

Constrained Coding for the Deep-Space Optical Channel

Bruce Moision, Jon Hamkins

Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Dr., Pasadena, CA 91109-8099

ABSTRACT

We investigate methods of coding for a channel subject to a large dead-time constraint, i.e. a constraint on the minimum spacing between transmitted pulses, with the deep-space optical channel as the motivating example. Several constrained codes designed to satisfy the dead-time constraint are considered and compared on the basis of throughput, complexity, and decoded error-rate. The performance of an iteratively decoded serial concatenation of a modulation code with an outer code is evaluated and shown to provide significant gains over Reed-Solomon concatenated with Pulse Position Modulation.

Keywords: Constrained coding, error-correcting codes, convolutional codes, Reed-Solomon codes, concatenated codes, pulse position modulation, deep-space communications, optical communications

1. INTRODUCTION

A free-space optical communications system is most efficient when the peak to average power ratio of the signal is large [1,2]. These large ratios can be achieved by M -ary Pulse Position Modulation (PPM), in which $\log_2 M$ bits choose the location of a single pulsed slot in an M -slot frame. In theory, PPM can lead to an unbounded capacity [3], but in practice bandwidth constraints place a limit on capacity [4]. Nevertheless, when no noise is present in the system, it has been shown that PPM is near capacity-achieving [1,5].

A Q-switched laser works well with the PPM format [6,7], because it can successfully confine a large pulse energy to a narrow slot. One side effect of Q-switched lasers, however, is a required delay, or dead-time, between pulses during which the laser is recharged. This delay is significant relative to the pulse duration. PPM may be modified to satisfy the dead-time constraint by following each frame by a period during which no pulses are transmitted. However, this affects the optimality of PPM as a modulation format.

There are more efficient—measured in throughput, or bits/second—ways to transmit information over a channel subject to a dead-time constraint. The problem of signaling efficiently under such a constraint has been well studied for applications in magnetic storage, where a similar restriction is imposed to compensate for the interference between magnetic media corresponding to closely recorded bits. Efficient signaling is affected by a modulation, or *constrained*, code. The deep-space problem is novel in that the dead-time is very large relative to the slot duration—on the order of 256-1024 times the slot duration compared to 1-2 times in magnetic storage applications. This report investigates the application of constrained codes to the deep-space optical channel.

2. PRELIMINARIES

This article considers the binary-input, real-valued output channel model shown in Fig. 1. Information bits are first encoded using an error correcting code. The constrained code then takes the coded bits and further encodes them in a way that ensures the laser can physically transmit them, as we explain here. Time is partitioned into slots of duration T_s during which the laser may either transmit a pulse (a one) or not transmit a pulse (a zero), i.e., On-Off Keying (OOK)—see Fig. 2. In unconstrained OOK, a zero or one may appear in any position within the sequence of transmitted binary symbols. Q-switched lasers requiring dead-time T_d between pulses impose the constraint that at least $d \stackrel{\text{def}}{=} \lceil T_d/T_s \rceil$ zeros occur between ones. Slot synchronization, typically implemented by an early-late gate tracking loop, imposes an additional constraint that no more than T_k seconds may elapse

The work described was funded by the TMOD Technology Program and performed by the Jet Propulsion Laboratory, California Institute of Technology under contract with the National Aeronautics and Space Administration.
Email: {bruce.moision, jon.hamkins}@jpl.nasa.gov

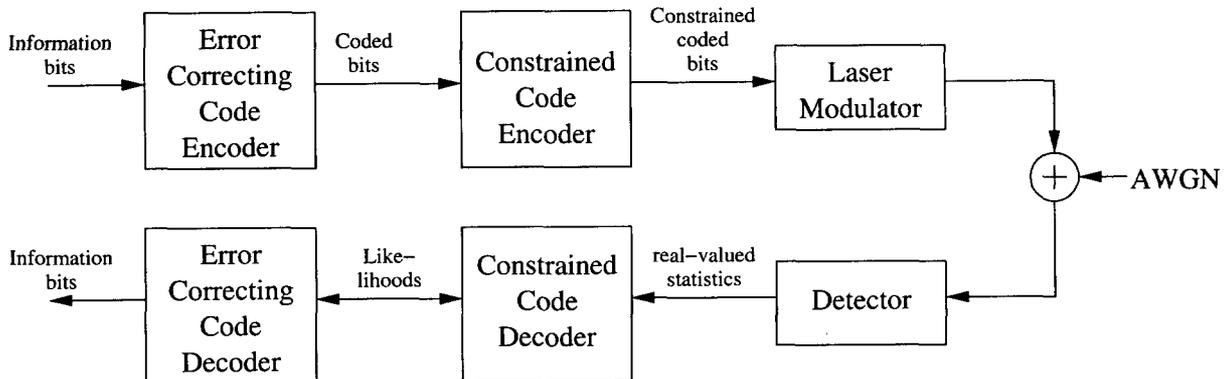


Figure 1: The communications system considered in this article.

between pulses. In the transmitted binary sequence, this corresponds to the constraint that no more than $k \stackrel{\text{def}}{=} \lceil T_k/T_s \rceil$ zeros occur between ones. Together, the two constraints are referred to as a (d, k) constraint and the collection of sequences satisfying the constraint as a (d, k) constrained system. An invertible mapping of unconstrained binary sequences into the (d, k) -system is referred to as a constrained code [8, 9]. (Although it may be advantageous to violate the dead-time constraint, sending lower SNR pulses at shorter intervals, in this work it is assumed the dead-time T_d is a hard constraint, i.e., pulses must be separated by T_d seconds.) The constrained code encoder in Fig. 1 makes sure the (d, k) constraint is satisfied.

