EXIT Function Aided Design of Iteratively Decodable Codes for the Poisson PPM Channel

Maged F. Barsoum, Bruce Moision, Michael P. Fitz, *Senior Member, IEEE*, Dariush Divsalar, *Fellow, IEEE*, and Jon Hamkins, *Senior Member, IEEE*

***Abstract*—This paper presents and compares two iterative coded modulation techniques for deep-space optical communications using pulse-position modulation (PPM). The first code, denoted by SCPPM, consists of the serial concatenation of an outer convolutional code, an interleaver, a bit accumulator, and PPM. The second code, denoted by LDPC-PPM, consists of the serial concatenation of an LDPC code and PPM. We employ Extrinsic Information Transfer (EXIT) charts for their analysis and design. Under conditions typical of a communications link from Mars to Earth, SCPPM is 1 dB away from capacity, while LDPC-PPM is 1.4 dB away from capacity, at a Bit Error Rate (BER) of approximately 10^{-5}. However, LDPC-PPM lends itself naturally to low latency parallel processing in contrast to SCPPM.**

***Index Terms*—Error control coding, pulse position modulation, optical modulation, optical communication.**

I. Introduction

In this paper we examine the design of error-correction-codes (ECCs) for the pulse-position-modulated (PPM) Poisson channel. This channel models a direct-detection deep-space optical communications link [1]. The codes described here were designed to support the (since canceled) Mars Laser Communications Demonstration (MLCD) [2]. MLCD was to be an Earth-Mars optical communications downlink with a laser transmitter orbiting Mars and a ground receiver (telescope) on Earth. The asymmetry of the link resources, typical of a deep-space link, supported the development of ECCs that had a low encoding complexity but could allow a high decoding complexity–a good candidate for use of modern iterative ECCs.

The received signal for a direct-detection optical channel is well modeled as a Poisson point process. Wyner [3] showed that under peak and average power constraints, a negligible loss in capacity is incurred when restricting the modulation to a binary, slotted scheme–that is, to on-off-keying (OOK). Shamai [4] extended this result to include a bandwidth constraint, illustrating regions where a binary, slotted scheme is near-optimal. One can show that the OOK capacity limit as the slotwidth tends to zero is closely approached by OOK capacity with a small finite slotwidth [5], [6]. For slotwidths that are both practical and of interest, there is little loss in restricting the input to OOK. Furthermore, when the optimum duty cycle, the ratio of pulsed to non-pulsed slots, is small, there is little loss in restricting the modulation to PPM [7]. Other approaches to implement a small duty cycle, such as constrained coding [8], or shaping codes [9], have demonstrated little or no gain over PPM at small duty cycles.

This leads to the following channel model. In each channel use a block of M slots is transmitted, with one of the M slots pulsed. The conditional probability mass functions of the pulsed and non-pulsed slots are Poisson, with mean $n_s + n_b$ in a pulsed slot and mean n_b in a noise slot. The optimum PPM order (the capacity maximizing order) under peak power, average power and bandwidth constraints dictated by recent link budgets for a Mars-Earth downlink is on the order of $M = 64$, and the typical received noise power is $n_b = 0.2$ photons/slot [7], [10]. We will use these nominal values as a design point throughout this paper. However, note that the code performance, and design, would depend in general on the pair (n_s, n_b).

Pierce [11] illustrated that photon efficiencies for the Poisson channel, that is, achievable signal-photons per bit, are limited only by practical constraints on implementing low duty cycle coded modulation. This observation spurred the development of efficient ECCs for the optical channel. Early work focused on the noiseless ($n_b = 0$) channel. The earliest ECCs proposed for the optical channel were Reed-Solomon codes, with the symbols drawn from the M-element Galois field, matching the PPM order [12], [13]. An (n, k) Reed-Solomon (RS) code, which maps k symbols to n symbols, is a maximum-distance-separable code, that is, it has the largest possible minimum distance for any (n, k) block code. It follows that an (n, k) RS code provides the optimal performance for any (n, k) code on the *symbol-erasure channel*, which occurs for $n_b = 0$, since it can correct up to $n - k$ symbol erasures, the maximum for any (n, k) code. However, the blocklength dictated by choosing $n = M - 1$ is restrictive. This can be extended by constructing RS code symbols by combining multiple PPM symbols [6], [14], although this yields only modest improvements. An extension to the noisy

Paper approved by K. Narayanan, the Editor for Optical Communication of the IEEE Communications Society. Manuscript received December 9, 2006; revised August 11, 2008.

This work was funded by the IND Technology Program and performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. The material in this paper was presented in part at the 2007 IEEE Information Theory Workshop, Lake Tahoe, California, September 2007.

M. F. Barsoum is with Broadcom Corporation, San Jose, CA, USA (e-mail: maged.barsoum@engineering.ucla.edu).

B. Moision, D. Divsalar, and J. Hamkins are with the Information Processing Group, Jet Propulsion Laboratory, Pasadena, CA, USA (e-mail: {bmoision, dariush.divsalar, jon.hamkins}@jpl.nasa.gov).

M. P. Fitz is with Northrop Grumman Space Technology, Redondo Beach, CA 90278 USA, and also with the University of California, Los Angeles, CA 90095 USA (e-mail: michael.fitz@ngc.com).

Digital Object Identifier 10.1109/TCOMM.2010.110310.060572

channel [15] allowed for both erasures and errors. On the noisy ($n_b > 0$) channel, RS-coded PPM with hard decision decoding provides performance typically $3 - 4$ dB from capacity (for parameters of interest). Although soft-decision decoding, e.g., [16], [17], would reduce the gap for the noisy channel, we do not investigate soft-decision-decoded RS PPM performance here, taking the hard-decision performance as a baseline.

Massey [18] used the problem of coding for the noiseless Poisson PPM channel to address some of the pressing questions of the day in the coding community: whether the cutoff-rate or the capacity provides a better metric of achievable performance, and whether block or convolutional codes provided better performance for the same complexity. Noting that the M-ary erasure channel is equivalent to the parallel combination of $\log_2 M$ binary erasure channels, Massey proposed coding each of the $\log_2 M$ bits mapping to each symbol coded with separate convolutional codes, illustrating performance competitive with RS coded PPM.

The advent of turbo codes [19] brought about the development of parallel and serially concatenated codes for the PPM channel. Parallel-concatenated-codes (PCCs) for binary PPM were investigated in [20], [21], and for higher order PPM in [22] and [23], which also developed serially concatenated codes. These codes illustrated gains over RS coded PPM, although they did not capitalize on iterative demodulation, which provides significant improvement at high orders. In [24] iterative demodulation of PPM with a PCC was investigated for use on the discrete-time Rayleigh fading channel, where low duty cycles are also optimum. They demonstrated performance 1–2 dB from capacity.

In this paper we illustrate the design of two iterative codes for the Poisson PPM channel: a low-density-parity-check (LDPC) code and a serially-concatenated-convolutional (SCC) code. These codes are representatives of two leading classes of iterative ECCs. Each incorporates iterative PPM demodulation as part of decoding, and the code design takes the iterative demodulation into account. This work is distinguished from prior work in that we focus on designing the codes to be compatible with high order PPM, that is, to utilize PPM as part of the code structure. In so doing, we provide codes with performance close to capacity with moderate decoding complexity.

