IPN Progress Report 42-173 • May 15, 2008

Universal Decoder for Variable Duty-Cycle Optical Communications

Bruce Moision*

Efficient use of a deep-space optical communications channel requires changes to the duty cycle of the modulation as the signal and noise powers change. This can be facilitated by modulating the signal with pulse-position-modulation (PPM), supporting multiple PPM orders. To implement iterative demodulation, which is required by certain error-correction-codes to obtain near-capacity performance, would nominally require a distinct hardware implementation for each PPM order. In this paper we describe a method to utilize a single hardware implementation of an iterative demodulator for any PPM order. The method may be applied to any coded modulation that utilizes iterative demodulation and maps to multiple modulation orders.

I. Introduction

A deep-space optical communications channel experiences a wide range of signal and noise powers due to changes in atmospheric conditions and the sun-earth-spacecraft geometry. Fully utilizing the capability of the link requires changing the peak-to-average power ratio over the course of the mission. This can be accomplished efficiently by modulating the data with pulse-position-modulation (PPM), choosing an optimum order for each operating point. For example, Figure 1 illustrates a range of incident signal and noise power pairs, measured in photons/ns, representative of an Mars–Earth link. These points were obtained from link budgets developed for the discontinued Mars Laser Communications Demonstration (MLCD) project [1]. The signal–noise plane is also partitioned into optimum PPM orders and their corresponding throughputs, assuming a slot-width $T_s = 1.6$ nsec. We see that the PPM order would ideally be varied over the duration of the mission from $M = 8$ to $M = 512$. (The partitions illustrated here make no allowance for system margin, which would increase the optimum order and decrease the achievable throughput).

*Communications Architectures and Research Section

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

[1] Link budget data provided by Abhijit Biswas, Jet Propulsion Laboratory

To achieve near-capacity performance with certain error-correction-codes (ECCs), a coded modulation for this channel decodes the ECC and modulation jointly, treating the modulation as a component of the code, see, e.g., [1]. This would nominally require a distinct hardware implementation of the decoder for each PPM order. However, designing, integrating, and maintaining a large number of designs may be prohibitively expensive. In this paper, we illustrate a method to utilize a single hardware implementation of the ECC for any PPM order, while still facilitating iterative demodulation.

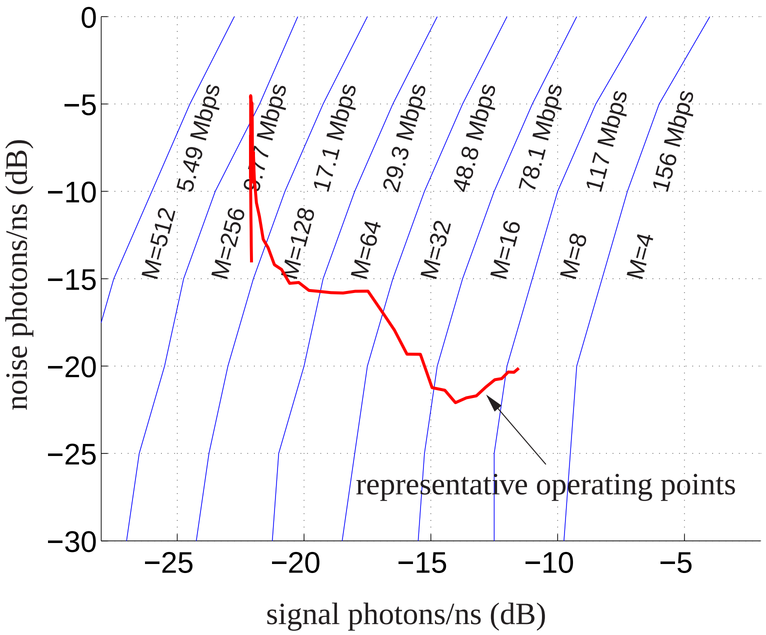


Figure 1. Mapping of MLCD incident signal and noise to optimum PPM order and corresponding throughput assuming a slotwidth of $T_s = 1.6$ nsec.