At the receiver, light is focused on a detector. Depending on the type of detector used, the detector output can be either discrete or continuous. For example, the output of a photon counting detector is the number of detected photons, according to a Poisson distribution. In most detectors—including photo-multiplier tubes (PMT's), Avalanche Photo-Diode (APD) detectors, and even coherent detectors—the output is a real-valued voltage or current that arises from the detector input as well as from random processes within the detector and follow-on circuitry. These effects may be modeled in a variety of ways: the Poisson model is often used for PMT's, although a more accurate model is known in that case [10]; a Gaussian, Webb, or Webb plus Gaussian model can be used for APD's [11]; and a Gaussian model is best for a coherent detector.

Throughout this paper, we shall use a Gaussian model for statistics called the AWGN-1 model [11], in which the slot statistics at the output of the detector are independent and of the form $y = s + n$, where $s \in \{0, 1\}$ is the binary symbol transmitted, and n is zero-mean Gaussian noise with variance $\sigma^2 = N_0/2$. The symbol energy is $E_s = E[s^2]$, so that $E_s/N_0 = E[s^2]/(2\sigma^2)$. When used with a rate R_c bits/slot code, $E_b = E_s/R_c$, and

$$\frac{E_b}{N_0} = \frac{E_s}{R_c N_0} = \frac{E[s^2]}{2R_c \sigma^2}.$$

We rely on the AWGN-1 channel model and report results as a function of the bit SNR E_b/N_0 due to its simplicity, and the fact that all the above mentioned channel models behave in a way that is largely dominated by a bit SNR parameter [11] analogous to E_b/N_0 . Hence, we expect coding results presented here will apply

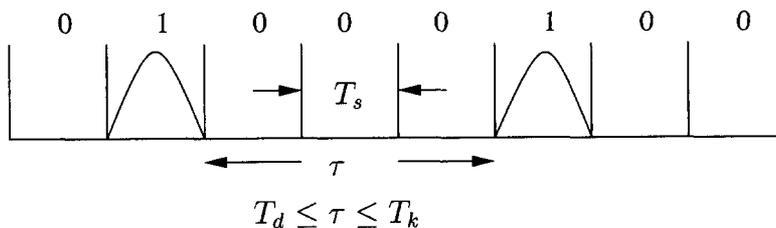


Figure 2. The optical channel is constrained on-off keying. A one represents a pulsed slot, and a zero a non-pulsed slot. There is at least T_d and at most T_k seconds between pulsed slots.

to a wide variety of channel models, in the sense that the relative coding gains of the various schemes will be about the same under different channel models and operating points.

Throughout this article, we compare the new schemes against a baseline scheme in which, in Fig. 1, the error correcting code is a Reed-Solomon code, and the constrained code is PPM with added dead-time.

3. THROUGHPUT OF THE CONSTRAINED CODE

Taking the k constraint as a design parameter, rather than as a hard constraint, we will investigate the achievable rates of constrained codes into the (d, ∞) constraint. Fig. 3 illustrates a graph presenting the (d, ∞) constrained system, where 0^x denotes a string of x zeros. The system is the set of sequences obtained by reading the labels of paths on the graph. The *capacity* of the (d, ∞) system when used on an error-free channel,

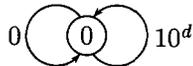


Figure 3: Graph presenting the (d, ∞) system.

$$C(d) \stackrel{\text{def}}{=} \lim_{n \rightarrow \infty} \frac{1}{n} \log_2 |\text{words of length } n \text{ in the } (d, \infty) \text{ system}| \text{ bits/slot}, \quad (1)$$

is the asymptotic growth rate of the number of distinct words, i.e., finite-length patterns, in the system and the least upper bound on the rate of a code into the system. From [12], we have $C(d) = \ln(\lambda)$ nats/slot, where λ is the largest positive root of

$$\lambda^{-(d+1)} + \lambda^{-1} - 1 = 0. \quad (2)$$

For small d exact solutions may be found efficiently for (2). For large d substitute $\lambda = e^{C(d)}$ and use the approximation $e^{-C(d)} \approx 1 - C(d)$ which yields

$$d + 1 \approx (d + 1)C(d)e^{(d+1)C(d)},$$

or

$$C(d) \approx \frac{W(d+1)}{T_s(d+1) \ln 2} \text{ bits/s}, \quad (3)$$

where $W(z)$ is the *productlog* function which gives the solution for w in $z = we^w$. Table 1 lists capacities for a range of d .

With $R_C(d)$ denoting the rate of a constrained code \mathcal{C} into the (d, ∞) system, $E_{C(d)} \stackrel{\text{def}}{=} R_C(d)/C(d)$ is the relative *efficiency* of the code, measuring how close the code rate is to the limit. There are well-known techniques to construct codes into a constrained system at rate arbitrarily close to capacity, e.g., [8, 9]. However, for our parameter range, a straight-forward application of these approaches may be prohibitively complex. In the following sections we present some approaches that trade off efficiency and complexity.