The remainder of this paper is organized as follows. In Section II we describe the channel model. In Section III we find the Extrinsic Information Transfer (EXIT) functions for PPM and accumulate-PPM, which are instrumental in code design. In Section IV, we describe the first code that uses APPM (to be defined). In Section V, we describe the LDPC code. In Section VI, we discuss the complexity and performance of both codes.

Notation is as follows. Lowercase u, w, y, x denote realizations of the corresponding random variables U, W, Y, X. Boldface $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ and $\mathbf{U} = (U_1, \ldots, U_n)$ denote vectors. $\mathbf{w}_{[k]}$ denotes the vector $\mathbf{w}$ with element w_k removed. Where clear or irrelevant the subscript k of an element u_k from vector $\mathbf{u}$ may be dropped for simplicity. The notation $p_Y(y)$ is used to denote the probability density or mass function of random variable Y evaluated at y. When the random variable is clear from the context, we simply write $p(y)$ for $p_Y(y)$.

II. Channel Model

For the purpose of ECC design, the channel model used throughout this paper is defined as follows. Binary symbols $\mathbf{c}$ are transmitted over the optical channel and received as $\mathbf{y}$. The channel $p(y|c)$ is modeled as a binary-input Poisson channel. In any time slot, either a pulse is transmitted to send a 1, or no pulse is transmitted to send a 0. Letting n_s be the mean received signal photons per pulsed slot and n_b the mean received noise photons per slot, we have

$$p_0(y) \triangleq p_{Y|C}(y|0) = \frac{e^{-n_b} {n_b}^y}{y!} \quad (1)$$

$$p_1(y) \triangleq p_{Y|C}(y|1) = \frac{e^{-(n_b+n_s)} (n_b + n_s)^y}{y!} \quad (2)$$

Channel uses are assumed to be conditionally independent, i.e.,

$$p(\mathbf{y}|\mathbf{c}) = \prod_k p(y_k|c_k) \quad (3)$$

and bits are modulated using M-ary PPM. In PPM, each $\log_2 M$ bits map to the location of a single pulsed slot in an M-slot frame. Performance of coded systems will be measured relative to the capacity of a PPM modulated channel, given by [6]:

$$C = \frac{1}{M} E_{Y_1,\ldots,Y_M} \log_2 \left[\frac{M L(Y_1)}{\sum_{j=1}^{M} L(Y_j)} \right] \text{ bits/slot}$$

where $L(y) = p_1(y)/p_0(y)$, Y_1 is distributed as p_1 and the $Y_j, j \neq 1$, are distributed as p_0. Substituting (1) and (2), for $n_b > 0$, this reduces to:

$$C = \log_2 M - E_{Y_1,\ldots,Y_M} \log_2 \left[\sum_{i=1}^{M} \left(1 + \frac{n_s}{n_b}\right)^{Y_j - Y_1} \right] \text{ bits/slot}$$

For $n_b = 0$, the expression for capacity simplifies further to:

$$C = (\log_2 M)\left(1 - e^{-n_s}\right) \text{ bits/slot}$$

The duty cycle of M-ary PPM is $1/M$. Link budgets for deep-space optical links show that the optimum duty cycle is less than $1/32$ for the entirety of a typical (in orbit) mission [6], [10]. As noted earlier, for these duty cycles, we see a negligible loss in capacity when choosing PPM.

III. PPM and APPM EXIT Curves

The performance of the serially concatenated code presented here is improved by adding memory to the PPM mapping. This is accomplished by precoding the binary input to the PPM mapping with a $1/(1 + D)$ binary accumulator. That is, an input binary sequence $\ldots, a_k, a_{k+1}, a_{k+2} \ldots$ is encoded to produce $b_k = a_k + b_{k-1}$, where addition is modulo-2. Consecutive $\log_2 M$-blocks of bits b_k are then mapped to PPM symbols. We refer to the mapping produced by concatenating the accumulator and PPM as APPM.

In this section we characterize PPM and APPM using extrinsic-information-transfer (EXIT) functions [25]–[27]. For this, we consider the following scenario. A block of bits $\mathbf{a}$ is encoded by a code $\mathcal{C}$ (either PPM or APPM) to yield the sequence of symbols $\mathbf{c}$. The sequence of symbols $\mathbf{c}$ is transmitted over the channel defined by equations (1-3) and received as $\mathbf{y}$.

The code is decoded via a soft-input-soft-output (SISO) algorithm (decoder). A SISO decoder receives, as soft inputs, noisy versions, or log-likelihoods, of the input and output of the encoder and produces updated log-likelihoods of the input, or output, or both. These likelihoods may then be transmitted to other SISO modules in the receiver where they are treated as noisy received inputs. Derivations of the SISO algorithm have appeared in various forms in the literature, e.g., [28], [29] and will not be presented here.

In our case, the SISO decoder of code $\mathcal{C}$ receives $\mathbf{y}$, a noisy version of $\mathbf{c}$ over the memoryless communications channel $p(y|c)$. We also think of it as receiving a sequence $\mathbf{w}$, a noisy version of $\mathbf{a}$ over a memoryless *extrinsic* channel $p(w|a)$. With the serial concatenation of an outer code with the inner code $\mathcal{C}$, the extrinsic channel models information coming from the outer decoder. The sequence $\mathbf{w}$ and channel $p(w|a)$ are artificial constructs introduced to aid in analysis of the decoder, as is done in [27]. The observations in $\mathbf{w}$ are assumed to be conditionally independent, i.e.,

$$p(w_k, w_j|\mathbf{a}, \mathbf{y}, \mathbf{w}_{[k],[j]}) = p(w_k|a_k)p(w_j|a_j)$$

Information from the noisy observations $\mathbf{w}$ are received by the SISO decoder as a priori log-likelihood ratios (LLRs)

$$l_k = \log \frac{P\left(A_k = 0|w_k\right)}{P\left(A_k = 1|w_k\right)}$$

from which the SISO algorithm computes, as a function of the a priori LLRs and the mapping of $\mathbf{a}$ to $\mathbf{c}$, extrinsic log-likelihoods

$$l_{e_k} = \log \frac{P\left(A_k = 0|\mathbf{y}, \mathbf{w}_{[k]}\right)}{P\left(A_k = 1|\mathbf{y}, \mathbf{w}_{[k]}\right)}$$

The LLRs

$$l_{p_k} = \log \frac{P\left(A_k = 0|\mathbf{y}, \mathbf{w}\right)}{P\left(A_k = 1|\mathbf{y}, \mathbf{w}\right)}$$

are referred to in the literature as the a posteriori information and are the basis for bit decisions. We see that these LLRs are related as

$$l_{e_k} = l_{p_k} - l_k$$

If $p(w|a)$ is an output symmetric channel, meaning that

$$p(w|A=1) = p(-w|A=0)$$

then as shown in [30] the conditional densities of the a priori LLRs are symmetric, i.e., they satisfy

$$p_{L|A}\left(-l|1\right) = p_{L|A}\left(l|1\right)e^{l} \qquad (4)$$

$$p_{L|A}\left(-l|0\right) = p_{L|A}\left(l|0\right)e^{-l} \qquad (5)$$

In EXIT analysis, $p_{L|A}$ is commonly assumed to be Gaussian. If the conditional LLRs are Gaussian and the channel is output symmetric, equations (4) and (5) imply that

$$E\left[L|A=1\right] = -Var\left[L|A=1\right]/2$$
$$E\left[L|A=0\right] = Var\left[L|A=0\right]/2$$

allowing a simple, single parameter characterization of the LLR statistics.