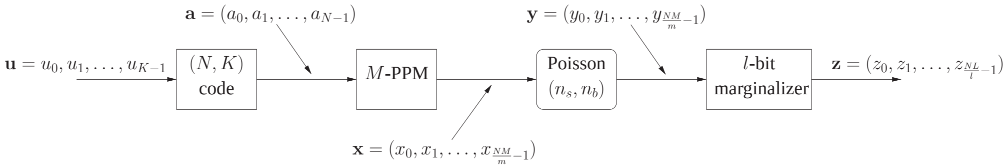

$\mathbf{u} = u_0, u_1, \ldots, u_{K-1}$
(N, K) code
$\mathbf{a} = (a_0, a_1, \ldots, a_{N-1})$
M-PPM
$\mathbf{x} = (x_0, x_1, \ldots, x_{\frac{NM}{m}-1})$
Poisson (n_s, n_b)
$\mathbf{y} = (y_0, y_1, \ldots, y_{\frac{NM}{m}-1})$
l-bit marginalizer
$\mathbf{z} = (z_0, z_1, \ldots, z_{\frac{NL}{l}-1})$

Figure 2. Coded M-ary PPM channel with l-bit marginals

II. A Marginalized l-bit Decoder

Figure 2 illustrates the processing sequence of the coded modulation. A K-bit vector $\mathbf{u} = (u_0, u_1, \ldots, u_{K-1})$ is encoded by an (N, K) binary ECC to produce the N-bit vector $\mathbf{a}$. Each block of $m = \log_2 M$ coded bits in $\mathbf{a}$ is mapped to an M-ary pulse-position-modulation (PPM) symbol. Each PPM symbol is represented as an M-ary binary vector, with each element referred to as a *slot*, where a 1 denotes the pulsed slot. We may think of the modulation as a non-linear binary $(NM/m, N)$ code. The binary vector of PPM symbols $\mathbf{x} = (x_0, x_1, \ldots, x_{(NM/m)-1})$ is transmitted over a Poisson channel such that a Poisson-distributed photon count y_i is observed over each slot, with mean $n_s + n_b$ for pulsed slots and mean n_b for noise, or background, slots.

The M slot photon counts corresponding to a single PPM symbol are conditionally correlated, given the m-bit vector that maps to those slots. This conditional correlation can be used in decoding by utilizing the joint m-bit conditional likelihoods. For any vector $\mathbf{y} = (y_0, y_1, \ldots)$ let $y^M_{k,i} = y_j$ where $i = j \mod M$ and $k = (j - i)/M$. That is, $y^M_{k,i}$ is the ith element of the kth M-ary block, and let $\mathbf{y}^M_k = (y^M_{k,0}, \ldots, y^M_{k,M-1})$, the kth M-ary block. Without loss of generality, assume the modulation sends a pulse in the ith slot of the kth symbol if $\mathbf{a}^m_k = i$, where we use the shorthand $\mathbf{a}^m_k = i$ to denote that the m-bit vector $\mathbf{a}^m_k$ has integer representation i. For example, the block $\mathbf{a}^3_1 = (011) = 3$ would map to $\mathbf{x}^8_1 = (000100000)$, and a pulse would be sent in the 3rd slot, with all other slots in that symbol unpulsed. For the Poisson PPM channel the conditional channel likelihoods may be expressed as

$$p(\mathbf{y}^M_k | \mathbf{a}^m_k = i) = C \left(1 + \frac{n_s}{n_b}\right)^{y^M_{k,i}} \tag{1}$$

where C is a constant that is not a function of i. Hence, we may think of the slot counts $y^M_{k,i}$ as each representing one likelihood.

Iterative soft demodulation is critical to achieving near-capacity performance for large PPM orders [1]. However, the complexity, measured in operations per bit, grows exponentially with m, and can become prohibitively expensive for large PPM orders that would be used by power-constrained deep-space missions. In addition, if a mission is to support multiple orders, a new hardware design would have to be generated for each order. Although a ground-based station could upload a new design for each order, developing, maintaining, and integrating a large number of designs would represent a considerable expense.