3.1. Pulse position modulation with added dead-time

First consider the efficiency of what will be considered the baseline, an M -slot PPM frame followed by a d -slot dead-time. A graph and tree presenting the allowable PPM code sequences with a dead-time constraint are illustrated in Fig. 4. Allowable sequences are read off the graph as described above. Code sequences on the tree are generated by traversing the tree. Considering PPM with added dead-time as a (d, ∞) constrained code, the rate is

$$R_{\text{PPM}}(d, M) = \frac{\log_2(M)}{T_s(M + d)} \text{ bits/s}.$$

For a given value of d , substituting the argument M that maximizes $R_{\text{PPM}}(d, M)$ yields

$$R_{\text{PPM}}(d) = \frac{W(d/e)}{T_s d \ln 2} \text{ bits/s}$$

d	Capacity (bits/s)	E_{PPM}	E_{STPPM}
1	.6942	.5787	.7202
2	.5515	.6057	.6045
4	.4057	.6382	.8217
8	.2788	.6729	.7175
16	.1813	.7069	.7880
32	.1130	.7380	.8847
64	6.785×10^{-1}	.7702	.8670
128	4.008×10^{-2}	.7921	.9242
256	2.319×10^{-2}	.8116	.9375
512	1.320×10^{-2}	.8286	.9352
1024	7.418×10^{-3}	.8434	.9427
2048	4.124×10^{-3}	.8562	.9435

Table 1: Capacity of (d, ∞) constrained codes and relative efficiencies of some particular schemes

We allow non-integer M in analysis to simplify expressions, since rounding has a negligible affect on rate for large d . One can show $E_{\text{PPM}}(d) \rightarrow 1$ as $d \rightarrow \infty$, i.e., PPM achieves capacity in the limit of large d . However, for d in our range of interest, significant gains in throughput over PPM are available.

3.2. Synchronous Variable-Length Codes

PPM is a fixed-rate code. Allowing a variable rate adds a degree of freedom in design, resulting in higher efficiency and/or lower complexity encoders. However, variable rate encoding and decoding has practical drawbacks. A compromise is to allow a synchronous rate, namely mappings of mp bits to mq bits, where p, q are fixed, positive integers, and m is a positive integer that can vary. Methods of constructing synchronous variable-length codes were initially described in [13], and reviews of various approaches may be found in [9, 14].

We describe a new systematic procedure to construct synchronous encoders and decoders for (d, ∞) constraints. Choose a rate $p/q < T_s C(d)$ bits/slot. We desire a set of variable-length codewords $\mathcal{C} = \{c_1, c_2, \dots, c_N\}$ such that any sequence formed by freely concatenating the codewords satisfies the constraint, the codeword lengths $l(c_i)$ are multiples of q , no codeword is the prefix of another (sufficient but not necessary to guarantee decodability), and the collection satisfies the Kraft (In)equality

$$\sum_{c_i \in \mathcal{C}} 2^{-l(c_i)p/q} = 1. \quad (4)$$

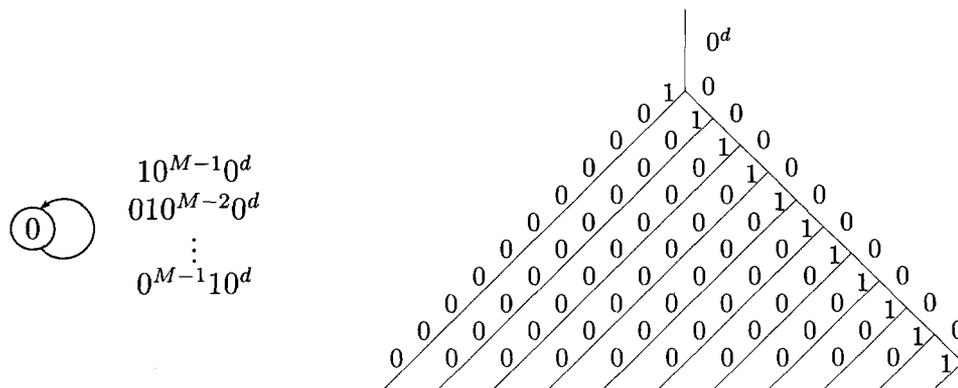


Figure 4: Graph and tree presenting PPM code sequences.

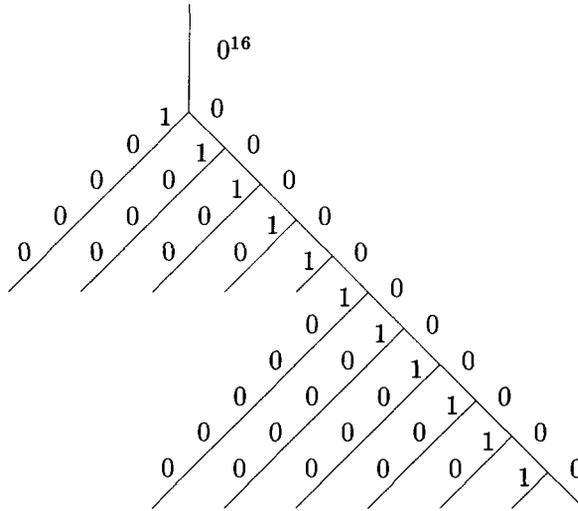


Figure 5: Construction for $(d, k) = (16, \infty)$

We can use such a set to construct a synchronous variable-length code mapping unconstrained binary sequences into the constraint.

