsymbol{x} = \{x_j\}_{j \in \mathbb{Z}}$ of $\mathcal{T}_{3;t}^*$. For all $j \in \mathbb{Z}$, let $\boldsymbol{x}_j$ denote the sub-block $(x_{j-t}x_{j-t+1}\ldots x_{j+2t})$ of $\boldsymbol{x}$. Since $\boldsymbol{x}$ satisfies the TGP(t) constraint, it follows from our construction of $\mathcal{G}_{3;t}$ that $\boldsymbol{x}_j$ is a vertex of $\mathcal{G}_{3;t}$ for all $j \in \mathbb{Z}$. Moreover, since the last $3t$ symbols of $\boldsymbol{x}_{j-1}$ are obviously equal to the first $3t$ symbols of $\boldsymbol{x}_j$, the graph $\mathcal{G}_{3;t}$ has a unique edge e_j from $\boldsymbol{x}_{j-1}$ to $\boldsymbol{x}_j$, which is labeled by x_{j+2t}. But then, the sequence of such edges $\{e_j\}_{j \in \mathbb{Z}}$ is a path in $\mathcal{G}_{3;t}$ that generates $\boldsymbol{x}$. It follows that $\boldsymbol{x} \in \mathcal{X}_{3;t}$.

In order to establish the inclusion $\mathcal{X}_{3;t} \subseteq \mathcal{T}_{3;t}^*$, consider any element $\boldsymbol{x} = \{x_j\}_{j \in \mathbb{Z}}$ of $\mathcal{X}_{3;t}$ and let $\{e_j\}_{j \in \mathbb{Z}}$ denote the path in $\mathcal{G}_{3;t}$ that generates $\boldsymbol{x}$. We again let $\boldsymbol{x}_j$ denote the sub-block $(x_{j-t}x_{j-t+1}\ldots x_{j+2t})$ of $\boldsymbol{x}$. Then, it follows from our construction of $\mathcal{G}_{3;t}$ that for all $j \in \mathbb{Z}$, the vertex at which e_{j+2t} terminates must be the sequence $\boldsymbol{x}_j$. Therefore, $\boldsymbol{x}_j$ is a vertex in $\mathcal{G}_{3;t}$ for all $j \in \mathbb{Z}$. Now, suppose we have $k, l, m \in \operatorname{supp}(\boldsymbol{x})$ such that $\max\{|k-l|, |l-m|, |m-k|\} \leqslant t$ and $x_k = x_l = x_m$. In order to prove that $\boldsymbol{x} \in \mathcal{T}_{3;t}^*$, we must show that $x_{k+l-m} \neq 0$. Since k, l, m are all within a distance of t of each other, there exists a $j \in \mathbb{Z}$ such that $k, l, m \in [j, j+t]$. Observe that for any $k, l, m \in [j, j+t]$, the integer $k + l - m$ lies in $[j - t, j + 2t]$.

But now, since x_j is a vertex of $\mathcal{G}_{3;t}$, it follows from our definition of the vertex set of $\mathcal{G}_{3;t}$ that $x_{k+l-m} \neq 0$. Thus $x \in \mathcal{T}_{3;t}^*$, which shows that $\mathcal{X}_{3;t} \subseteq \mathcal{T}_{3;t}^*$ and completes the proof. $\blacksquare$

We now define $\mathcal{T}_{3;t}'$ as the set of all finite-length sequences that are sub-blocks of some sequence in $\mathcal{T}_{3;t}^*$. Stated another way, $\mathcal{T}_{3;t}'$ is a subset of the set $\mathcal{T}_{3;t}$ defined in Section II, consisting of all finite-length sequences that a) satisfy the TGP(t) constraint and b) can be extended to a bi-infinite sequence that satisfies the TGP(t) constraint. It is possible that some finite-length sequences in $\mathcal{T}_{3;t}$ *cannot* be extended in this way, in which case $\mathcal{T}_{3;t}'$ is strictly smaller than $\mathcal{T}_{3;t}$.

Corollary 14. *Let $X_{3;t}$ denote the constrained system of the graph $\mathcal{G}_{3;t}$. Then, for all $t \in \mathbb{Z}^+$, we have*

$$X_{3;t} = \mathcal{T}_{3;t}'. \tag{54}$$

Moreover, the graph $\mathcal{G}_{3;t}$ is a deterministic presentation of its constrained system $X_{3;t} = \mathcal{T}_{3;t}'$.

