

Higher Bit Rates for Dispersion-Managed Soliton Communication Systems via Constrained Coding

Vladimir Pechenkin and Frank R. Kschischang, *Fellow, IEEE*

Abstract—Constrained coding as a method to increase the data rate in dispersion-managed soliton (DMS) communication systems is proposed. This approach is well known and widely used in the context of magnetic and optical recording systems. This paper shows that it is also applicable to DMS systems due to certain similarities between the underlying physical channels. Since timing jitter is an important error-generating mechanism for solitons, a coding scheme specifically designed to combat pulse shifts is also presented, and its properties in the framework of a particular information-theoretic channel model are analyzed. A connection between the model used and the real physical channel is then established. Next, the coded system is compared with the original one from the channel capacity point of view with the help of numerical examples. Finally, the fact that the application of constrained coding may alleviate soliton pulse-to-pulse interaction is exploited. This, in turn, opens the door to the usage of higher-than-usual map strengths and ultimately leads to a significant increase of up to 50% in the bit rate.

Index Terms—Constrained coding, dispersion-managed solitons, error control, fiber-optic communications.

I. INTRODUCTION

CONSTRAINED coding is an approach originally used in magnetic and optical recording systems to increase data storage density [1]. As the name implies, the key idea of this technique is to impose certain constraints on transmitted sequences in order to avoid particular channel effects or to achieve some other specific goals. For example, to avoid intersymbol interference, one or more “0”s may be inserted between pairs of “1”s, or, to facilitate clock synchronization, long strings of “0”s may be forbidden.

The connection between soliton communication systems and constrained coding is established through soliton pulse-to-pulse interaction, which can be regarded as a form of intersymbol interference. It imposes a limitation on how tightly the pulses can be packed, i.e., a certain minimum initial separation Δ between two solitons at the input is required. The main idea of this paper is based on the observation that the bit slot duration T_B does not need to coincide with Δ (as it is usually chosen) but can be less. The introduction of constraints guarantees that the input pulses are separated by at least Δ , and intersymbol interference is as benign as in the original system. As we will show, flexible T_B in conjunction with constrained coding

alleviates the problem of soliton interaction and allows one to use stronger dispersion management.

The paper is organized as follows. Section II describes a general system model and specifies our main assumptions. In Section III, we present a new (constrained) coding scheme and investigate its properties when a certain channel model is applied. Section IV introduces a few key parameters of the conventional system and explains the relation between the original channel and the one used in Section III. Section V contains numerical results. Section VI concludes the paper.

II. SYSTEM MODEL

Accurate design of a high-bit-rate fiber-optic communication system requires attention to many details, as the number of physical phenomena necessary to take into account may be quite significant. Full quantitative analysis is thus very complex and is usually done via extensive numerical simulations. When a novel idea is proposed, however, it is legitimate to focus on the most relevant effects only to see the potential of the new technique.

In this work, we consider long-haul optically amplified single-channel dispersion-managed soliton (DMS) systems with no in-line control elements like optical filters or modulators. We assume that solitons are only affected by amplitude jitter, Gordon–Haus timing jitter, and pulse-to-pulse interaction. This is a minimum set of physical phenomena that are necessary to include in a DMS study. We do not take into account such effects as third-order dispersion or polarization-mode dispersion.

The theory of single-channel DMS data transmission is well developed, and the key design tradeoffs are well understood. First, stronger dispersion management both reduces Gordon–Haus timing jitter and improves the signal-to-noise ratio (SNR). At the same time, pulse-to-pulse interaction increases at high map strengths, limiting the system performance severely. It is known that there exists an optimum map strength for which soliton interaction is the weakest [2], and this point of minimum interaction is often chosen as a system operating point [3], [4]. Second, smaller average dispersion $|\bar{\beta}_2|$ implies less timing jitter and weaker interaction but lower SNR at the receiver.

We quantify the channel impairments through the following parameters:

- 1) p_u —probability of bit error due to the amplitude jitter;
- 2) p_t —probability of bit error due to the timing jitter;
- 3) L_{coll} —collision distance of a pair of solitons.

Manuscript received December 8, 2004; revised November 8, 2005.

The authors are with the Edward S. Rogers, Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: vlad@comm.utoronto.ca; frank@comm.utoronto.ca).

Digital Object Identifier 10.1109/JLT.2005.863265

Each parameter reflects the influence of the corresponding effect as if all the other factors were absent. They can be estimated as follows.

To obtain L_{coll} , the propagation of two pulses with prescribed separation is simulated and the point of their collision is recorded. In this case, in-line optical amplifiers are assumed to be ideal, i.e., amplified spontaneous emission (ASE) is turned off.

To estimate p_t , no pulse distortion is assumed and the pulse arrival time is approximated by a zero-mean Gaussian distribution with the known standard deviation σ_{GH} . Then, p_t is taken to be the probability that the peak of the pulse moves out of its original bit slot, i.e.,

$$p_t = \text{erfc} \left(\frac{T_B}{2\sqrt{2}\sigma_{\text{GH}}} \right).$$

The probability of error due to the amplitude jitter p_u strongly depends on the details of the receiver. We will use a simple receiver structure that makes the computation of p_u feasible. The details are given in Section IV.

Traditional design of a DMS system includes the estimation of the Q -factor from an eye diagram obtained with the help of Monte Carlo simulations. This approach allows one to see how the combination of all the effects involved affects the system performance. A popular milestone is a system with the probability of error equal to 10^{-9} or less. In our study, however, we deal with bit error rates on the order of 10^{-2} and measure the probability of error directly from simulations.

As a benchmark with which to compare our coded system, we design a conventional periodic DMS system in accordance with the following guidelines. First, σ_{GH} must not exceed $0.08T_B$, which means that p_t is on the order of 10^{-9} or less. Second, p_u may be as large as 10^{-2} assuming that an error-correcting code (ECC) can be applied afterward to achieve the target bit error rate. Third, the dispersion map strength S_{map} must correspond to the point of minimum soliton interaction or, equivalently, maximum collision distance. Fourth, L_{coll} must be large enough in comparison with the transmission distance so that pulse-to-pulse interactions would contribute negligibly to the timing jitter. Note that we allow large amplitude jitter but impose a severe limitation on timing jitter. The reason is given in [5]. Large timing jitter means that solitons are likely to come close to each other, which implies enhanced pulse-to-pulse interaction. As a result, the system performance degrades quickly [5]. We turn to the description of the coding scheme and its properties in the next section.