We detail one method to construct such a set that leads to a low-complexity encoder and decoder. The codewords are constructed as nodes on a binary tree. The root of the tree is the pattern 0^d . Branches with a label 1 are extended with zeros to the first length that is a multiple of q . At this point, the branch label is taken as a codeword. The tree is expanded until we have a set of codewords that satisfies (4). Fig. 5 illustrates the procedure for the $(d, k) = (16, \infty)$ constraint with $p = 1, q = 7$. The all zeros pattern is not allowed as a codeword, since allowing it reduces the minimum Euclidean distance from 2 to 1, the small gain in throughput does not offset the loss in distance (allowing the all zeros codeword does yield significant throughput gains for small d), and a finite k is desired for synchronization. The encoding and decoding may be done at a fixed rate by using encoders and decoders with appropriate memory. Codes constructed via this method will be referred to as Synchronous Truncated Pulse Position Modulation (STPPM). A simple encoder implementation exists if we allow variable-out-degree states.

This procedure does not allow rates arbitrarily close to capacity. One can show a rate p/q encoder may be constructed via this method into a (d, ∞) constraint if

$$K(q, d, p) \geq 1,$$

where

$$K(q, d, p) = 2^{-lp} \left(lq - d + \frac{q-1}{2^p - 1} \right),$$

and $l = \lfloor d/q \rfloor + 1$. A simple encoder/decoder trellis may be constructed if states with variable-out-degrees are allowed. An encoder with variable-out-degrees states exists with

$$m = \left\lceil \frac{1}{p} \log_2 \left(\frac{q-1}{(K(q, d, p) - 1)(2^p - 1)} \right) \right\rceil$$

states and no more than

$$mq + l - 1 - d$$

edges. Table 2 lists the parameters of a number of codes for a range of d of interest where in each case we take $p = 1$. The encoder/decoder complexity may be traded off for efficiency in a systematic manner by specifying a lower rate. There are fewer than $q + 1$ distinct edge labels in each trellis stage.

d	q	states	edges	E_{STPPM}
16	7	4	14	.788
32	10	8	51	.885
64	17	6	41	.867
128	27	10	146	.924
256	46	12	301	.938
512	81	11	385	.935

Table 2: STPPM Variable-out-degree Encoder Parameters

The variable-out-degree trellis may also be used without modification to form a maximum-likelihood (ML) estimate via, for example, the Viterbi Algorithm. Decisions would be delayed, however typically no longer than existing delay due to the truncation depth. On the other hand, certain modifications will be necessary to form a maximum a posteriori (MAP) estimate via, for example, the BCJR algorithm.

3.3. Comparisons

Fig. 6 illustrates the efficiencies of fixed-orders of PPM and STPPM as a function of d . The PPM order begins with 2 and increases in powers of two to 256. The efficiency is measured relative to a (d, ∞) constraint. However, the codes all impose a maximum run-length constraint—necessary for timing recovery and desired for distance properties (a fairer measure of efficiency would be relative to the appropriate (d, k) constraint). As noted before, the efficiencies of the schemes—when allowed to choose optimal order—will approach 1 for large d . Note the larger gains over PPM that are possible for smaller values of d , and the dependence of PPM order on d .

As seen in Fig. 7, the STPPM codes demonstrate throughput (measured in bits per second) gains of 11%–17% over PPM. However, this may come at the cost of higher complexity, as seen in Fig. 8, and of reducing the number of bits transmitted per pulse. Fig. 9 illustrates the performance of a maximum-likelihood decoded STPPM code relative to 8-PPM and 16-PPM for $d = 16$. The minimum distance of the three codes is the same, and the throughput of the PPM schemes are the same, whereas STPPM has a 14% higher throughput. The performance is differentiated due to the energy per bit requirements. STPPM transmits an average 3.375 bits per pulse whereas 16-PPM transmits 4 bits per pulse—yielding a net gain relative to STPPM of $\approx .74\text{dB}$ —and 8-PPM transmits 3 bits per pulse—for a net loss relative to STPPM of $\approx .5\text{dB}$.