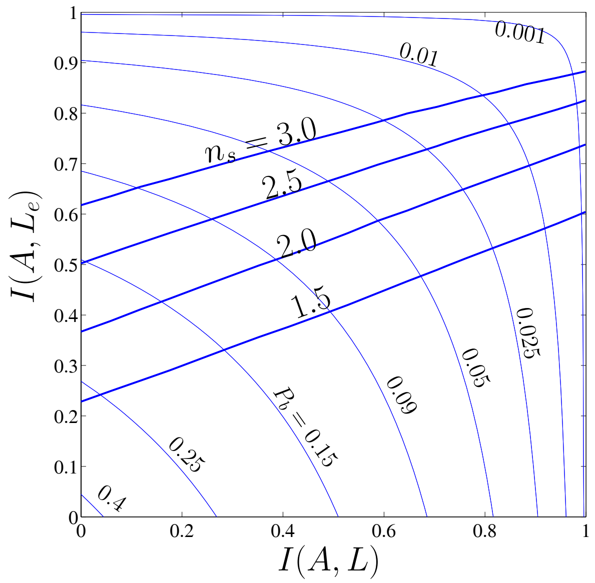


Fig. 1. Output mutual information vs input mutual information for the PPM SISO, $n_b = 0.2$, M=64.

Ten Brink [25], [26] proposed tracking the evolution of the mutual information between bits and their corresponding LLRs in order to predict the decoder behavior, which has proven to be a useful tool in code design. In our example, an extrinsic information transfer (EXIT) function is a plot of the mutual information $I(A; L_e)$ as a function of $I(A; L)$. For equiprobable binary A, the mutual information $I(A; L_e)$ may be expanded as

$$I(A; L_e) = \frac{1}{2} \sum_{a \in \{0,1\}} \int p_{L_e|A}(l|a) \log_2 \frac{2p_{L_e|A}(l|a)}{p_{L_e|A}(l|0) + p_{L_e|A}(l|1)} dl \qquad (6)$$

which may be evaluated via numerical integration using estimates of the densities $p_{L_e|A}$ obtained by simulation. To estimate $p_{L_e|A}$, we assume $p_{L|A}$ is Gaussian, and determine $p_{L_e|A}$ from samples of L_e generated by simulation, and assumed independent and identically distributed. A kernel density estimator along with trapezoidal numerical integration is used, treating the densities as continuous. This is seen to be more accurate than assuming that both L and L_e are Gaussian.

Fig. 1 and 2 show $I(A; L_e)$ versus $I(A; L)$ for PPM and APPM respectively with $M = 64$, $n_b = 0.2$, and $n_s \in \{1.5, 2.0, 2.5, 3.0\}$. Assuming a symmetric Gaussian distribution for both L and L_e, bit error rate contours are also shown from which the error rate in estimating $\mathbf{a}$ can be predicted at any combination of $I(A; L)$ and $I(A; L_e)$.

We note that APPM allows $I(A, L_e)$ to go to 1 as $I(A; L)$ approaches 1, whereas for PPM $I(A, L_e)$ remains below 1 at $I(A, L) = 1$. This comes at the cost of a slightly lower initial $I(A, L_e)$ for APPM compared to PPM, as can be noted by comparing Fig. 1 and Fig. 2.

If we assume that $p_{L_e|A}(l|1) = p_{L_e|A}(-l|0)$ and that $p_{L_e|A}$ is Gaussian, then $I(A; L_e)$ becomes a function of $Var[L_e|A]$

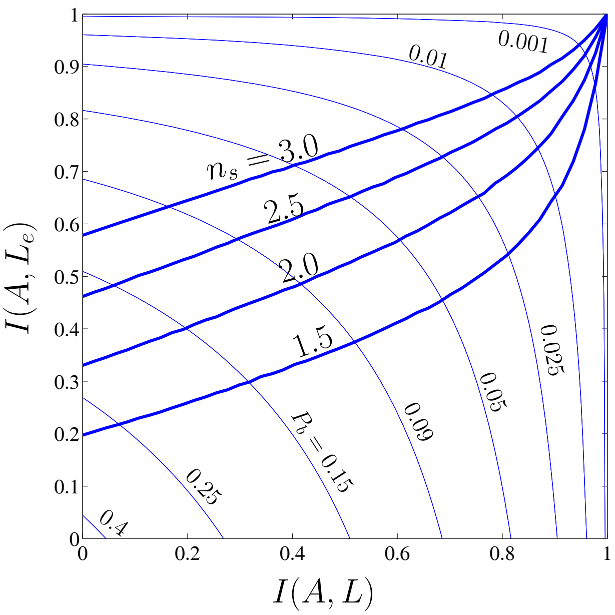


Fig. 2. Output mutual information vs input mutual information for the APPM SISO, $n_b = 0.2$, M=64.

only and (6) reduces to

$$\begin{aligned} I(A; L_e) &= 1 - E_{L_e|A=1}\left[\log_2\left(1 + e^{L_e}\right)\right] \\ &= 1 - E_{L_e|A=0}\left[\log_2\left(1 + e^{-L_e}\right)\right] \end{aligned} \quad (7)$$

This result will be used in the analysis of the following section.

A. Initial and End Points for the PPM EXIT Curve

As can be seen in Fig. 1, the PPM EXIT curves can be accurately modeled as linear for this range of n_s. To facilitate using a linear model, expressions for the initial and end points of the PPM EXIT curves will be derived in this section. From symmetry of the PPM constellation, the memoryless property of the communications channel, and under the assumption that the extrinsic channel is memoryless, the EXIT charts of each of the $\log_2 M$ bits mapping to a PPM symbol will be identical. Therefore, without loss of generality, we only treat the first bit of $\mathbf{a}$, a_1. We assume a natural mapping, where a zero in the first bit position maps to a PPM symbol with pulses in one of the first $M/2$ slots and a one in the first bit position maps to a PPM symbol with pulses in slots $M/2 + 1$ to M.

In this analysis, we treat the sequence l_k as deterministic and observe the limits for $l_k = 0$ and as $|l_k| \to \infty$, $\forall k$. On the initial iteration, the a priori information from the inner code will be all zeros ($l_k = 0$, $\forall k$). Hence the extrinsic output LLRs for a_1 on the first iteration will be

$$l_{e_1} = \log\left(\sum_{i=1}^{M/2}\left(1 + \frac{n_s}{n_b}\right)^{y_i}\right) - \log\left(\sum_{i=M/2+1}^{M}\left(1 + \frac{n_s}{n_b}\right)^{y_i}\right) \quad (8)$$

Using (8), we obtain an estimate of the initial mutual information by assuming that (7) holds:

$$\begin{aligned} I(A; L_e) &\approx 1 - E_{L_e|A=0}(\log_2(1 + e^{-L_e}) \\ &= 1 - E_{Y_1,\ldots,Y_M|A_1=0}\left[\log_2\left(\frac{\sum_{i=1}^{M}(1 + n_s/n_b)^{Y_i}}{\sum_{i=1}^{M/2}(1 + n_s/n_b)^{Y_i}}\right)\right] \end{aligned} \quad (9)$$

Equation (9) can be evaluated by generating samples of Y_1 distributed as p_1, and samples of $Y_j, j \neq 1$ distributed as p_0 (as defined by equations 1-3), and finding the sample mean of the above expression. This is found to give an accurate estimate of the initial mutual information, i.e., it agrees with the numerical evaluation of (6).