Proof. It should be obvious from our construction of $\mathcal{G}_{3;t}$ that outgoing edges at each vertex of $\mathcal{G}_{3;t}$ are labeled distinctly. Hence $\mathcal{G}_{3;t}$ is a deterministic presentation of its constrained system. Furthermore, it is well known (and obvious) that (53) implies (54). In the terminology of symbolic dynamics, the sets $X_{3;t}$ and $\mathcal{T}_{3;t}'$ are precisely the *languages* of the sofic shifts $\mathcal{X}_{3;t}$ and $\mathcal{T}_{3;t}^*$. Since the shifts are equal (by Theorem 13), their languages must be also equal. $\blacksquare$

Corollary 14 implies that we can find the capacity of $\mathcal{T}_{3;t}'$ from the largest eigenvalue of the adjacency matrix of $\mathcal{G}_{3;t}$. However, we are not interested in $\mathcal{T}_{3;t}'$, but rather in the set

$$\mathcal{B}_{3;t}' = \xi(\mathcal{T}_{3;t}') = \{ \xi(\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{T}_{3;t}' \}. \tag{55}$$

Letting $\mathcal{B}_{3;t}'(n)$ denote the number of sequences of length n in $\mathcal{B}_{3;t}'$, we get the expression (51) for the capacity $H_3'(t)$.

Remark. Here is a heuristic argument in support of our claim that $H_3'(t)$ is likely to be equal to $H_3(t)$. The difference between $H_3'(t)$ and $H_3(t)$ stems from the difference between the sets $\mathcal{T}_{3;t}'$ and $\mathcal{T}_{3;t}$. It is well known [13], [18], [21] that the capacity of a language is equal to the entropy of the underlying shift. Thus, instead of looking at $\mathcal{T}_{3;t}'$, we might as well look at the underlying sofic shift $\mathcal{X}_{3;t} = \mathcal{T}_{3;t}^*$. The TGP$(t)$ constraint defining $\mathcal{T}_{3;t}$ is a finite restriction of the TGP(t) constraint defining $\mathcal{T}_{3;t}^*$. Furthermore, the TGP(t) constraint is *local*, in the sense that it is defined through a finite window of length t.

Now, it is generally observed in the literature [18] that if a constrained system $\mathcal{S}$ is obtained via a finite restriction of a local constraint that defines a sofic shift $\mathcal{X}$, then the capacity of $\mathcal{S}$ equals the entropy of $\mathcal{X}$. Of course, this is clearly true whenever any finite sequence in $\mathcal{S}$ can be extended to a bi-infinite sequence in $\mathcal{X}$. However, "edge effects" sometimes make it impossible to extend certain sequences in $\mathcal{S}$ without violating the constraint. But, in the case of a local constraint, these edge effects are usually not strong enough to affect a significant proportion of the sequences in $\mathcal{S}$, so that the capacity of $\mathcal{S}$ is still equal to the entropy of $\mathcal{X}$. This is not always true, but the exceptions to this rule tend to be pathological.

It may be possible to prove rigorously that $H_3'(t) = H_3(t)$, but such a proof would have to deal in detail with the "edge effects" and is likely to be too tedious to be worth the effort.

The remaining problem is to construct a deterministic presentation for the set $\mathcal{B}_{3;t}'$ in (52) and (55). Given the graph $\mathcal{G}_{3;t}$, constructing a presentation for $\mathcal{B}_{3;t}'$ is easy: simply apply $\xi(\cdot)$ to all the labels in $\mathcal{G}_{3;t}$. Specifically, let $\mathcal{G}_{3;t}'$ denote the graph obtained from $\mathcal{G}_{3;t}$ by replacing the labels of all the edges with their absolute values. Then it is obvious from (55) and Corollary 14 that the graph $\mathcal{G}_{3;t}'$ is a presentation for $\mathcal{B}_{3;t}'$.

Note, however, that although $\mathcal{G}_{3;t}$ is a deterministic presentation of $\mathcal{T}_{3;t}'$, the graph $\mathcal{G}_{3;t}'$ is not necessarily a *deterministic* presentation of $\mathcal{B}_{3;t}'$. Indeed, there may be two edges emanating from the same vertex $\boldsymbol{x}$ in $\mathcal{G}_{3;t}$, one labeled with $+$ and the other with $-$, whose labels in $\mathcal{G}_{3;t}'$ would both be 1. Fortunately, there is a well-known procedure that, given an arbitrary presentation of a constrained system, constructs a deterministic presentation for it. This procedure is called the *subset construction method*; it is described in detail in [21, Section 2.2.1] and in [18, Theorem 3.3.2]. Applying the subset construction method to the graph $\mathcal{G}_{3;t}'$, we finally obtain a deterministic presentation $\mathcal{H}_{3;t}$ for the set $\mathcal{B}_{3;t}'$. Given this presentation, we can compute the capacity $H_3'(t)$ and construct encoders into $\mathcal{B}_{3;t}'$, as described in the previous subsection.