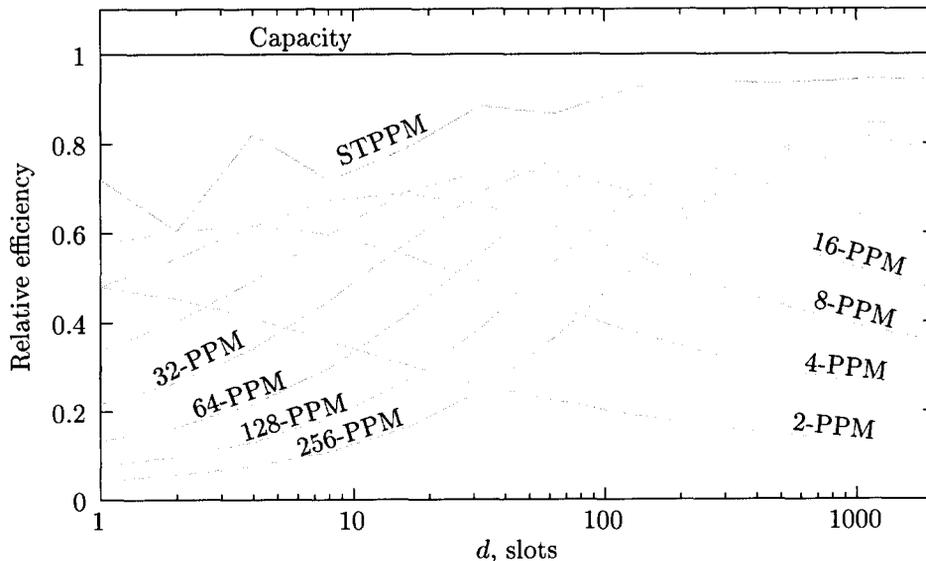


Figure 6: Relative efficiency of PPM for various orders, and STPPM

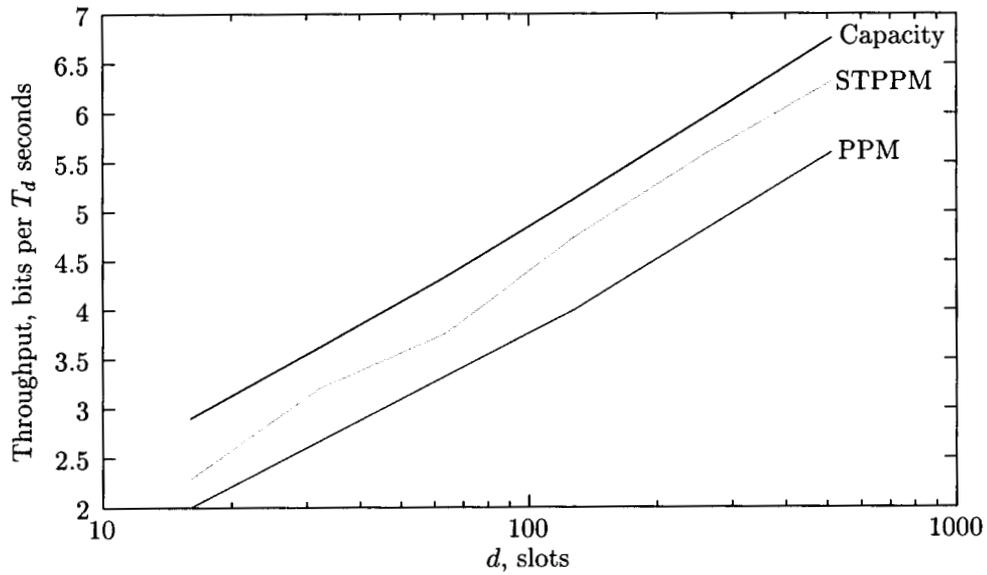


Figure 7: Throughput

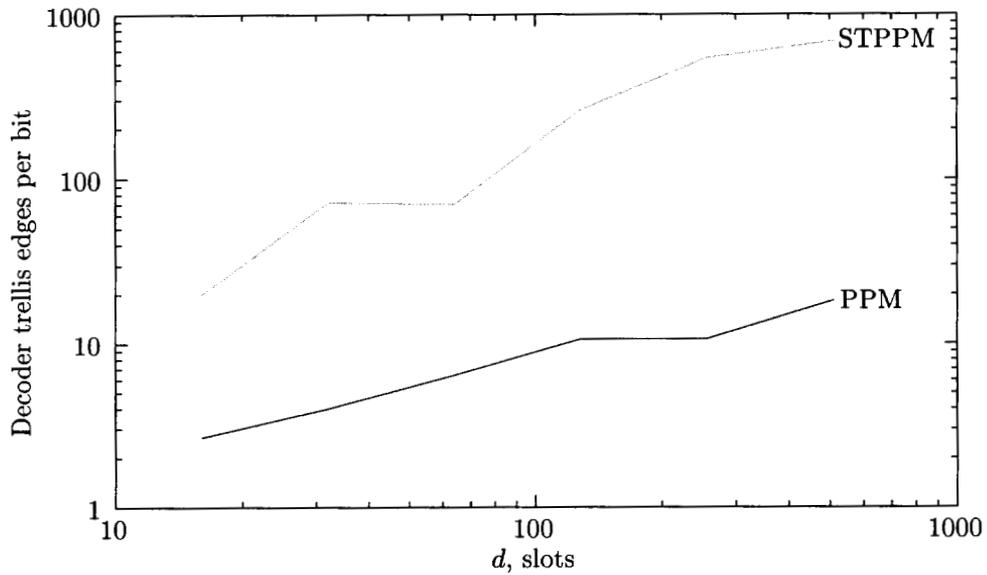


Figure 8: Complexity

Any sequences in a (d, ∞) system may be uniquely parsed into patterns, or phrases, $0^j 1$ where $j \geq d$. The phrases $0^j 1$ are referred to as *run-lengths* and their distribution will have an impact on slot-synchronization schemes. Fig. 10 illustrates the distribution of run-lengths for 8-PPM, 16-PPM, one implementation of a STPPM code, and the capacity-achieving distribution at $d = 16$.