We will now derive the end point of the PPM EXIT curve, i.e., $I(A, L_e)$ when $I(A, L) = 1$. Assuming, without loss of generality, that the all-zeros bit pattern was transmitted, in the limit of large a priori information we have

$$\begin{aligned} l_{e_1} &\xrightarrow[|l_k| \to \infty \forall k]{} \frac{p(\mathbf{y}|c_1 = 1, c_i = 0, i = 1, \ldots, M)}{p(\mathbf{y}|c_{M/2+1} = 1, c_i = 0 \text{ for } i \neq M/2 + 1)} \\ &= (y_1 - y_{M/2+1})\log\left(1 + \frac{n_s}{n_b}\right) \end{aligned} \quad (10)$$

Under the assumption that (7) holds, we have

$$I(A; L_e) \approx 1 - E_{L_e|A=0}(\log_2(1 + e^{-L_e}) \xrightarrow[|l_k| \to \infty, \forall k]{} E_{Y_1, Y_{M/2+1}|A_k=0\forall k}\left[\log_2\left(1 + \left(1 + \frac{n_s}{n_b}\right)^{(Y_{M/2+1} - Y_1)}\right)\right] \quad (11)$$

Equation (11) can be evaluated by generating samples of Y_1 and $Y_{M/2+1}$, similar to the method proposed earlier for evaluating (9). For fixed (n_s, n_b), the end point of the PPM EXIT curve is a constant that is independent of the PPM order, as seen from (11). The relative gain in extrinsic mutual information from $I(A, L) = 0$ to $I(A, L) = 1$, therefore, increases with the PPM order. We see from (8) and (10), that for $M = 2$ the relative gain in extrinsic mutual information is zero, as for memoryless modulations such as BPSK.

IV. Convolutional Code Concatenated With APPM

When iteratively decoding a serially concatenated system, the extrinsic information from the inner SISO is fed to the outer SISO as a priori information. As can be envisioned from Fig. 1 and Fig. 2, for continuous improvement in the error rate with iterations, the output mutual information from the outer SISO has to be greater than the previous input mutual information to the inner SISO at each iteration. As proposed in [25], one can view the interaction between the two decoders by plotting $I(A; L_e)$ as a function of $I(A; L)$ for the inner SISO alongside $I(A; L)$ as a function of $I(A; L_e)$ for the outer SISO. In order to drive the error rate to zero, the curve of the inner SISO has to lie above the curve of the outer SISO.

The input mutual information for a convolutional code must go to one in order to drive the output mutual information to one (or the error rate to zero). Since a PPM SISO will never reach an output mutual information of one, its EXIT curve will always be lower at some points than the EXIT curve of

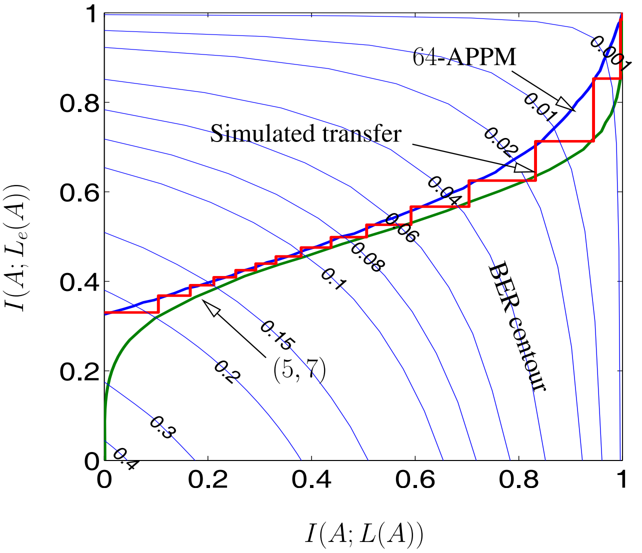


Fig. 3. EXIT curves for SCPPM and simulated evolution of mutual information, $M=64$, $n_b=0.2$, $n_s=2$.

any convolutional code used for the outer code. The serial concatenation of a convolutional code with APPM instead of PPM, remedies this situation.

With the modulation fixed to be APPM, how should one choose the convolutional code? The optimizing criterion is to find a convolutional code whose EXIT curve will be lower than the EXIT curve of the APPM SISO at as low n_s as possible, while providing a concatenated code with sufficiently large minimum distance. The distance properties of the concatenated code depend on the interleaver design and block length (the EXIT charts do not predict distance properties), and are beyond the scope of this paper. In practice, for block lengths on the order of kilobits, the constraint length 3, rate $1/2$ one input two outputs $(5,7)$ convolutional code (*i.e.* defined by generator polynomials $1+D^2$ and $1+D+D^2$) was found to give the best results.

The resulting code design is the serial concatenation of the $(5,7)$ convolutional code, followed by an interleaver, followed by APPM. We refer to this code as SCPPM. Details of an interleaver and the decoder algorithm may be found in [31]. The decoder uses two SISO's, one for APPM as one unit and another for the $(5,7)$ convolutional code. Fig. 3 shows the output mutual information versus the input mutual information for APPM, and the input mutual information versus the output mutual information for the $(5,7)$ code, for $M=64$ and $n_b=0.2$. The two mutual information curves are shown at $n_s=2$, which is a point near the threshold $n_s=1.94$ at which the two mutual information curves intersect. A stair steps like curve showing the actual evolution of mutual information from a simulation is also shown. It should be noted that decoupling the APPM SISO into two SISO's, one for PPM and another for an accumulator, will result in some reduction in complexity, but a 1-dB loss in performance [31].

V. LDPC Code Concatenated With PPM

Since an LDPC code may be designed to have an output mutual information of one for a input mutual information less than one, it lends itself to concatenation with either PPM or APPM. However, as we will see in Sections VI-A and VI-B, the APPM SISO requires a much higher processing latency, and a larger number of computations particularly for large PPM orders. In order to limit the complexity, and, since we have seen no performance gain by using APPM in concatenation with an LDPC code, we choose to design an LDPC code to be used with PPM.

A. EXIT Analysis for LDPC Coded PPM

To determine the degree distribution of our LDPC code we follow the approach of [32]. Fig. 4 shows a block diagram of the decoder illustrating the flow of messages (likelihoods) passed between an LDPC decoder and a PPM SISO and within the LDPC decoder. It also shows different points to monitor mutual information for our EXIT analysis. We refer the reader to [32] for further details on EXIT analysis for LDPC coded modulation.

To facilitate our analysis, we will determine an EXIT curve for the PPM SISO (which will be denoted by DET to match the notation of [32]) combined with the variable nodes of the LDPC decoder, and another EXIT curve for the check nodes of the LDPC decoder. The following EXIT analysis assumes that all variable nodes connected to the same PPM symbol will have the same degree, although we don't necessarily enforce this condition when we construct the code. However, the performance predicted by the EXIT analysis is still accurate.Hereinafter, we will denote the first curve by VND and the latter by CND.