III. CODING SCHEME

A. Introduction to Coding for Constrained Channels

A binary channel is said to be runlength limited (RLL), or (d, k) constrained, if channel sequences are required to have at least d but at most k “0”s between any pair of adjacent “1”s. Information transmission through an RLL channel implies two necessary steps. First, an arbitrary source word must be mapped unambiguously to a sequence that satisfies the channel

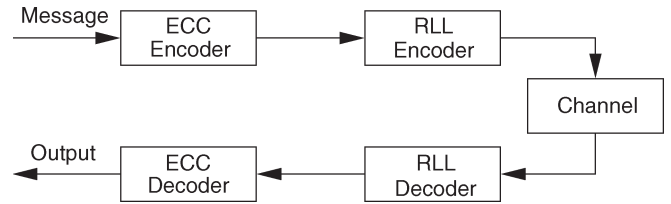


Fig. 1. Traditional ECC–RLL concatenation.

constraints. Second, some error control mechanism must be implemented to protect the data from noise.

A popular approach is to utilize a concatenation of an inner modulation RLL (or constrained) code with an outer ECC (Fig. 1). A message to be transmitted is first encoded with the ECC. The result is supplied to the RLL encoder (or modulator). The modulator transforms an incoming unconstrained word to a channel sequence that satisfies the required constraints. At the receiver, the decoders work in the reverse order. The received word is first demodulated by the RLL decoder (or demodulator) and the ECC decoder cleans up possible errors afterward.

Proper design of the ECC requires the knowledge of the intermediate channel at the output of the RLL decoder. The knowledge of the physical channel is not enough for this purpose as errors may also be introduced by the RLL decoder itself. This phenomenon is referred to as error propagation or error amplification and may considerably increase the burden on the ECC. In this section, we pursue two goals. 1) We construct an RLL coding scheme with very limited error propagation. 2) We aim at channel estimation at the output of the demodulator. The design of the ECC for a given channel is a well-known problem addressed elsewhere so we do not consider it here (but see, e.g., [6]).

B. RLL Code Construction

In this work, we choose a $(2, \infty)$ -constraint. Our choice is justified by the following arguments. First, we do not really need a k -constraint, which is usually imposed for the purpose of better clock synchronization. Second, the maximum code rate for the $(2, \infty)$ -constraint is known to be approximately 0.5515. Hence, it should be possible to design a rate 1/2 code with simple and fast encoding and decoding algorithms, keeping the rate efficiency high enough at the same time. Third, if the minimum pulse separation is fixed, the channel becomes noisier very quickly as d increases simply because T_B gets smaller. We hope that the degradation of the channel for $d = 2$ is still tolerable.

Now, let us describe the code itself (see also [7]). The encoding algorithm maps a source word $u = (u_1, \dots, u_m)$ of length m to a codeword x of length $n = 2m$ with the help of Table I. First, u is parsed into elementary blocks of at most 3 bits, only the blocks from the first column of Table I being allowed. If u has “01” or “011” at the end, a dummy “0” is appended to u . Each block is then mapped to the corresponding channel sequence, or phrase. If an extra “0” was appended to u , the two trailing “0”s of x are discarded. For example, let $u = “101”$. After a dummy “0” is appended, u parses as “10,”

TABLE I
ENCODING TABLE

User Data	Channel Bits
0	00
10	1000
110	010000
111	100100

“10.” Since “10” maps to “1000,” $x = “10001000.”$ We discard the last two “0”s and finally obtain $x = “100010.”$

To concatenate two constrained words $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$ without a constraint violation, we employ a trick from [8]. Note that x_n is always 0. The 2-constraint can then be violated if and only if $x_{n-1} = y_1 = 1$. In this case, we set $x_{n-1} = y_1 = 0$ and $x_n = 1$. Thus, x_n works as a merging bit. To decode x , x_n will be used by the RLL decoder along with x_0 , the merging bit preceding x_1 . It is also important to mention that since n is even, x_1 is always located at an odd position in an RLL-coded bit stream.

The decoder is implemented in a sliding-window fashion (Fig. 2). A window of length 3 slides along the word $x = (x_0, x_1, \dots, x_n)$ by two positions each time. The i th output bit is the logical “OR” (denoted with “+”) of all three bits in the window: $u_i = x_{2i-2} + x_{2i-1} + x_{2i}$, $i = 1, \dots, m$. Thus, if $x_j = 1$, it “triggers” two “1”s at the output of the decoder if j is even and a single “1” if j is odd.

Note that since both the encoding table and the decoding rule are extremely simple, very little hardware is needed for their implementation. If the encoder is required to have a constant output rate, it can be slightly modified and implemented as a finite-state machine with 12 states. Concatenation of codewords can then be done by simply padding each codeword with one or two merging “0”s. If the block length is large, this loss in code rate is negligible.

C. Intermediate Channel Statistics

The output of the demodulator is an unconstrained binary sequence $v = (v_1, \dots, v_m)$ with possible errors introduced by both the channel and the demodulator itself. We want to know how u and v are related probabilistically. To simplify the problem, we ignore possible correlation between different symbols of v , i.e., we assume a memoryless channel. Since u and v are binary words, the simplified intermediate channel is fully characterized by the crossover probabilities $a_c = p(v_i = 1|u_i = 0)$ and $b_c = p(v_i = 0|u_i = 1)$. Here, index c stands for “constrained.” We also naturally introduce p_c , the overall probability of error, as $p_c = (a_c + b_c)/2$.

To estimate a_c and b_c , we make use of the facts that the timing jitter is small and there are at least two “0”s between any pair of “1”s. In particular, if “000” is sent, the central “0” is not affected by pulses that occupy other bit slots. Furthermore, if “010” is sent, only the central “1” and no other “1”s can potentially influence either “0” due to the timing jitter. In this approximation, we can consider only triples of bits and compute the conditional probabilities $p(xyz|0w0)$ that “ xyz ” is received given that “ $0w0$ ” is sent. Let \mathcal{E} denote the set of all possible

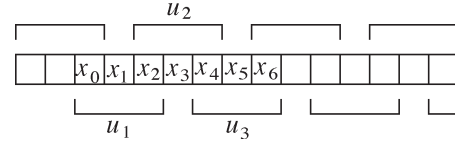


Fig. 2. Sliding-window decoder.

types of errors. We introduce the following notation for the error events and their corresponding probabilities:

- 1) $p_{di} = p(010|000)$ —drop-in (DI);
- 2) $p_{do} = p(000|010)$ —drop-out (DO);
- 3) $p_{ls} = p(100|010)$ —left bit shift (LS);
- 4) $p_{rs} = p(001|010)$ —right bit shift (RS);
- 5) $p_{lsdi} = p(110|010)$ —left stimulated drop-in (LSDI);
- 6) $p_{rsdi} = p(011|010)$ —right stimulated drop-in (RSDI);
- 7) $p_{ddi} = p(111|010)$ —double drop-in (DDI);
- 8) $p_{non} = p(101|010)$ —nonsense (NON).

Thus, $\mathcal{E} = \{DI, DO, LS, RS, LSDI, RSDI, DDI, NON\}$. Since the timing jitter obeys a zero-mean symmetric distribution, we can also define the probability of a bit shift as $p_s = p_{ls} + p_{rs}$ and the probability of the stimulated drop-in as $p_{sdi} = p_{lsdi} + p_{rsdi}$.