4. THE CONSTRAINED CODE IN A CONCATENATED CODING SCHEME

The larger coding structure will use the constrained code in concert with an Error Correcting Code (ECC). The codes will be concatenated serially, as illustrated in Fig. 1. A bit interleaver may be inserted between the codes, serving to disperse error bursts in decoding, and providing particular performance improvement in iterative decoding schemes. We will use the notation $\mathcal{C}_o \rightarrow \mathcal{C}_i$ to denote the non-iteratively decoded serial concatenation

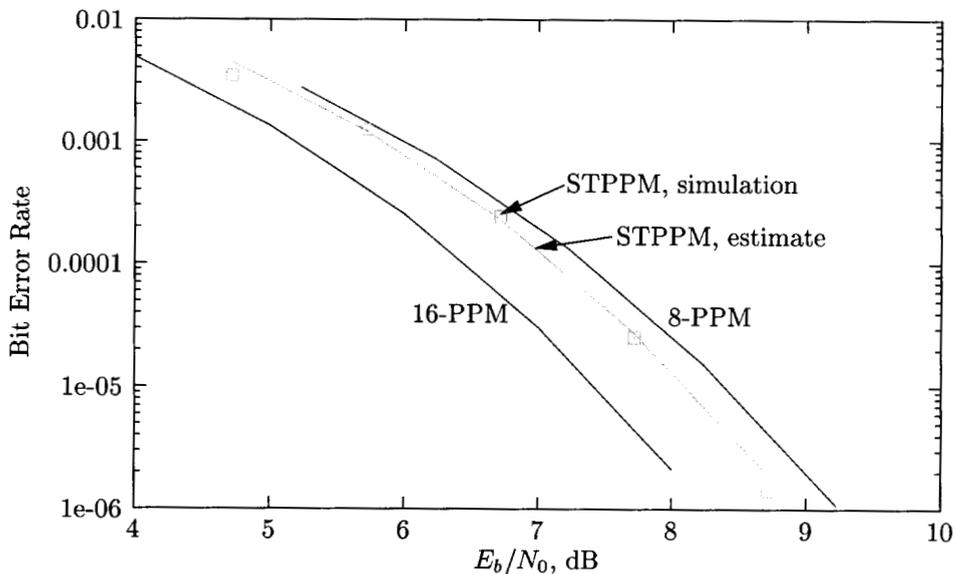


Figure 9: STPPM, 16-PPM, and 8-PPM, $d = 16$

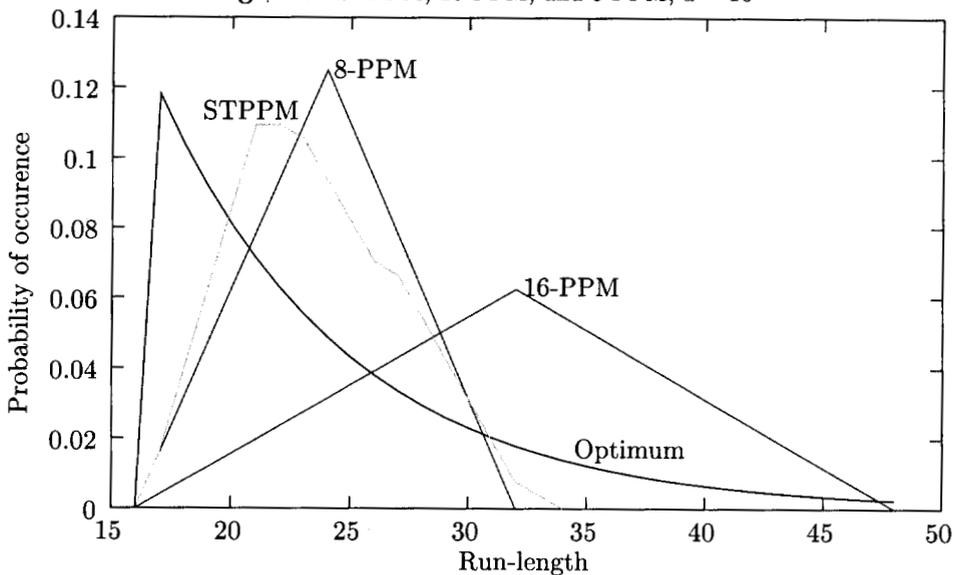


Figure 10: Runlengths distributions, $d = 16$

of outer code \mathcal{C}_o and inner code \mathcal{C}_i . Iteratively decoded codes are denoted by $\mathcal{C}_o \leftrightarrow \mathcal{C}_i$. Iterative decoding follows the description in [15].

The baseline system is taken to be $\text{RS}(M-1, k) \rightarrow M\text{-PPM}$, where $\text{RS}(M-1, k)$ denotes a rate $k/(M-1)$ Reed-Solomon code and the PPM demodulator produces hard-decisions. (Here, k does not refer to the runlength constraint.) Prior work investigated the system $\text{PCCC} \rightarrow M\text{-PPM}$ [16], where PCCC is an iteratively decoded Parallel Concatenated Convolutional Code and the PPM demodulator produces soft-decisions—although it is not included in iterations. Peleg and Shamai [17] investigated the system $\text{PCCC} \leftrightarrow M\text{-PPM}$ on a discrete-time Rayleigh-fading model, illustrating performance 1–2 dB from capacity.