We can now summarize the entire procedure for computing the capacity $H_3'(t)$, as follows:

1 Construct the graphs $\mathcal{G}_{3;t}$ and $\mathcal{G}_{3;t}'$ as described above, and let $\mathcal{B}_{3;t}'$ be the constrained system presented by $\mathcal{G}_{3;t}'$.

2 Apply the subset construction method to $\mathcal{G}_{3;t}'$ in order to obtain a deterministic presentation $\mathcal{H}_{3;t}$ for $\mathcal{B}_{3;t}'$.

3 Construct the adjacency matrix $A_{3;t}$ of $\mathcal{H}_{3;t}$, and compute its largest eigenvalue $\lambda = \lambda(A_{3;t})$. Set $H_3'(t) = \log_2 \lambda$.

Of course, in theory, $\mathcal{H}_{3;t}$ can also be used to construct finite-state encoders for converting unconstrained binary sequences to sequences in $\mathcal{B}_{3;t}' \subset \mathcal{B}_{3;t}$, as explained in Section IV-C. In turn, the graphs $\mathcal{G}_{3;t}$ and $\mathcal{G}_{3;t}'$ provide a method for transforming a binary sequence $\boldsymbol{y} \in \mathcal{B}_{3;t}'$ into a ternary sequence $\boldsymbol{x}$, with $\xi(\boldsymbol{x}) = \boldsymbol{y}$, that satisfies the TGP$(t)$ constraint. For each given $\boldsymbol{y} \in \mathcal{B}_{3;t}'$, there is a path in $\mathcal{G}_{3;t}'$ whose label sequence is $\boldsymbol{y}$. We may then take $\boldsymbol{x}$ to be the sequence of labels along the *same path* in $\mathcal{G}_{3;t}$. The practicality of this method depends on the existence of a systematic procedure for finding a path in $\mathcal{G}_{3;t}'$ that generates $\boldsymbol{y}$. Of course, it also depends on the order of the graphs $\mathcal{H}_{3;t}$, $\mathcal{G}_{3;t}'$, and $\mathcal{G}_{3;t}$.

We have already observed that the order of $\mathcal{G}_{3;t}$ and $\mathcal{G}_{3;t}'$ is exponential in t. However, since we are interested primarily in small values of t, such exponential growth could still be tolerated. The main computational problem is with the subset construction method at Step 2 above. The subset construction technique, when applied to a graph with n vertices, produces a graph with $2^n - 1$ vertices. As a result, the graph $\mathcal{H}_{3;t}$ constructed in Step 2 has at least 2^{9^t} vertices. In fact, this is likely to be a vast underestimate of the order of $\mathcal{H}_{3;t}$.

V. SUMMARY

We have defined and analyzed a number of "ghost-pulse" constraints that can be used to design coding schemes which mitigate the formation of ghost pulses in the optical fiber channel. We show that coding schemes based upon sequences that satisfy the *binary* ghost-pulse (BGP) constraint must necessarily have poor rates, since the capacity of this constraint is zero. Sequences satisfying a more relaxed constraint, which we call the BGP(t) constraint, are more suitable for use as codes; however, the rate of such codes is still too low for practical applications. A more promising approach is to use the phase-modulation idea, which leads to *ternary* constraints. Thus we study the ternary ghost-pulse (TGP) and TGP(t) constraints. We leave the analysis of the TGP constraint with unbounded memory as an open problem, conjecturing that it has zero capacity. But we do provide a detailed analysis of the TGP(1) and TGP(2) constraints. Our analysis suggests that coding schemes using TGP(t)-constrained sequences can achieve much higher rates than those using BGP(t)-constrained sequences. We are therefore led to believe that TGP(t) constraints yield reasonably efficient schemes for mitigating the ghost-pulse problem. We also discuss the design of encoders and decoders for coding schemes involving the BGP, the BGP(t), and the TGP(t) constraints. While the procedures we suggest for coding into the BGP(t), TGP(1), and TGP(2) constraints can be implemented in practice, the corresponding design procedure for the general TGP(t) constraint with $t \geqslant 3$ is too computationally intensive to be implementable in its present form.

ACKNOWLEDGMENT

REFERENCES

[1] M.J. Ablowitz and T. Hirooka, "Intrachannel pulse interactions in dispersion-managed transmission systems: energy transfer," *Opt. Lett.*, vol. 27, no. 3, pp. 203–205, Feb. 2002.

[2] M.J. Ablowitz and T. Hirooka, "Resonant intrachannel pulse interactions in dispersion-managed transmission systems," *IEEE J. Sel. Topics Quant. Electron.*, vol. 8, no. 3, pp. 603–615, May/June 2002.

[3] N. Alic, personal communication.