The crossovers a_c and b_c can be estimated following the technique described in [9]. To simplify the argument, we let $n \rightarrow \infty$ and neglect merging effects.

The key idea is stated as follows. We transmit a very long constrained sequence x and corrupt it by a single channel error of a certain type $T \in \mathcal{E}$ at a randomly selected position. Then we manually demodulate the received sequence and record the number of errors committed. Let $N_{01}(T)$ and $N_{10}(T)$ be random variables representing the number of erroneous 0-to-1 and 1-to-0 transitions, respectively, when a channel error of type T occurs. We construct their probability mass functions by averaging over all possible error locations. It turns out to be feasible due to the short decoding window. Taking the expected values, $E[N_{01}(T)]$ and $E[N_{10}(T)]$, we obtain the average number of demodulation errors per single channel error of type T .

Next, we compute $N_{ch}(T)$, the average number of channel errors of type T expected to occur when a random x is sent. Since u has length m and contains an equal number of “0”s and “1”s on average, the crossovers are calculated as $a(T) = 2N_{ch}(T)E[N_{01}(T)]/m$ and $b(T) = 2N_{ch}(T)E[N_{10}(T)]/m$. Here, $a(T)$ and $b(T)$ (with index c omitted for brevity) are “contributions” to a_c and b_c from effect T . We approximate a_c and b_c as a sum of independent $a(T)$ and $b(T)$ over all types of errors. A detailed derivation of a_c and b_c is left for the Appendix, the final result being

$$a_c = \frac{37}{112}(p_s + p_{sdi} + 2p_{ddi} + 2p_{non}) + \frac{131}{56}p_{di} \quad (1)$$

$$b_c = \frac{2}{7}p_s + p_{do}. \quad (2)$$

D. Discussion

There are small corrections to a_c and b_c due to possible cancellation of neighboring errors. For example, let $u = “111”$

so that $x = "100100"$ is sent. Suppose the second "1" shifts to the right so $y = "100010"$ is received. The demodulator outputs "101" making an error. However, if the first "1" also shifts to the right and $y = "010010"$ is now received, the decoder makes no error. Thus, one bit shift is cancelled out by another one.

Despite this potential mixture of channel errors, we found that (1) and (2) are in a good agreement with Monte Carlo simulations as long as $a_c < 0.1$ and $b_c < 0.1$. For noisier channels, errors occur more frequently and tend to cancel each other more often, so the true crossovers are actually less than predicted. Such noisy channels are unlikely to be practically interesting, however.

Of course, our model does not include important higher order effects like multiple-position bit shifts and errors caused by pulse-to-pulse interactions. One can conclude, however, that they are not detrimental to the system. For example, a two-position bit shift causes no more than two errors. Since timing jitter has a Gaussian distribution, it is a much less likely event than a simple bit shift. Hence, multiple-position bit shifts should not contribute to the overall probability of error appreciably.

To illustrate potential consequences of soliton interaction, let "100100" be sent and let the "1"s strongly attract each other. Possible received patterns are, for example, "110100," "101100," "111100," or "011000." Nevertheless, the RLL decoder shows impressive resilience to such error events. The first three words are demodulated correctly and only the last sequence causes the decoder to commit an error.

An interesting question is whether there exist other RLL codes of the same rate with better properties. We compared the code proposed here with one that can be obtained by a straightforward application of the state-splitting algorithm [1]. The alternative code we constructed had a three-state encoder and a sliding-window decoder with the window size of 4 bits. Despite a wider decoding window, that code was slightly better at handling drop-outs and drop-ins. However, the probability of error due to bit shifts was greater by a factor of almost 2. Thus, if bit shifts dominate, our coding scheme is a good choice.

IV. FROM PHYSICAL FIBER TO CHANNEL MODEL

The purpose of this section is to obtain a relation between the system parameters and the probabilities of error events in \mathcal{E} . We first recall how the probability of error is calculated for the benchmark system.

A. Probability of Error for the Benchmark System

Let us assume a conventional receiver model consisting of an ideal optical bandpass filter of bandwidth B_o , a square-law detector, an integrate-and-dump electrical filter, and a slicer. An important parameter of this receiver structure is the so-called number of noise modes $2B_oT_B + 1$. For mathematical convenience, it is often assumed to be an even integer, $2M$. Let E_0 be the pulse energy and N_0 be the noise power spectral density. The slicer compares the output of the integrator with a threshold E_d and decides that a "1" was sent if the output is greater than E_d . In the absence of timing jitter and pulse-to-pulse

interaction, the probability of making an error is determined by [10], [11]

$$p(1|0) \equiv a_u = \exp\left(-\frac{E_d}{N_0}\right) \sum_{i=0}^{M-1} \frac{1}{i!} \left(\frac{E_d}{N_0}\right)^i \quad (3)$$

$$p(0|1) \equiv b_u = 1 - Q_M\left(\sqrt{\frac{2E_0}{N_0}}, \sqrt{\frac{2E_d}{N_0}}\right) \quad (4)$$

where Q_M is the generalized Marcum Q -function of order M [12]. Note that a_u and b_u have the same meaning as a_c and b_c in Section III, i.e., they can be thought of as the crossover probabilities of a binary asymmetric channel (BAC). Here, index u stands for "unconstrained." Expressions (3) and (4) may also be used in the presence of timing jitter as long as amplitude jitter dominates [13].

B. Probabilities of Error Events for the Constrained System

As soon as we introduce constrained coding and reduce T_B , timing jitter can no longer be neglected. Let δ denote a random variable representing the pulse arrival time with respect to its original position. In the absence of soliton interaction, δ is known to have a zero-mean Gaussian distribution, which is defined by

$$p_\delta(\delta) = \frac{1}{\sqrt{2\pi}\sigma_{GH}} \exp\left(-\frac{\delta^2}{2\sigma_{GH}^2}\right)$$

where σ_{GH}^2 is the Gordon–Haus timing jitter variance.

In the presence of strong timing jitter, the SNR becomes a random variable with the distribution depending on both δ and the pulse shape parameters. We assume that DMSs at the point of detection have an unchirped Gaussian envelope expressed as

$$U(t) = \sqrt{\frac{E_0}{\sqrt{\pi}T_m}} \exp\left(-\frac{t^2}{2T_m^2}\right)$$

where T_m is the characteristic width, related to the full-width at half-maximum of the pulse τ through

$$\tau = 2\sqrt{\ln 2}T_m \approx 1.665T_m.$$

Now, let an isolated soliton be sent and displaced randomly according to the distribution of δ . If δ is given, the fraction of pulse energy remaining within the detection window $[-T_B/2; T_B/2]$ is given as

$$g_1(\delta) = \frac{1}{E_0} \int_{-\frac{T_B}{2}}^{\frac{T_B}{2}} |U(t - \delta)|^2 dt.$$