We will first determine the EXIT curves for a regular code with variable node degree d_v and check node degree d_c. We define the function $J(\cdot)$ as follows:

Definition 1: Let X be a random variable such that $Pr(X=1)=Pr(X=-1)=\frac{1}{2}$, N be zero mean Gaussian noise with variance σ^2 and $Y=X+N$. We define the function $J\left(\frac{2}{\sigma}\right)$ to be the mutual information $I(X;Y)$. □

With the assumption that the LLR's exchanged between the PPM SISO and the variable nodes as well as those exchanged between the variable nodes and the check nodes are all Gaussian, then following the analysis and notation proposed in [32] and referring to Fig. 4 we can write

$$I_{A,DET}=J\left(\sqrt{d_v}J^{-1}\left(I_{A,VND}\right)\right)$$

$I_{E,DET}$ is computed from $I_{A,DET}$ by evaluating (6) numerically, obtaining the densities via simulation, i.e. without assuming that extrinsic LLR's of the PPM SISO are Gaussian. Following [32], $I_{E,VND}$ can then be given as:

$$I_{E,VND}=J\left(\sqrt{(d_v-1)\left[J^{-1}\left(I_{A,VND}\right)\right]^2+\left[J^{-1}\left(I_{E,DET}\right)\right]^2}\right)$$

For the check node curve, we use the following accurate approximation which is exact in the case of an erasure channel [32]:

$$I_{E,CND}\approx 1-J\left(\sqrt{d_c-1}\cdot J^{-1}\left(1-I_{A,CND}\right)\right) \qquad (12)$$

We may express (12) in terms of its inverse function to match the way it is plotted in Fig. 5:

$$I_{A,CND}\approx 1-J\left(\frac{J^{-1}\left(1-I_{E,CND}\right)}{\sqrt{d_c-1}}\right)$$

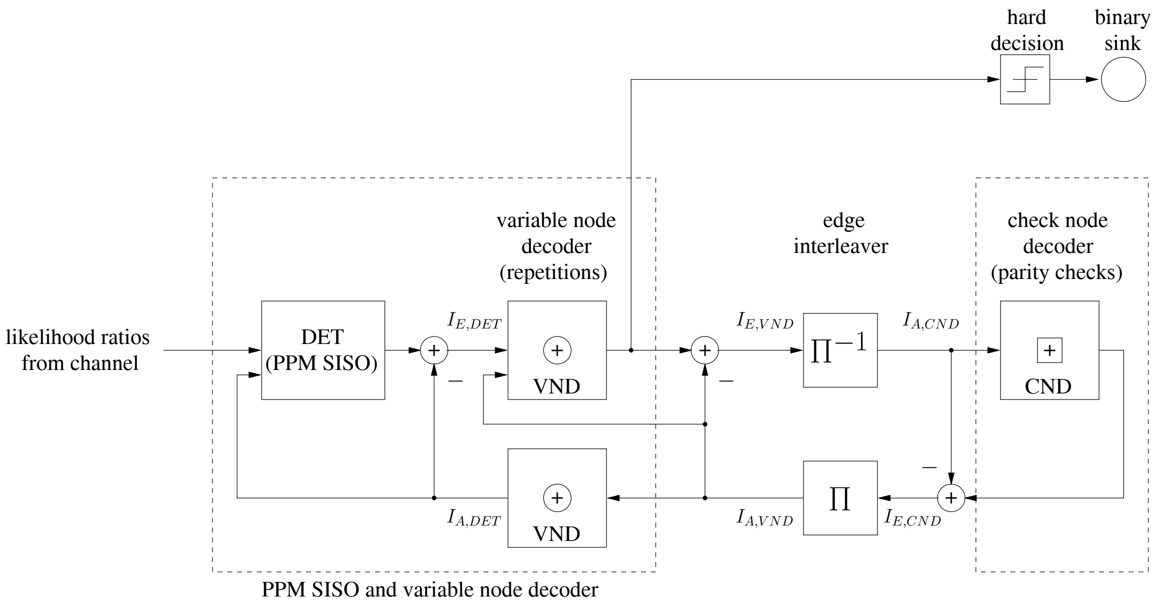


Fig. 4. Receiver model showing message passing flow and points for monitoring mutual information.

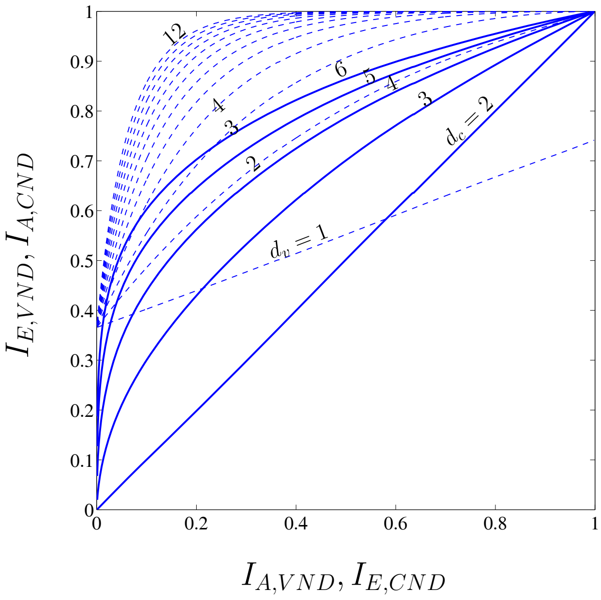


Fig. 5. VND and CND EXIT curves.

Fig. 5 shows our VND curves and CND curves for various values of d_v and d_c. These curves were generated using $M = 64$, $n_b = 0.2$ and $n_s = 1.995$. For a regular code to be converging at any particular n_s, its VND curve should be above its CND curve at this n_s. For an irregular code, the overall VND and CND curves are weighted averages of the different VND and CND curves depending on the degree distribution of the variable and check nodes. Fixing n_b, for any degree distribution, the n_s at which the overall VND curve touches the overall CND curve determines the threshold of the corresponding code. Minimization of this threshold is our optimization criteria.

B. LDPC Code Construction

An unconstrained optimization of the degree distributions of variable and check nodes may be unnecessarily complex, and could yield impractical degree distributions. Moreover, some heuristic constraints are helpful to reduce the error floor of the resulting code, which is not strictly predicted via EXIT curves alone. Using the EXIT curves, we optimized the degree distribution of our rate $1/2$ LDPC code subject to the following heuristic constraints:

1) All check nodes have the same degree, to simplify our analysis.
2) The number of degree two variable nodes should be no more than 50% of the total number of check nodes to avoid a high error floor.
3) The maximum variable node degree should be no more than 24. The rate of improvement in performance due to increasing the maximum variable node degree becomes negligible beyond a maximum variable node degree of 24. Also, allowing a higher maximum variable node degree makes the decoder less practical from a computational and implementation point of view.

Applying an exhaustive search for the best threshold subject to the above conditions, and rounding the degree distribution to rational numbers with a denominator less than 100, the following degree distribution was found: $21/88$ of the variable nodes are degree 2, $66/88$ are degree 3, $1/88$ are degree 24 and all check nodes are degree 6. It should be noted that the above conditions inherently exclude linear codes with a dense parity check matrix from the search space, eliminating therefore most non-trivial convolutional codes. Using this degree distribution, and fixing n_b to 0.2, the overall VND curve is above the overall CND curve for $n_s > 2.13$, which is the predicted threshold using this analysis technique.