[4] N. Alic and Y. Fainman, "Data dependent phase coding for suppression of ghost pulses in optical fibers," *IEEE Photon. Technol. Lett.*, vol. 16, no. 4, pp. 1212–1214, Apr. 2004.

[5] J.J. Ashley and P.H. Siegel, "A note on the Shannon capacity of run-length-limited codes," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 601–605, July 1987. (See also J.J. Ashley, M. Hilden, P. Perry, and P.H. Siegel, "Correction to 'A note on the Shannon capacity of runlength-limited codes'," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1110–1112, May 1993)

[6] P.E. Bender and J.K. Wolf, "A universal algorithm for generating optimal and nearly optimal run-length-constrained, charge-constrained binary sequences," in *Proc. 1993 IEEE Int. Symp. Inform. Theory,* San Antonio, Texas, Jan. 17-22, 1993, p. 6.

[7] K.S. Cheng and J. Conradi, "Reduction of pulse-to-pulse interaction using alternative RZ formats in 40-GB/s systems," *IEEE Photon. Technol. Lett.*, vol. 14, no. 1, pp. 98–100, Jan. 2002.

[8] T.M. Cover, "Enumerative source encoding," *IEEE Trans. Inform. Theory*, vol. IT-19, no. 1, pp. 73–77, Jan. 1973.

[9] R.-J. Essiambre, B. Mikkelsen, and G. Raybon, "Intra-channel cross-phase modulation and four-wave mixing in high-speed TDM systems," *Electron. Lett.*, vol. 35, no. 18, pp. 1576–1578, Sept. 1999.

[10] R.L. Graham, B.L. Rothschild, and J.H. Spencer, *Ramsey Theory*, John Wiley & Sons, New York, NY, 1980.

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

[12] K.A.S. Immink, *Coding Techniques for Digital Recorders*, Prentice Hall, New York, 1991.

[13] K.A.S. Immink, *Codes for Mass Data Storage Systems*, Shannon Foundation Publishers, The Netherlands, 1999.

[14] K.A.S. Immink, P.H. Siegel, and J.K. Wolf, "Codes for digital recorders," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2260–2299, Oct. 1998.

[15] N. Kashyap, P.H. Siegel, and A. Vardy, "An application of Ramsey theory to coding for the optical channel," *SIAM J. Discrete Math*, to appear; preprint available on request.

[16] S. Kumar, "Intrachannel four-wave mixing in dispersion managed RZ systems," *IEEE Photon. Technol. Lett.*, vol. 13, pp. 800–802, Aug. 2001.

[17] P. Lee, *Combined error-correcting/modulation recording codes*, Ph.D. Dissertation, University of California, San Diego, 1988.

[18] D. Lind and B.H. Marcus, *An Introduction to Symbolic Dynamics and Coding*, Cambridge Univ. Press, Cambridge, UK, 1995.

[19] X. Liu, X. Wei, A.H. Gnauck, C. Xu, and L.K. Wickham, "Suppression of intrachannel four-wave-mixing induced ghost pulses in high-speed transmissions by phase inversion between adjacent marker blocks," *Opt. Lett.*, vol. 27, no. 13, pp. 1177–1179, July 2002.

[20] P. V. Mamyshev and N. A. Mamysheva, "Pulse-overlapped dispersion-managed data transmission and intrachannel four-wave mixing," *Opt. Lett.*, vol. 24, no. 21, pp. 1454–1456, Nov. 1999.

[21] B.H. Marcus, R.M. Roth, and P.H. Siegel, "Constrained systems and coding for recording channels," in *Handbook of Coding Theory*, V. Pless and W.C. Huffman (Eds.), Elsevier, Amsterdam, 1998.

[22] J. Moon and B. Brickner, "Maximum transition run codes for data storage systems," *IEEE Trans. Magn.*, vol. 32, no. 5, pp. 3992–3994, Sept. 1996.

[23] I. Shake, H. Takara, K. Mori, S. Kawanishi, and Y. Yamabayashi, "Influence of inter-bit four-wave mixing in optical TDM transmission," *Electron. Lett.*, vol. 34, no. 16, pp. 1600–1601, Aug. 1998.

[24] B. Vasic, V.S. Rao, I.B. Djordjevic, R.K. Kostuk and I. Gabitov, "Ghost-pulse reduction in 40-Gb/s systems using line coding," *IEEE Photon. Technol. Lett.*, vol. 16, no1. 7, pp. 1784–1786, July 2004.

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

[26] J. Zweck and C.R. Menyuk, "Analysis of four-wave mixing between pulses in high-data-rate quasi-linear subchannel-multiplexed systems," *Opt. Lett.*, vol. 27, no. 14, pp. 1235–1237, July 2002.