If we introduce the error function

$$\text{erf}(x) = \frac{2}{\pi} \int_0^x \exp(-u^2) du$$

then g_1 can also be expressed in terms of $\text{erf}(\cdot)$ as

$$g_1(\delta) = \frac{1}{2} \left[\text{erf} \left(\frac{\frac{T_B}{2} + \delta}{T_m} \right) + \text{erf} \left(\frac{\frac{T_B}{2} - \delta}{T_m} \right) \right]. \quad (5)$$

Due to the timing jitter, a portion of the soliton may enter the left or the right neighboring bit slot. If δ is given, the fraction of energy transferred to the left bit slot is given by

$$g_{0L}(\delta) = \frac{1}{E_0} \int_{-\frac{3T_B}{2}}^{-\frac{T_B}{2}} |U(t - \delta)|^2 dt$$

which can also be written as

$$g_{0L}(\delta) = \frac{1}{2} \left[\text{erf} \left(\frac{\frac{3T_B}{2} - \delta}{T_m} \right) - \text{erf} \left(\frac{\frac{T_B}{2} - \delta}{T_m} \right) \right]. \quad (6)$$

Similarly, for the fraction of energy transferred to the right bit slot, we have

$$g_{0R}(\delta) = \frac{1}{E_0} \int_{\frac{T_B}{2}}^{\frac{3T_B}{2}} |U(t - \delta)|^2 dt$$

$$g_{0R}(\delta) = \frac{1}{2} \left[\text{erf} \left(\frac{\frac{3T_B}{2} + \delta}{T_m} \right) - \text{erf} \left(\frac{\frac{T_B}{2} + \delta}{T_m} \right) \right]. \quad (7)$$

Now, we are in a position to estimate the probabilities of possible error events. Recall that they are defined as the conditional probabilities $p(xyz|0w0)$ that “ xyz ” is received given that “ $0w0$ ” is sent. If $w = 0$ (a drop-in), we have exactly the same situation as in the unconstrained case so that

$$p(010|000) = p_{\text{di}} = \exp \left(-\frac{E_d}{N_0} \right) \sum_{i=0}^{M-1} \frac{1}{i!} \left(\frac{E_d}{N_0} \right)^i.$$

When $w = 1$, the key step is to realize that x , y , and z are independent if δ is given. Therefore, $p(xyz|010, \delta)$ factors as

$$p(xyz|010, \delta) = p(x**|010, \delta)p(*y*|010, \delta)p(**z|010, \delta) \quad (8)$$

where the symbols that do not matter are replaced by asterisks. Consequently, we only need to compute the following “elementary” probabilities: $p(1**|010, \delta)$; $p(*0*|010, \delta)$; and $p(**1|010, \delta)$. They can be readily obtained with the help of (4)–(7) as

$$p(*0*|010, \delta) = 1 - Q_M \left(\sqrt{\frac{2g_1 E_0}{N_0}}, \sqrt{\frac{2E_d}{N_0}} \right) \quad (9)$$

$$p(1**|010, \delta) = Q_M \left(\sqrt{\frac{2g_{0L} E_0}{N_0}}, \sqrt{\frac{2E_d}{N_0}} \right) \quad (10)$$

$$p(**1|010, \delta) = Q_M \left(\sqrt{\frac{2g_{0R} E_0}{N_0}}, \sqrt{\frac{2E_d}{N_0}} \right). \quad (11)$$

The dependence on δ comes through g_1 , g_{0L} , and g_{0R} , respectively. The other probabilities of interest are immediately obtained from (9)–(11). For example, $p(0**|010, \delta) = 1 - p(1**|010, \delta)$. Finally, $p(xyz|010)$ for any triple xyz is obtained by integrating δ out of

$$p(xyz|010) = \int_{-\infty}^{\infty} p(xyz|010, \delta) p_{\delta}(\delta) d\delta. \quad (12)$$

Equations (8)–(12) completely establish the connection between the physical channel and the error events in set \mathcal{E} .

V. CONSTRAINED CODING FOR DMSS

In this section, we give a full description of the main idea and illustrate it with the help of numerical examples. Since the primary goal of this paper is to show how constrained coding increases the capacity of a DMS system, we do not try to find “optimum” dispersion maps for benchmark systems. Instead, we consider maps with reasonably well-chosen parameters and adjust other system parameters accordingly to satisfy the design criteria enumerated in Section II. Then we apply constrained coding and observe how much gain in channel capacity can potentially be achieved.

A. System Comparison Approach

Let $C(a, b)$ stand for the capacity (in bits) of a binary asymmetric channel (BAC) with crossover probabilities a and b . $C(a, b)$ is known to be [14]

$$C(a, b) = \log_2 \left[2^{\frac{aH(b) + (b-1)H(a)}{1-a-b}} + 2^{\frac{bH(a) + (a-1)H(b)}{1-a-b}} \right]$$

where $H(x) = -x \log_2(x) - (1-x) \log_2(1-x)$ is the binary entropy function. The benchmark system is characterized by a BAC with the crossovers a_u and b_u given by (3) and (4). The corresponding channel capacity in bits per second is obviously $C_u = C(a_u, b_u)/T_B$.

The situation after the application of constrained coding is slightly different. The underlying BAC is taken at the output of the demodulator and has crossover probabilities a_c and b_c given by (1) and (2). Due to the application of a rate 1/2 RLL code, the channel capacity in bits per second becomes $C_c = C(a_c, b_c)/2T_B$.

That C_c can be greater than C_u comes from the possibility of making the bit slot shorter. Since every pair of “1”s is now separated by at least two “0”s, T_B can be reduced by a factor of up to 3 while preserving at least the same minimum pulse separation Δ . This indicates a potential gain in channel capacity of about 50%.

Since such significant reduction of T_B implies a noticeable channel degradation because of the timing jitter, it is not obvious that C_c will necessarily be greater than C_u . To demonstrate it, we first consider a somewhat abstract example of noninteracting Gaussian pulses, when both p_u and σ_{GH} may be varied arbitrarily and independently of each other. The benchmark system parameters used in our experiment are $E_0 =$

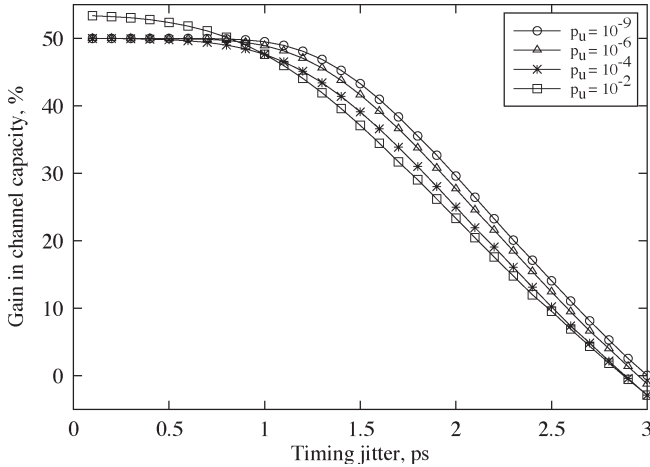


Fig. 3. Gain in channel capacity after the application of constrained coding in our ideal experiment. $\Delta = 24$ ps, $T_B = 8$ ps, and $T_m = 3$ ps.