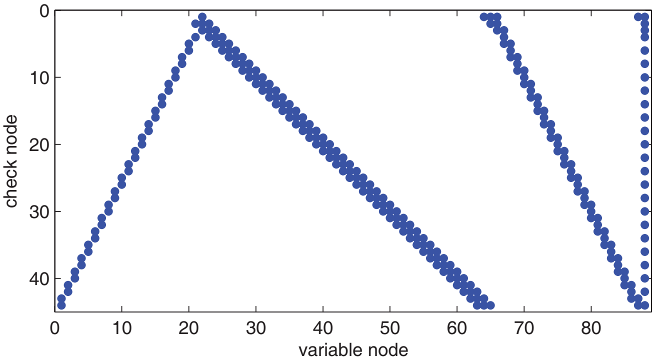


Fig. 6. Protograph of LDPC code developed for PPM.

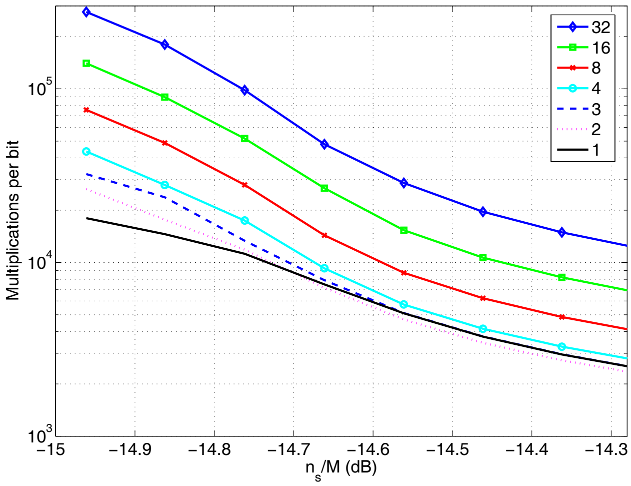


Fig. 7. Multiplication operations per bit for different numbers of LDPC iterations per PPM SISO iteration, $n_b = 0.2$, M=64.

The implementation of the encoder and decoder in hardware may be facilitated by constructing the LDPC code graph via repetitions of a basic underlying graph, or protograph. Moreover, a protograph lends itself to improving the performance by controlling the connections between variable and check nodes. Using a methodology similar to that of [33], we constructed a protograph, shown in Fig. 6, for this degree distribution. The protograph has 88 variable nodes and 44 check nodes. A dot in Fig. 6, means that the corresponding variable node and check node are connected. The full code graph is built by connecting 93 copies of this protograph in a way similar to that described in [33]. The variable nodes of the LDPC code are then ordered at random, which is functionally equivalent to placing a random interleaver between the LDPC code and PPM.

C. Decoder Scheduling

For the SCPPM decoder, the outer code SISO is run once each time the APPM SISO is run, in order for each to have updated inputs. However, with LDPC-PPM, each decoding of the LDPC variable nodes sends updated messages to both the check nodes and the PPM SISO, and one has the option of scheduling these iterations in various manners. If the maximum allowed number of iterations between the PPM SISO and the LDPC decoder is small, scheduling does have some impact on performance. However, with the maximum number of iterations between the PPM SISO and the LDPC decoder sufficiently large (≈ 100), the number of iterations between the LDPC variable and check nodes each time the LDPC decoder is run (hereinafter denoted by LDPC iterations per PPM SISO iteration), has negligible impact on performance. However, it does have a large impact on decoding complexity.

Fig. 7, shows the number of multiplication operations per user bit versus the average received signal photons per slot (n_s/M) for different numbers of LDPC iterations per PPM SISO iteration, at $M = 64$ and $n_b = 0.2$. It is clear from this figure that for this PPM order and background noise, the number of multiplications in the operation region of practical interest (as will be seen is Section VI-C), is minimized by choosing two LDPC iterations per PPM SISO iteration. The number of addition operations follows a similar behavior, and is also minimized by choosing two LDPC iterations per PPM SISO iteration. This schedule is therefore used in subsequent numerical results, with the maximum number of PPM SISO iterations set to at least 100. Our stopping rule terminates the iterative decoding process once a valid LDPC codeword is found, or the maximum number of PPM SISO iterations is reached. With this scheme, simulation results show that the average number of PPM SISO iterations is less than 10, at a word error rate of 10^{-4}.

VI. Comparison of the Two Coded Modulations

A. Parallel Processing

The decoder of SCPPM consists of two SISO's. The first SISO is for APPM and the second SISO is for the rate $1/2$ one input two outputs convolutional code defined by the generator polynomials $1 + D^2$ and $1 + D + D^2$. For the forward or backward paths of the BCJR algorithm [28], the codeword may be partitioned to smaller blocks and the blocks can be processed in parallel with a negligible loss in performance (up to some limit). However, within each block of the codeword, processing a symbol in the forward or backward paths depends on the results of processing the previous symbol. Therefore, within each block, the forward and backward paths have to be completed in a serial fashion, no matter how many parallel processors there are. This represents a major algorithmic computational difference between the APPM SISO used in the SCPPM code and the PPM SISO used in the LDPC-PPM code. Unlike APPM, a PPM SISO can theoretically process all the symbols in parallel (although the number of processors may be too high for a practical implementation in today's technology), because there aren't any dependencies between symbols. Furthermore, the message passing algorithm of an LDPC decoder [34], as used in LDPC-PPM, lends itself naturally to parallel processing, as opposed to the SISO required for the convolutional code of SCPPM. Hence an LDPC-PPM decoder lends itself more naturally to parallel processing and a smaller decoding latency. It should also be noted that a decoder that can decode one codeword using parallel hardware requires significantly less memory than multiple parallel decoders decoding multiple codewords in parallel.

B. Operations Count

In this section, we find the approximate operations count for SCPPM and LDPC-PPM. Throughout this section, we assume that multiplications and divisions belong to the same category of operations. Table I shows the operations count for the PPM SISO and the APPM SISO. We ignore $\log_2 M$ with respect to M, i.e., we use the approximation $M \pm \log_2 M \approx M$. We also ignore operation counts that are not a function of M if the number of operations per bit is less than 2. It is insightful to note that in the log domain, multiplication operations are transformed to addition operations, and addition operations are transformed to $\max^*$ operations defined as:

$$\begin{aligned} max^*(x,y) &= log(e^x + e^y) \\ &= max(x,y) + log(1 + e^{-|x-y|}) \end{aligned}$$

For $M = 64$, at a word error rate of about 10^{-4}, LDPC-PPM requires about 8.8 PPM iterations on the average using the schedule proposed in Section V-C, whereas SCPPM requires about 7 iterations. For LDPC-PPM with $M = 1024$, using an analysis similar to that of Section V-C, we choose a decoding schedule of 4 LDPC iterations per PPM iteration, yielding an average of about 8.4 PPM iterations at a word error rate of 10^{-4}. Table II gives the total number of operations per user bit for the complete SCPPM decoder and LDPC-PPM decoder, for $M = 64$ and $M = 1024$, at a word error rate of about 10^{-4}.