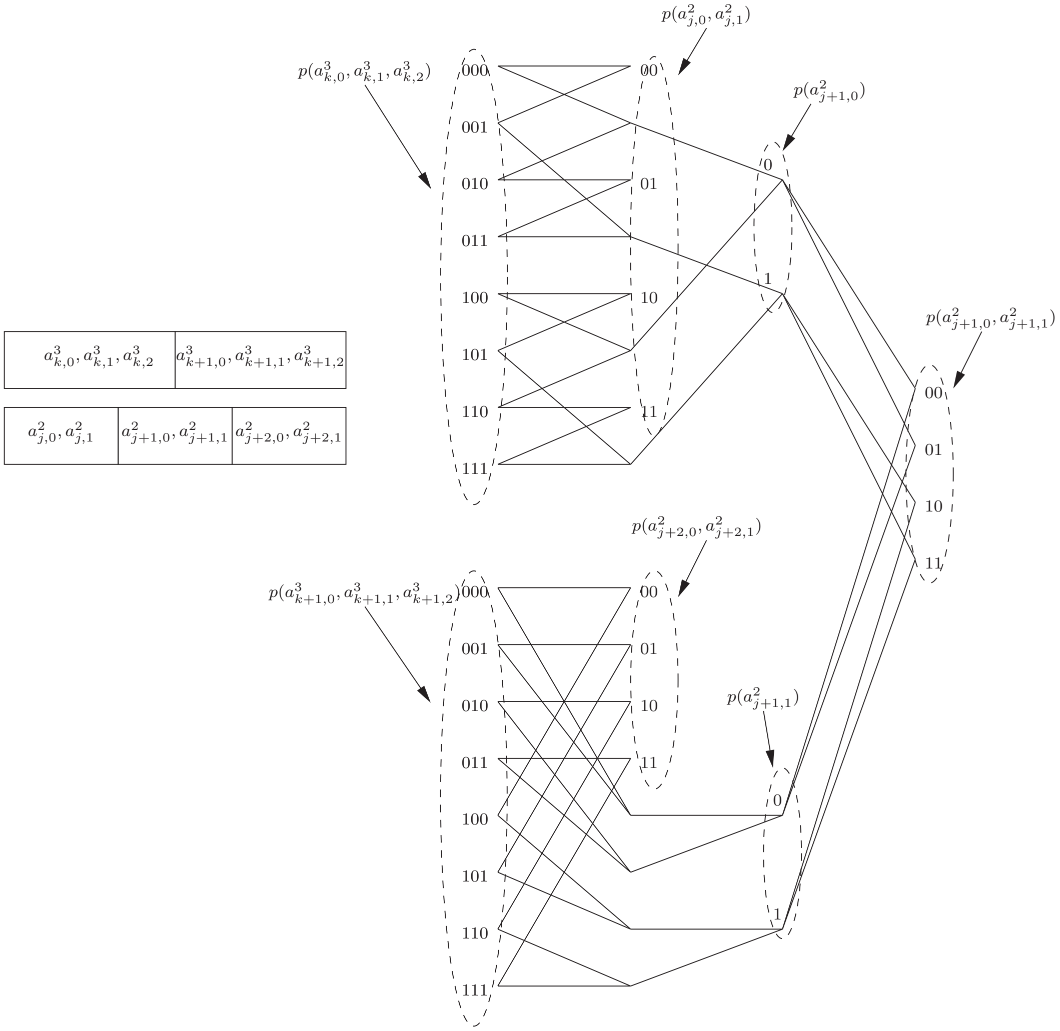
To alleviate these issues, we introduce an l-bit marginalizer into the processing sequence, as illustrated in Figure 2. The l-bit marginalizer computes joint l-bit conditional likelihoods for some fixed l regardless of the PPM order. We can think of it as replacing the computation of conditional likelihoods given by (1) that would be different for each order $M = 2^m$. For example, consider the case $l = 1$. The marginalizer obtains bit-likelihoods from the symbol likelihoods by computing the marginals

$$p(\mathbf{y}^M_k | a^m_{k,i} = 0) = \sum_{\mathbf{a}^m_k | a^m_{k,i} = 0} p(\mathbf{y}^M_k | \mathbf{a}^m_k)$$

The decoder may then operate on the bit-likelihoods, significantly reducing the complexity and allowing a single hardware implementation to be used for all orders. However, utilizing bit-likelihoods incurs a loss in performance–in Section III we illustrate a case of interest where the loss is 1.5 dB for large orders.