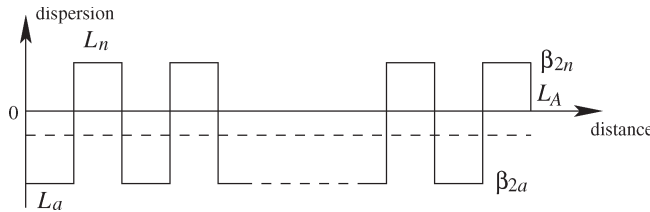


Fig. 4. Dispersion map.

0.1 pJ, $T_m = 3$ ps, $B_o = 375$ GHz, and $T_B = \Delta = 24$ ps. The constrained system has the same parameters, except for $T_B = 8$ ps. We compute p_u based on (3) and (4) for four different noise levels N_0 . Then, for each p_u , we estimate p_c as in (1) and (2) for 30 different values of σ_{GH} . The corresponding C_u and C_c are compared in Fig. 3. For small values of σ_{GH} , potential gain is indeed around 50%. It may be even higher than that in the case $p_u = 10^{-2}$ simply because $p_c < p_u$ in the region of very low timing jitter. Fig. 3 also shows that designing benchmark systems with low amplitude jitter is not so important from the point of view of constrained systems.

B. Real Benchmark System Design

We employ a conventional dense dispersion management with alternating fiber segments of anomalous and normal dispersion (Fig. 4). Let the lengths of the anomalous and normal dispersion fiber segments be $L_a = L_n = 8$ km. Let the corresponding local dispersion parameters be $\beta_{2a} = -4$ ps²/km and $\beta_{2n} = 3.98$ ps²/km. The average dispersion is

$$\bar{\beta}_2 = \frac{\beta_{2a}L_a + \beta_{2n}L_n}{L_a + L_n} = -0.01 \text{ ps}^2/\text{km}$$

and the map strength can be written as

$$S_{\text{map}} = \frac{\beta_{2n}L_n - \beta_{2a}L_a}{(1.665T_m)^2}$$

All fibers have attenuation $\alpha = 0.25$ dB/km and a nonlinear parameter $\gamma = 2.5$ W⁻¹/km. Optical amplifiers with gain $G = 100$ are placed at the beginning of the anomalous dis-

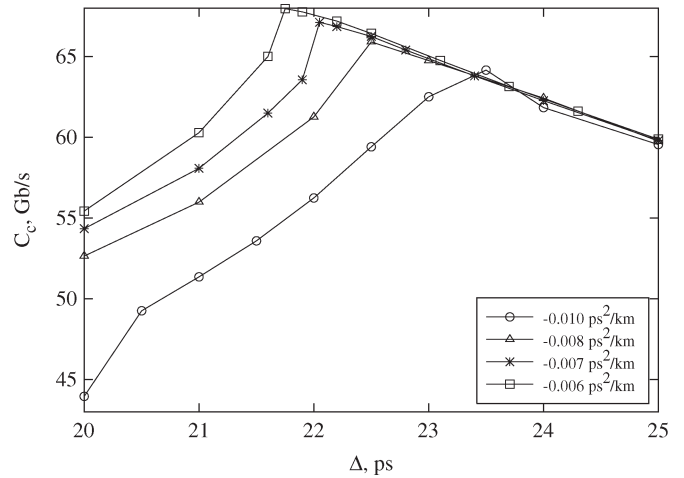


Fig. 5. C_c as a function of Δ for different $\bar{\beta}_2$.

persion span every $L_A = 80$ km, their spontaneous-emission factor being $n_{sp} = 1.4$. The amplifiers produce ASE noise with power spectral density $S_{sp} = n_{sp}h\nu(G - 1)$, where h is the Planck's constant and ν is the optical frequency. We assume $\lambda = 1.55$ μm .

The detection is done at a point where the pulsewidth is minimum. This implies an extra preamplifier with different gain at the receiver. Since the number of in-line amplifiers N_A is very large, the relative impact of the preamplifier on p_u is negligible anyway. The noise power spectral density at the receiver is thus taken to be $N_0 = N_A S_{sp}$.

Gordon-Haus timing jitter variance σ_{GH}^2 is governed by the pulse shape $V(t)$ at the amplifier. We approximate $V(t)$ as a chirped Gaussian, which is defined as

$$V(t) = \sqrt{\frac{E_0}{\sqrt{\pi}T_0}} \exp \left[-\frac{1 + iC_0}{2} \left(\frac{t}{T_0} \right)^2 \right]$$

T_0 and C_0 are obtained numerically with the help of the averaging method described in [15]. Then, σ_{GH}^2 is obtained as proposed in [13], the leading term being

$$\sigma_{GH}^2 = \frac{S_{sp}}{E_0} \frac{|\bar{\beta}_2|^2}{3T_m^2 L_A} L_T^3$$

where $L_T = N_A L_A$ is the total transmission distance.

We choose $T_B = 20$ ps and $L_T = 6400$ km and select S_{map} corresponding to the weakest pulse-to-pulse interaction. The benchmark system designed has $T_m = 3.75$ ps, $L_{\text{coll}} = 29.44$ Mm, $p_u = 0.01$, and $\sigma_{GH} = 1.74$ ps. The channel capacity is estimated to be $C_u = 45.9$ Gb/s.

When we apply constrained coding to this system, we consider variable Δ and $T_B = \Delta/3$. Greater initial pulse separation means that soliton interaction is weaker. Consequently, we can launch pulses of higher energy, which both reduces the timing jitter and improves the SNR. To make some quantitative comparisons, we assume that the highest map strength we are allowed to use is the one for which L_{coll} is the same as for the benchmark system. The resultant capacity curve is the one that corresponds to $\beta_2 = -0.01$ ps²/km in Fig. 5. The maximum

TABLE II
EXTENDED ENCODING TABLE

Input	Phrase	X	$Q(X)$	$p_1(X)$	$p_0(X)$	$p_{11}(X)$	$p_{00}(X)$
0	00	A	1/2	—	8/23	—	4/23
10	1000	B	1/4	2/5	6/23	2/5	2/23
110	010000	C	1/8	1/5	5/23	1/5	1/23
111	100100	D	1/8	2/5	4/23	1/5	1/23

C_c in this case is 64.2 Gb/s, a 40% increase in comparison with C_u .

Another thing that we can do is to change β_{2n} slightly. We consider three other values of β_{2n} corresponding to the average dispersion of -0.008 , -0.007 , and -0.006 ps²/km. The capacity curves are also plotted in Fig. 5. The maximum C_c obtained is 68 Gb/s, a 48% increase this time!