C. Performance

In this section we present simulation results for the SCPPM code and LDPC-PPM code described in this paper, as well as for an Accumulate-Repeat-Accumulate with repetition 4 (AR4A) LDPC code recently proposed for deep space RF communications [35]. As a baseline for comparison, the performance of $(63, 31)$ RS-coded PPM with RS symbols over $GF(64)$ is also presented [13]. The AR4A code is serially concatenated with PPM and iteratively decoded in the same manner as the LDPC-PPM code. All simulations are for the Poisson PPM channel with $n_b = 0.2$ $photons/slot$ and $M = 64$. The codeword length is 8208 bits (i.e. 4104 user bits) for SCPPM, 8184 bits (i.e. 4092 user bits) for LDPC-PPM, and 8192 bits (i.e. 4096 user bits) for AR4A-PPM. Fig. 8 shows the bit error rate P_b of these codes relative to the capacity of the Poisson PPM channel. At $P_b \approx 10^{-5}$ (corresponding roughly to a word error rate of 10^{-4} for the codes presented), SCPPM is about 1 dB away from capacity, LDPC-PPM is about 1.4 dB away from capacity, AR4A-PPM is about 2.2 dB away from capacity, while RS-PPM is about 4 dB away from capacity. The gap between SCPPM and LDPC-PPM agrees with the thresholds predicted for both codes in Sections IV and V-B. The performance of AR4A-PPM demonstrates that LDPC codes optimized for the BPSK AWGN channel are not optimal for the PPM Poisson channel.

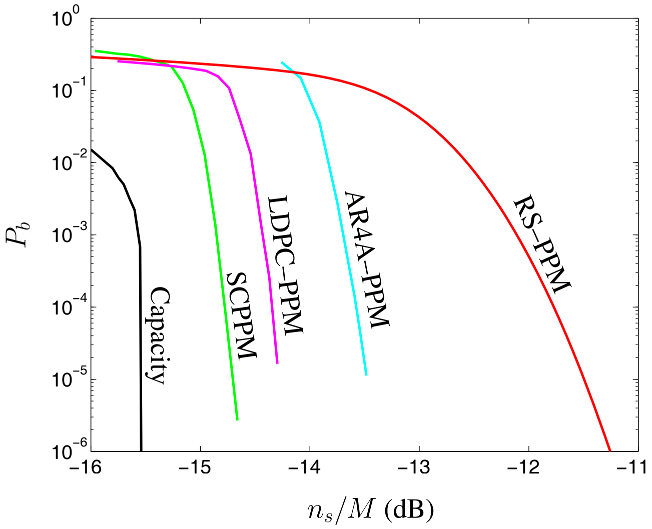


Fig. 8. Bit error rate performance of SCPPM, LDPC-PPM, AR4A-PPM, and RS-PPM in comparison to capacity, $n_b = 0.2$, M=64.

VII. Conclusion

We have developed a SCPPM code as well as an LDPC-based code for deep space optical communications. For block-lengths of ≈ 8 Kbits, the SCPPM code performance is ≈ 1 dB from capacity while the LDPC-based code is ≈ 1.4 dB away from capacity. However, the LDPC code lends itself very naturally to parallel processing in contrast to the SCPPM code. Either code would make a sensible solution depending on the PPM order, latency requirements, and the desired implementation architecture.

Acknowledgments

Ken Andrews at JPL kindly provided us with an LDPC software encoder and decoder and other useful software tools. The authors would also like to thank the reviewers of this paper for providing valuable comments and suggestions that significantly improved its presentation.

References

[1] H. Hemmati, editor, "Deep space optical communications," ser., *Deep-Space Commun. Navigation*. John Wiley & Sons, 2006.

[2] D. Boroson, C.-C. Chen, and B. Edwards, "Overview of the Mars laser communications demonstration project," ser. *Digest of the LEOS Summer Topical Meetings*, July 2005, pp. 5–7.

[3] A. D. Wyner, "Capacity and error exponent for the direct detection photon channel–part II," *IEEE Trans. Inf. Theory*, vol. 34, no. 6, pp. 1462–1471, Nov. 1988.

[4] S. Shamai, "Capacity of a pulse amplitude modulated direct detection photon channel," *IEE Proceedings I Commun., Speech Vision*, vol. 137, no. 6, pp. 424–430, Dec. 1990.

[5] D. Boroson, "Unified theory of channel capacity for photon-counting pulsed signals in noise, revision 1.0," MIT Lincoln Laboratory, Tech. Rep., Feb. 2004, mLCD Project Memo.

[6] B. Moision and J. Hamkins, "Deep-space optical communications downlink budget: modulation and coding," *IPN Progress Report*, vol. 42-154, Aug. 2003. [Online]. Available: tda.jpl.nasa.gov/progress_report/42-154/title.htm

[7] ——, "Multipulse PPM on discrete memoryless channels," *IPN Progress Report*, vol. 42-160, May 2005. [Online]. Available: tda.jpl.nasa.gov/progress_report/42-161/title.htm

[8] ——, "Constrained coding for the deep-space optical channel," *IPN Progress Report*, vol. 42-149, Jan. 2002. [Online]. Available: tda.jpl.nasa.gov/progress_report/42-149/title.htm

[9] R. Barron, "Binary shaping for low duty-cycle communications," in *Proc. IEEE International Symposium Inf. Theory*, 2004, p. 513.

[10] A. Biswas and S. Piazzolla, "Deep-space optical communications downlink budget from Mars: system parameters," *IPN Progress Report*, vol. 42–154, Aug. 2003. [Online]. Available: tda.jpl.nasa.gov/progress_report/42-154/title.htm

TABLE I
OPERATIONS COUNT FOR EACH PPM SYMBOL ($\log_2 M$ OUTER CODE BITS) PER SISO ITERATION

	PPM		APPM	
Steps	Multiplications	Additions	Multiplications	Additions
Computing edge likelihoods	$3M$		$4M$	
Computing state likelihoods				$2M$
Computing bit likelihoods		$M\log_2 M$	$4M$	$2M\log_2 M$
Total	$3M$	$M\log_2 M$	$8M$	$2M(\log_2 M + 1)$
For $M = 64$	192	384	512	896
For $M = 1024$	3072	10240	8192	22528

TABLE II
TOTAL OPERATIONS COUNT PER USER BIT FOR SCPPM AND LDPC-PPM

	SCPPM		LDPC-PPM	
	Multiplications	Additions	Multiplications	Additions
Total operations per user bit for M=64	1419	2231	2042	2394
Total operations per user bit for M=1024	8352	22628	8024	19723

[11] J. R. Pierce, “Optical channels: practical limits with photon counting,” *IEEE Trans. Commun.*, vol. 26, no. 12, pp. 1819–1821, Dec. 1978.

[12] R. McEliece and L. Welch, “Coding for optical channels with photon counting,” *Jet Propulsion Laboratory Deep Space Network Progress Reports*, vol. 42–52, pp. 61–66, 1979.

[13] R. J. McEliece, “Practical codes for photon communication,” *IEEE Trans. Inf. Theory*, vol. 27, no. 4, pp. 393–398, July 1981.

[14] G. S. Mecherle, “Maximized data rate capability for optical communication using semiconductor devices with pulse position modulation,” Ph.D. dissertation, University of Southern California, May 1986.

[15] D. Divsalar, R. M. Gagliardi, and J. H. Yuen, “PPM demodulation for Reed-Solomon decoding for the optical space channel,” *TDA Progress Report*, vol. 42–70, pp. 47–59, May 1982. [Online]. Available: tda.jpl.nasa.gov/progress_report/42-70/70title.htm

[16] V. Guruswami and M. Sudan, “Improved decoding of Reed-Solomon and algebraic-geometry codes,” *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 1757–1767, Sep. 1999.