4.1. Simulation Results

Fig. 11 illustrates performance under a $d = 16$ constraint. Uncoded 16-PPM satisfying $(d, k) = (16, 46)$, a rate

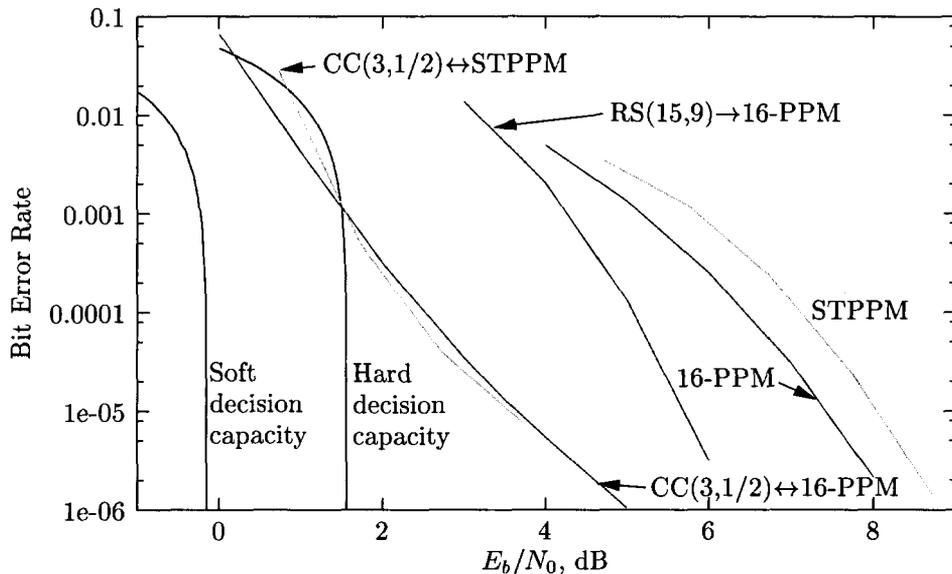


Figure 11: BER performance comparisons, $d = 16$.

1/7 STPPM code satisfying $(d, k) = (16, 32)$, $RS(15, 9) \rightarrow 16\text{-PPM}$, $CC(3, 1/2) \leftrightarrow 16\text{-PPM}$ and $CC(3, 1/2) \leftrightarrow STPPM$ are illustrated, where $CC(3, 1/2)$ is the 4-state convolutional code with generator polynomial $g(D) = [1(1 + D^2)/(1 + D + D^2)]$. The capacity limits for throughput (1/14) bits/slot on a hard and soft-decision channel constrained to use a 16-ary orthogonal signal set, e.g., 16-PPM are also illustrated. These curves represent the theoretical limits of a channel using 16-PPM with hard and soft decisions, respectively.

Both $CC(3, 1/2) \leftrightarrow 16\text{-PPM}$ and $CC(3, 1/2) \leftrightarrow STPPM$ used a 512-bit interleaver and 8 iterations. They illustrate gains of $\approx 2.5\text{dB}$ over $RS(15, 9) \rightarrow 16\text{-PPM}$ at a bit-error-rate of 10^{-5} . Simulation results demonstrate that 4 iterations would be sufficient at the higher values of SNR. A small additional gain of approximately 0.2dB was found for a 4096-bit interleaver with $CC(3, 1/2) \leftrightarrow 16\text{-PPM}$. Note that $CC(3, 1/2) \leftrightarrow STPPM$ has a higher throughput (1/14 bits/slot) than $CC(3, 1/2) \leftrightarrow 16\text{-PPM}$ (1/16 bits/slot).

Fig. 12 illustrates performance under a $d = 1024$ constraint. Uncoded 256-PPM satisfying $(d, k) = (1024, 1534)$, $RS(255, 128) \rightarrow 256\text{-PPM}$, and $CC(3, 1/2) \leftrightarrow 256\text{-PPM}$ for two interleaver sizes are illustrated. The capacity limits for throughput 1/320 bits/slot on a hard and soft-decision channel constrained to use a 256-ary orthogonal signal set are also illustrated. $CC(3, 1/2) \leftrightarrow 256\text{-PPM}$ with a 4096-bit interleaver and 8 iterations shows gains of 2.3dB over $RS(255, 128) \rightarrow 256\text{-PPM}$ at a bit-error-rate of 10^{-5} . $CC(3, 1/2) \leftrightarrow 256\text{-PPM}$ performs 0.4dB better than *any* system with the same throughput that uses hard-decision 256-PPM.

These results are surprising in light of the low-complexity of the constituent codes and lack of recursiveness in the inner code. They provide a strong argument for replacing the baseline $RS(M - 1, k) \rightarrow M\text{-PPM}$ with a low-complexity ECC serially-concatenated with PPM or some other constrained code.

5. CONCLUSIONS/FUTURE WORK

There are certain trade-offs in replacing PPM with STPPM or another constrained code. The constrained codes considered provide higher throughput at the cost of increased complexity. Whether the code gains in energy per transmitted bit and run length distribution depends on implemented parameters.

The gains of the concatenated, iteratively decoded schemes over the baseline $RS \rightarrow PPM$ are more clear. We have illustrated that low-complexity iterative schemes provide significant gains over the baseline. We expect to improve on these gains with a better understanding of the interaction between the outer code and constrained code. For example, recent results that include an accumulator—a $1/(1 + D)$ mapping—prior to the PPM mapping in order to add recursiveness to the mapping have shown significant additional gains for small d .