VI. CONCLUSION

In this paper, we introduced constrained coding as a new approach to DMS data transmission and proposed a particular coding scheme. The new method exploits the key tradeoffs in DMS system design and permits the usage of higher than usual map strengths. The results obtained in our examples indicate that appreciable gains in the bit rate (of up to 50%) are quite achievable with negligible increase in system complexity.

Our study assumes an ideal ECC, whose rate R is equal to the channel capacity. For example, if $a_c = b_c = 0.013$, $R = C_c = 0.9$. Practical ECCs always have $R < C_c$ implying an additional loss in data rate. However, recent results indicate that designing codes with rates close to capacity is feasible [16]. Furthermore, if p_u and p_c are comparable, both the benchmark and the constrained system require ECCs with approximately the same redundancy. Therefore, our comparison is fair in this case even when nonideal ECCs are assumed.

APPENDIX

Here, we present a detailed derivation of (1) and (2).

A. Distribution of “0”s and “1”s

First, we need a few auxiliary quantities concerning the distribution of “0”s and “1”s. We will use Table II, which is an extended version of Table I. The data is obtained as follows. Suppose, a very long sequence of N channel phrases is generated randomly. Let $X \in \{A, B, C, D\}$ denote a phrase type. Then, the symbols in Table II have the following meaning.

- 1) $Q(X)$ is the probability that a randomly picked phrase is of type X .
- 2) $p_1(X)$ is the probability that a randomly picked “1” belongs to a phrase of type X .
- 3) Similarly, $p_0(X)$ is the probability that a randomly picked “0” belongs to a phrase of type X .
- 4) $p_{11}(X)$ is the probability that a randomly picked “1” comes from a phrase of type X and corresponds to the “1” at a certain position. $p_{11}(X) = p_1(X)$ for all X except $X = D$ because phrase D has two “1”s.
- 5) p_{00} is the probability that a randomly picked “0” comes from a phrase of type X and corresponds to the “0” at a certain position.

Let $L(X)$ be the length of the channel phrase X and $l(X)$ be the length of the corresponding user sequence. Then $Q(X) = 2^{-l(X)}$. The average phrase length L_{av} is computed as

$$L_{av} = \sum_X L(X)Q(X) = \frac{7}{2}.$$

Next, let $N_1(X)$ and $N_0(X)$ denote the number of “1”s and “0”s in the phrase of type X , respectively. Then the average number of “1”s/“0”s in a random sequence of N phrases is given by

$$M_i = N \sum_X N_i(X)Q(X), \quad i = 0, 1. \quad (13)$$

We obtain $M_1 = (5/8)N$ and $M_0 = (23/8)N$.

Now, we are in a position to find p_i s. On average, a sequence of N phrases contains $NN_1(X)Q(X)$ “1”s and $NN_0(X)Q(X)$ “0”s that belong to a phrase of type X . Therefore

$$p_1(X) = \frac{NN_1(X)Q(X)}{M_1} = \frac{8}{5}N_1(X)Q(X)$$

$$p_0(X) = \frac{NN_0(X)Q(X)}{M_0} = \frac{8}{23}N_0(X)Q(X).$$

It is clear that $p_{11}(X) = p_1(X)/N_1(X)$ and $p_{00}(X) = p_0(X)/N_0(X)$. This allows us to complete Table II.

Finally, we compute the probabilities q_1 and q_0 that a randomly picked bit is a “1” or a “0,” respectively, as

$$q_1 = \frac{M_1}{NL_{av}} = \frac{5}{28} \quad q_0 = \frac{M_0}{NL_{av}} = \frac{23}{28}. \quad (14)$$

We now turn to the calculation of $N_{01}(T)$ and $N_{10}(T)$ for each error type T .

B. Single Drop-Out

It is obvious that a drop-out contributes to $N_{10}(\text{DO})$ only so $N_{01}(\text{DO}) \equiv 0$ and $a(\text{DO}) = 0$. From the decoding rule, we see that the sliding window always spans two even bits and one odd. Therefore, a “1” located at an odd position appears in the window only once so its drop-out results in just one error. On the other hand, a “1” located at an even position participates in two “OR” operations so its drop-out causes two errors. Let EP and OP denote the events that a randomly picked “1” is at an even or odd position, respectively. With the help of Table II, we obtain $p(\text{EP}) = p_{11}(C) + p_{11}(D) = 2/5$ and $p(\text{OP}) = p_{11}(B) + p_{11}(D) = 3/5$. Hence, the number of demodulation errors $N_{10}(\text{DO}) \in \{1, 2\}$ per single drop-out has a distribution $p(N_{10}(\text{DO}) = 1) = 3/5$ [$p(N_{10}(\text{DO}) = 2) = 2/5$]. The expected value of $N_{10}(\text{DO})$ is $E[N_{10}(\text{DO})] = 7/5$, which is slightly less than the intuitive value of 1.5.

To compute $N_{ch}(\text{DO})$, we notice that a channel sequence of length n has nq_1 “1”s on average, each of which independently drops out with probability p_{do} . Thus, $N_{ch}(\text{DO}) = nq_1p_{do}$. We finally obtain $b(\text{DO}) = p_{do}$.

TABLE III
ERRORS INDUCED BY LEFT BIT SHIFTS

YX	Input	Transmitted	Output	$N_{01}(LS)$	$N_{10}(LS)$
AB	010	00 <u>1</u> 000	110	1	0
BB	1010	1000 <u>1</u> 000	1110	1	0
CB	11010	010000 <u>1</u> 000	11110	1	0
DB	11110	100100 <u>1</u> 000	11110	0	0
AC	0110	00 <u>01</u> 0000	0100	0	1
BC	10110	1000 <u>01</u> 0000	10100	0	1
CC	110110	010000 <u>01</u> 0000	110100	0	1
DC	111110	100100 <u>01</u> 0000	111100	0	1
AD	0111	00 <u>1001</u> 00	1111	1	0
BD	10111	1000 <u>1001</u> 00	111111	1	0
CD	110111	010000 <u>1001</u> 00	111111	1	0
DD	111111	100100 <u>1001</u> 00	111111	0	0
$D2$	111	100 <u>100</u>	110	0	1

TABLE IV
ERRORS INDUCED BY RIGHT BIT SHIFTS

YX	Input	Transmitted	Output	$N_{01}(RS)$	$N_{10}(RS)$
AB	010	00 <u>1000</u>	011	1	0
BB	1010	1000 <u>1000</u>	1011	1	0
CB	11010	010000 <u>1000</u>	11011	1	0
DB	11110	100100 <u>1000</u>	11111	1	0
AC	0110	00 <u>010000</u>	0010	0	1
BC	10110	1000 <u>010000</u>	10010	0	1
CC	110110	010000 <u>010000</u>	110010	0	1
DC	111110	100100 <u>010000</u>	111010	0	1
AD	0111	00 <u>100100</u>	0111	0	0
BD	10111	1000 <u>100100</u>	101111	0	0
CD	110111	010000 <u>100100</u>	110111	0	0
DD	111111	100100 <u>100100</u>	111111	0	0
$D2$	111	100 <u>100</u>	101	0	1

C. Single Bit Shift

We follow an approach suggested in [9]. Suppose that a “1” from a phrase X gets shifted. Consider combinations YXZ of X with all possible noiseless “pasts” Y and “futures” Z . Both Y and Z must be long enough to make sure that the influence of the shift on all the bits in the sequence YXZ is taken into account. Decoding is then done for each valid YXZ assuming that some “1” from X is shifted to the left or to the right. The number of demodulation errors in each case is determined manually.