[17] R. Koetter and A. Vardy, “Algebraic soft-decision decoding of Reed-Solomon codes,” *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 2809–2825, Nov. 2003.

[18] J. L. Massey, “Capacity, cutoff rate, and coding for a direct-dectection optical channel,” *IEEE Trans. Commun.*, vol. 29, no. 11, pp. 1615–1621, Nov. 1981.

[19] C. Berrou and A. Glavieux, “Near optimum error correcting coding and decoding: turbo-codes,” *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1276–1271, Oct. 1996.

[20] K. Kiasaleh, “Turbo-coded optical PPM communication systems,” *J. Lightwave Technol.*, vol. 16, no. 1, pp. 18–26, Jan. 1998.

[21] T. Ohtsuki and J. M. Kahn, “BER performance of turbo-coded PPM CDMA systems on optical fiber,” *J. Lightwave Technol.*, vol. 18, no. 12, pp. 1776–1784, Dec. 2000.

[22] J. Hamkins, “Performance of binary turbo coded 256-PPM,” *TMO Progess Report*, vol. 42, no. 138, pp. 1–15, Aug. 1999. [Online]. Available: tda.jpl.nasa.gov/progress_report/42-138/title.htm

[23] J. Y. Kim and H. V. Poor, “Turbo-coded optical direct-detection CDMA system with PPM modulation,” *J. Lightwave Technol.*, vol. 19, no. 3, pp. 312–323, Mar. 2001.

[24] M. Peleg and S. Shamai, “Efficient communication over the discrete-time memoryless Rayleigh fading channel with turbo coding/decoding,” *European Trans. Telecommun.*, vol. 11, no. 5, pp. 475–485, Sep.-Oct. 2000.

[25] S. ten Brink, “Convergence of iterative decoding,” *Electron. Lett.*, vol. 35, no. 10, pp. 806–808, May 1999.

[26] ——, “Convergence behaviour of iteratively decoded parallel concatenated codes,” *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

[27] A. Ashikhmin, G. Kramer, and S. ten Brink, “Extrinsic information transfer functions: model and erasure channel properties,” *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2657–2673, Nov. 2004.

[28] L. Bahl, J. Cocke, F.Jelinek, and J.Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inf. Theory*, vol. 20, pp. 284–287, Mar. 1974.

[29] 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*, vol. 42-127, pp. 1–20, July 1996. [Online]. Available: tda.jpl.nasa.gov/progress_report/42-127/title.htm

[30] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[31] B. Moision and J. Hamkins, “Coded modulation for the deep space optical channel: serially concatenated PPM,” *IPN Progress Report*, vol. 42-161, May 2005. [Online]. Available: tda.jpl.nasa.gov/progress_report/42-161/title.htm

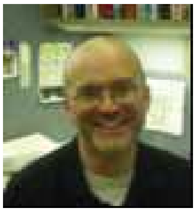
[32] S. ten Brink, G. Kramer, and A. Ashikhmin, “Design of low-density parity-check codes for modulation and detection,” *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, Apr. 2004.

[33] J. Thorpe, K. Andrews, and S. Dolinar, “Methodologies for designing LDPC codes using protographs and circulants,” in *Proc. IEEE International Symposium Inf. Theory*, 2004, p. 236.

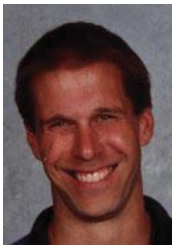
[34] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[35] D. Divsalar, S. Dolinar, J. Thorpe, and C. Jones, “Constructing LDPC codes from simple loop-free encoding modules,” in *Proc. IEEE International Conf. Commun.*, May 2005, vol. 1, pp. 658–662.

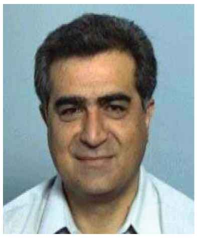
Maged F. Barsoum was born in Alexandria in 1970. He received his B.Sc. degree in computer science and automatic control from Alexandria University in 1991, his M.S. degree in electrical engineering from the University of Central Florida (UCF) in 1994, and his Ph.D. degree in electrical engineering from the University of California, Los Angeles (UCLA), in 2007. Dr. Barsoum is currently with Broadcom Corporation, San Jose, CA. He has previously held technical positions at the Jet Propulsion Laboratory (JPL), L-3 Communications, and AMD, and has consulted for different companies including Memscap, Newport Media, and Wavesat. He has been involved in the research and development of the physical layer of different standardized and proprietary wired, wireless, and optical modems. Dr. Barsoum has also served as a consulting expert for patent infringement disputes in communications. Dr. Barsoum's interests include modern codes, bandwidth efficient modulation, communication theory, and signal processing. He is named an inventor on sixteen granted US patents and several pending patents.



Bruce Moision received a B.S. in engineering from Harvey Mudd College in 1991, and a PhD. in electrical engineering from UCSD in 1999 He has been with the Jet Propulsion Laboratory (JPL) since 2001. His research interests include coding, modulation, and signal processing for the optical communications channel.



Michael P. Fitz (S'82-M'83-SM'02) received the B.E.E. degree (summa cum laude) from the University of Dayton, Dayton, OH, in 1983, and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1984 and 1989, respectively. From 1983 to 1989, he was a Communication Systems Engineer with Hughes Aircraft and TRW Inc. From 1989-2005, he held academic positions of increasing rank at Purdue University, West Lafayette, IN; The Ohio State University (OSU), Columbus; and the University of California, Los Angeles. Dr. Fitz is currently with Northrop Grumman Corp. as a Distinguished Engineer of Communication Systems and an Advanced Systems Manager working on satellite communications. He is the author of *Fundamentals of Communications Systems* (New York: McGraw Hill, 2007). Prof. Fitz received the 2001 IEEE Communications Society Leonard G. Abraham Prize Paper Award in the field of communications systems.



Dariush Divsalar received the Ph.D. degree in electrical engineering from UCLA in 1978. Since then, he has been with the Jet Propulsion Laboratory (JPL), California Institute of Technology (Caltech), where he is a Principal Scientist. At JPL, he has been involved with developing state-of-the-art technology for advanced deep space communications systems and future NASA space exploration. Since 1986, he has taught graduate courses in communications and coding at UCLA and Caltech. He has published more than 150 papers, coauthored a book entitled *An Introduction to Trellis Coded Modulation with Applications*, contributed to two other books, and holds twelve U.S. patents in the above areas. He was co-recipient of the 1986 paper award of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He was also co-recipient of the joint paper award of the IEEE Information Theory and IEEE Communication Theory societies in 2008. The IEEE Communication Society has selected one of his papers for inclusion in a book entitled *The Best of the Best: Fifty Years of Communications and Networking Research*, containing the best 56 key research papers ever published in the Society's 50-year history. He has received over 35 NASA Tech Brief awards and a NASA Exceptional Engineering Achievement Medal in 1996. He served as an Area Editor for the IEEE TRANSACTIONS ON COMMUNICATION from 1989 to 1996.

Jon Hamkins received his B.S. from Caltech in 1990, and Ph.D. from the University of Illinois at Urbana-Champaign in 1996, both in electrical engineering. Dr. Hamkins has been at JPL since 1996, where he is the supervisor of the Information Processing group, which performs research in autonomous radios, optical communications, information theory, channel coding, data compression, and synchronization.