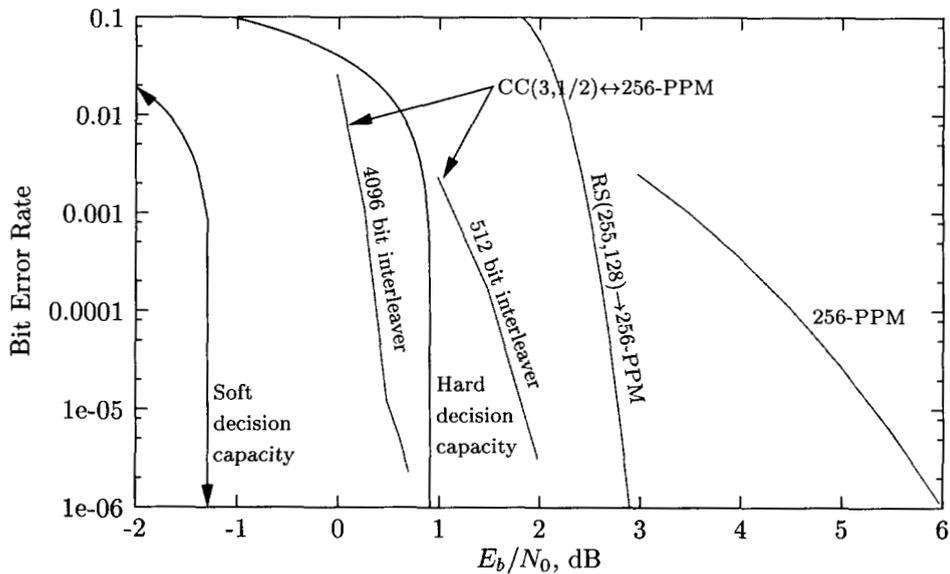


Figure 12: BER performance comparisons, $d = 1024$.

We conjecture that the interleaver in the serial concatenated system need only be large enough to distribute each bit in the most likely error bursts from the outer code into distinct PPM symbols. We expect that interleavers larger than this will show only small improvements.

The iteratively decoded schemes do, however, rely on statistics from each signal slot. This may be unfeasible at the proposed operating rates. Future work will investigate the degradation in performance when approximations to complete slot statistics are used.

REFERENCES

1. R. G. Lipes, "Pulse-position-modulation coding as near-optimum utilization of photon counting channel with bandwidth and power constraints," *DSN Progress Report* **42**, pp. 108–113, Apr. 1980.
2. A. D. Wyner, "Capacity and error exponent for the direct detection photon channel—Part I," *IEEE Transactions on Information Theory* **34**, pp. 1449–1461, Nov. 1988.
3. J. R. Pierce, "Optical channels: Practical limits with photon counting," *IEEE Transactions on Communications* **COM-26**, pp. 1819–1821, Dec. 1978.
4. S. A. Butman, J. Katz, and J. R. Lesh, "Bandwidth limitations on noiseless optical channel capacity," *IEEE Transactions on Communications* **COM-30**, pp. 1262–1264, May 1982.
5. H. H. Tan, "Capacity of a multimode direct detection optical communication channel," *TDA Progress Report* **42-63**, pp. 51–70, May 1981.
6. G. G. Ortiz and A. Biswas, "Design of an opto-electronic receiver for deep-space optical communications," *TMO Progress Report* **42-142**, pp. 1–17, Aug. 2000.
7. M. Srinivasan, J. Hamkins, B. Madden-Woods, A. Biswas, and J. Beebe, "Laboratory characterization of silicon avalanche photodiodes (APDs) for pulse-position modulation (PPM) detection," *TMO Progress Report* **42-142**, pp. 1–14, Aug. 2000.
8. K. A. S. Immink, P. H. Siegel, and J. K. Wolf, "Codes for digital recorders," *IEEE Transactions on Information Theory* **44**, pp. 2260–2299, Oct. 1998.
9. B. H. Marcus, R. M. Roth, and P. H. Siegel, "Constrained systems and coding for recording channels," in *Handbook of Coding Theory*, V. S. Pless and W. C. Huffman, eds., ch. 20, Elsevier Science, 1998.
10. H. H. Tan, "A statistical model of the photomultiplier gain process with applications to optical pulse detection," *TDA Progress Report* **42-68**, pp. 55–67, Apr. 1982.

11. S. Dolinar, D. Divsalar, J. Hamkins, and F. Pollara, "Capacity of PPM on Gaussian and Webb channels," *TMO Progress Report*, Aug. 2000.
12. C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal* **27**, pp. 379–423, July 1948.
13. P. A. Franzaszek, "Sequence-state coding for digital transmission," *The Bell System Technical Journal* **47**, pp. 143–157, 1968.
14. K. A. S. Immink, *Codes for Mass Data Storage Systems*, Shannon Foundation Publishers, The Netherlands, 1999.
15. S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A soft-input soft-output maximum a posteriori (MAP) module to decode parallel and serial concatenated codes," *TDA Progress Report* **42-127**, pp. 1–20, 1996.
16. J. Hamkins, "Performance of binary turbo coded 256-ppm," *TMO Progress Report* **42**, pp. 1–15, Aug. 1999.
17. M. Peleg and S. Shamai, "Efficient communication over the discrete-time memoryless Rayleigh fading channel with turbo coding/decoding," *European Transactions on Telecommunications* **11**, pp. 475–485, September-October 2000.