A closer look at the encoding table reveals, however, that in our case, a bit shift in X never affects any future phrase Z . This is due to the fact that only the last bit of X takes part in the decoding of Z and the last bit of any phrase is always “0.”

Therefore, we restrict ourselves to the combinations of the form YX only. Notice that X cannot be equal to A , which does not have a “1,” and Y can be anything. Thus, 12 combinations are possible. There is a special case, however, when the second “1” of D is selected. Its shift, however, cannot affect any past Y . We denote this case symbolically as $D2$.

The results are summarized in Tables III and IV. The bits corrupted are underlined. One can see that all four random variables involved take on values 0 or 1 only, the probability masses at 1 being

$$p(N_{01}(LS) = 1) = [p_{11}(B) + p_{11}(D)] \{1 - Q(D)\} = \frac{21}{40}$$

$$p(N_{10}(LS) = 1) = p_{11}(C) + p_{11}(D) = \frac{2}{5}$$

$$p(N_{01}(RS) = 1) = p_{11}(B) = \frac{2}{5}$$

$$p(N_{10}(RS) = 1) = p_{11}(C) + p_{11}(D) = \frac{2}{5}.$$

These are equal to the corresponding expected values. For example, $E[N_{01}(RS)] = 2/5$. We can also compute a few other interesting quantities such as the fraction of bit shifts resulting in no errors, which turns out to be equal to 11/80.

Similar to the previous section, $N_{ch}(LS) = N_{ch}(RS) = nq_1p_s/2$. Hence, $a(LS) = (3/16)p_s$ and $b(LS) = a(RS) = b(RS) = (1/7)p_s$.

TABLE V
ERRORS INDUCED BY DOUBLE DROP-INS

YX	Input	Transmitted	Output	$N_{01}(DDI)$
AB	010	00 <u>1000</u>	111	2
BB	1010	1000 <u>1000</u>	1111	2
CB	11010	010000 <u>1000</u>	11111	2
DB	11110	100100 <u>1000</u>	11111	1
AD	0111	00 <u>100100</u>	1111	1
BD	10111	1000 <u>100100</u>	11111	1
CD	110111	010000 <u>100100</u>	111111	1
DD	111111	100100 <u>100100</u>	111111	0

D. Single Stimulated Drop-In

It is clear that $N_{10}(LSDI) = N_{10}(RSDI) \equiv 0$. The rest of the analysis is almost the same as in the previous section. In fact, Tables III and IV can be used to determine that $N_{01}(LSDI)$ and $N_{01}(RSDI)$ actually coincide with $N_{01}(LS)$ and $N_{01}(RS)$, respectively. Consequently, $a(LSDI) = (3/16)p_{sdi}$, $a(RSDI) = (1/7)p_{sdi}$ and $b(LSDI) = b(RSDI) = 0$.

E. Double Drop-In and Nonsense

These error patterns usually make negligible contribution to a and b and are included here for the sake of completeness. Again, $N_{10}(DDI) = N_{10}(NON) \equiv 0$. Let a double drop-in be induced by a “1” that belongs to phrase X . As in Appendix C, we analyze combinations YX of X with all possible noiseless pasts Y . It can be immediately concluded that $N_{01}(DDI) = 0$ if $X = C$ or $X = D$ and the second “1” drops in. The other cases are considered in Table V.

$N_{01}(DDI)$ takes on the value of 0, 1, or 2. The corresponding probabilities are defined as

$$p(N_{01}(DDI) = 2) = p_{11}(B) [1 - Q(D)] = \frac{7}{20}$$

$$p(N_{01}(DDI) = 1) = p_{11}(B)Q(D) + p_{11}(D) [1 - Q(D)] = \frac{9}{40}$$

$$p(N_{01}(DDI) = 0) = p_{11}(C) + p_{11}(D) [1 + Q(D)] = \frac{17}{40}.$$

The expected value of $N_{01}(DDI)$ is thus

$$E[N_{01}(DDI)] = \frac{37}{40}.$$

TABLE VI
ERRORS INDUCED BY DROP-INS

XZ	Input	Transmitted	Output	$N_{01}(DI)$
$A-$	0	<u>00</u>	1	1
AA	00	<u>00.00</u>	11	2
AC	0110	<u>00.010000</u>	1110	1
$B-$	10	<u>1000</u>	11	1
BA	100	<u>1000.00</u>	111	2
BC	10110	<u>1000.010000</u>	11110	1
$C-$	110	<u>010000</u>	111	1
$C-$	110	<u>010000</u>	111	1
CA	1100	<u>010000.00</u>	1111	2
CC	110110	<u>010000.010000</u>	111110	1
DA	1110	<u>100100.00</u>	1111	1
DC	111110	<u>100100.010000</u>	111110	0

Again, as in the previous sections, $N_{ch}(DDI) = nq_1p_{ddi}$, and

$$a(DDI) = \left(\frac{37}{56}\right)p_{ddi} \quad b(DDI) = 0.$$

The error event $T = \text{NON}$ can be thought of as a double drop-in followed by a drop-out of the central “1.” However, this additional drop-out has no influence on the output. Therefore, $N_{01}(\text{NON})$ has exactly the same distribution as $N_{01}(DDI)$ and $a(\text{NON}) = (37/56)p_{\text{non}}$.

F. Single Drop-In

The analysis in this section is slightly different from the one stated above. Recall that in a long channel sequence, “1”s occur with probability $q_1 = 5/28$, while “0”s cover up the other $q_0 = 23/28$. Every “1” is accompanied by two surrounding “0”s that must be excluded from the consideration. This is due to the fact that their drop-ins are regarded as stimulated drop-ins and have been taken into account already. The fraction of “0”s to be ignored constitutes $2q_1 = 5/14$ of the total sequence length. We will call the remaining “0”s isolated. The probability that a randomly selected bit is an isolated “0” is $q_{00} = q_0 - 2q_1 = 13/28$. In addition, the probability that a randomly picked “0” is isolated is $q_{\text{iso}} = 13/23$.

We use the same technique as in Appendix-C. Here, the analysis of all possible YXZ combinations is simplified by an observation that a drop-in error in X never affects the decoding of Y . Furthermore, the decoding of Z can be affected only if the last bit of X drops in.

The results are summarized in Table VI. The flipping “0”s are underlined. Naturally, $N_{10}(DI) \equiv 0$. $N_{01}(DI)$ takes on the value of 0, 1, or 2 with the probabilities given as

$$\begin{aligned} p(N_{01}(DI) = 2) &= [p_{00}(A) + p_{00}(B) + p_{00}(C)]Q(A) = \frac{7}{46} \\ p(N_{01}(DI) = 1) &= p_{00}(C) + p_{00}(D)Q(A) \\ &+ [p_{00}(A) + p_{00}(B) + p_{00}(C)]\{1 + Q(A)\} = \frac{75}{184} \\ p(N_{01}(DI) = 0) &= p_{00}(D)Q(C) = \frac{1}{184}. \end{aligned}$$

Since we consider only isolated “0”s, these probabilities do not represent a valid distribution. We must normalize them by $q_{\text{iso}} = 13/23$ with the result

$$\begin{aligned} p(N_{01}(DI) = 2) &= \frac{7}{26} \\ p(N_{01}(DI) = 1) &= \frac{75}{104} \\ p(N_{01}(DI) = 0) &= \frac{1}{104}. \end{aligned}$$

The corresponding expected value is defined by

$$E[N_{01}(DI)] = \frac{131}{104}.$$

A long channel sequence of length n contains nq_{iso} isolated “0”s on average. Hence, $N_{ch} = nq_{\text{iso}}p_{di}$, and

$$a(DI) = \left(\frac{131}{56}\right)p_{di} \quad b(DI) = 0.$$

G. Final Crossovers

The total probability of error is approximated as

$$a_c = \sum_{T \in \mathcal{E}} a(T) \quad b_c = \sum_{T \in \mathcal{E}} b(T).$$

Using the results of Appendices B–F, we obtain

$$a_c = \frac{37}{112}(p_s + p_{sdi} + 2p_{ddi} + 2p_{\text{non}}) + \frac{131}{56}p_{di} \quad (15)$$

$$b_c = \frac{2}{7}p_s + p_{do}. \quad (16)$$

REFERENCES

- [1] K. A. S. Immink, P. H. Siegel, and J. K. Wolf, “Codes for digital recorders,” *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2260–2299, Oct. 1998.
- [2] T. Yu, E. A. Golovchenko, A. N. Pilipetskii, and C. R. Menyuk, “Dispersion-managed soliton interaction in optical fibers,” *Opt. Lett.*, vol. 22, no. 11, pp. 793–795, Jun. 1997.
- [3] T. Hirooka, T. Nakada, and A. Hasegawa, “Feasibility of densely dispersion managed soliton transmission at 160 Gb/s,” *IEEE Photon. Technol. Lett.*, vol. 12, no. 6, pp. 633–635, Jun. 2000.
- [4] J. Mårtensson and A. Berntson, “Dispersion-managed solitons for 160-Gb/s data transmission,” *IEEE Photon. Technol. Lett.*, vol. 13, no. 7, pp. 666–668, Jul. 2001.
- [5] F. Matera and M. Settembre, “Exploitation of the fiber capacity in optically amplified transmission systems,” *Opt. Fiber Technol.*, vol. 4, no. 1, pp. 34–82, Jan. 1998.
- [6] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Upper Saddle River, NJ: Prentice-Hall, 1995.
- [7] V. Pechenkin and F. R. Kschischang, “A new coding scheme for runlength-limited channels,” in *Proc. 22nd Biennial Symp. Communications*, Kingston, ON, Canada, Jun. 2004, pp. 130–132.
- [8] G. Khachatryan and K. A. S. Immink, “Construction of simple runlength-limited codes,” *Electron. Lett.*, vol. 35, no. 2, p. 140, Jan. 1999.
- [9] D. G. Howe and H. M. Hilden, “Shift error propagation in 2, 7 modulation code,” *IEEE J. Sel. Areas Commun.*, vol. 10, no. 1, pp. 223–232, Jan. 1992.

- [10] D. Marcuse, "Derivation of analytical expressions for the bit-error probability in lightwave systems with optical amplifiers," *J. Lightw. Technol.*, vol. 8, no. 12, pp. 1816–1823, Dec. 1990.
- [11] P. A. Humblet and M. Azizoglu, "On the bit error rate of lightwave systems with optical amplifiers," *J. Lightw. Technol.*, vol. 9, no. 11, pp. 1576–1582, Nov. 1991.
- [12] D. A. Shnidman, "The calculation of the probability of detection and the generalized Marcum Q -function," *IEEE Trans. Inf. Theory*, vol. 35, no. 2, pp. 389–400, Mar. 1989.
- [13] V. S. Grigoryan, C. R. Menyuk, and R.-M. Mu, "Calculation of timing and amplitude jitter in dispersion-managed optical fiber communications using linearization," *J. Lightw. Technol.*, vol. 17, no. 8, pp. 1347–1356, Aug. 1999.
- [14] R. B. Ash, *Information Theory*. New York: Dover, 1990.
- [15] J. H. B. Nijhof, W. Forsysak, and N. J. Doran, "The averaging method for finding exactly periodic dispersion-managed solitons," *IEEE J. Sel. Topics Quantum Electron.*, vol. 6, no. 2, pp. 330–336, Mar./Apr. 2000.
- [16] T. Mizouochi, "Recent progress in forward error correction for optical communication systems," *IEICE Trans. Commun.*, vol. E88-B, no. 5, pp. 1934–1946, May 2005.



Vladimir Pechenkin received the B.Sc. and M.Sc. degrees (both with honors) in applied physics and mathematics from the Moscow Institute of Physics and Technology, Moscow, Russia, in 1999 and 2001, respectively. He is currently working toward the Ph.D. degree in electrical engineering at the University of Toronto, Toronto, ON, Canada.

His research interests are mainly in the area of error-control coding, especially for optical communications. He was a recipient of a number of scholarships (including the Edward S. Rogers Sr. Scholarship) while at the University of Toronto.



Frank R. Kschischang (S'83–M'91–SM'00–F'05) received the B.A.Sc. degree (with honors) from the University of British Columbia, Vancouver, BC, Canada, in 1985 and the M.A.Sc. and Ph.D. degrees from the University of Toronto, Toronto, ON, Canada, in 1988 and 1991, respectively, all in electrical engineering.

He is a Professor of Electrical and Computer Engineering and Canada Research Chair in Communication Algorithms at the University of Toronto, where he has been a Faculty Member since 1991. During 1997–1998, he spent a sabbatical year as a Visiting Scientist at the Massachusetts Institute of Technology (MIT), Cambridge. His research interests are focused on the area of coding techniques, primarily on soft-decision decoding algorithms, trellis structure of codes, codes defined on graphs, and iterative decoders. He has taught graduate courses in coding theory, information theory, and data transmission.

Dr. Kschischang was the recipient of the Ontario Premier's Research Excellence Award. From October 1997 to October 2000, he served as an Associate Editor for Coding Theory for the *IEEE TRANSACTIONS ON INFORMATION THEORY*. He also served as Technical Program Co-Chair for the 2004 IEEE International Symposium on Information Theory held in Chicago, IL.