An intermediate solution is to marginalize the m-bit likelihoods to l-bit likelihoods for some $l > 1$. This allows a tradeoff of performance for complexity, while requiring a single hardware implementation (based on an l-bit inner trellis). Figure 3 illustrates an example of mapping $m = 3$ bit likelihoods to $l = 2$ bit likelihoods, reducing a collection of 16 likelihoods (two $M = 8$

symbols) to 12 (corresponding to three 2-bit symbols). In general l may be larger or smaller than m, as described in the following.



$p(a^2_{j,0}, a^2_{j,1})$
$p(a^3_{k,0}, a^3_{k,1}, a^3_{k,2})$
000
001
010
011
100
101
110
111
00
01
10
11
$p(a^2_{j+1,0})$
0
1
$p(a^2_{j+1,0}, a^2_{j+1,1})$
00
01
10
11
$a^3_{k,0}, a^3_{k,1}, a^3_{k,2}$
$a^3_{k+1,0}, a^3_{k+1,1}, a^3_{k+1,2}$
$a^2_{j,0}, a^2_{j,1}$
$a^2_{j+1,0}, a^2_{j+1,1}$
$a^2_{j+2,0}, a^2_{j+2,1}$
$p(a^2_{j+2,0}, a^2_{j+2,1})$
$p(a^3_{k+1,0}, a^3_{k+1,1}, a^3_{k+1,2})$
000
001
010
011
100
101
110
111
00
01
10
11
$p(a^2_{j+1,1})$
0
1

Figure 3. Forming $l = 2$ bit marginals from $m = 3$ bit likelihoods

Let $l = \log_2 L$ and put $p = \text{lcm}(m, l)$. Let k and j be integers satisfying $jm = kl$, with jm a multiple of p. Then the vectors $[\mathbf{a}^m_j, \mathbf{a}^m_{j+1}, \ldots, \mathbf{a}^m_{j+(p/m)-1}]$ and $[\mathbf{a}^l_k, \mathbf{a}^l_{k+1}, \ldots, \mathbf{a}^l_{k+(p/l)-1}]$ are identical–they index the same sequence. The first vector maps to (p/m) consecutive m-bit blocks, corresponding to p/m consecutive transmitted M-ary PPM symbols. The second vector re-groups these into (p/l) l-bit blocks. The function of the marginalizer is to compute the conditional likelihoods of the l-bit blocks given the conditional likelihoods of the m-bit blocks. That is, it computes, for each $k \in \{0, 1, \ldots, N/l - 1\}, i \in \{0, 1, \ldots, l - 1\}$

$$z^L_{k,i} = p(\mathbf{y}^M_{k'}, \ldots, \mathbf{y}^M_{k''} | \mathbf{a}^l_k = i)$$

where $k' = \lfloor kl/m \rfloor$ and $k'' = \lfloor ((k+1)l - 1)/m \rfloor$, such that the symbols $\mathbf{y}^M_{k'}, \ldots, \mathbf{y}^M_{k''}$ span the set of symbols mapped to by $\mathbf{a}_k^l$. The marginal $z_{k,i}^L$ may be computed as

$$\begin{aligned} z_{k,i}^L &= C_1 \sum_{\mathbf{a}_{k'}^m,\ldots,\mathbf{a}_{k''}^m|\mathbf{a}_k^l=i} p(\mathbf{y}_{k'}^M,\ldots,\mathbf{y}_{k''}^M|\mathbf{a}_{k'}^m,\ldots,\mathbf{a}_{k''}^m) \\ &= C_1 \sum_{\mathbf{a}_{k'}^m,\ldots,\mathbf{a}_{k''}^m|\mathbf{a}_k^l=i} \prod_{j=k'}^{k''} p(\mathbf{y}_j^M|\mathbf{a}_j^m) \\ &= C_1 \prod_{j=k'}^{k''} \sum_{\mathbf{a}_j^m|\mathbf{a}_k^l=i} p(\mathbf{y}_j^M|\mathbf{a}_j^m) \end{aligned} \tag{2}$$

where we've assumed the bits a_i, a_j, $i \neq j$ are IID Bernoulli$(1/2)$, so that C_1 is a constant relative to i and that the symbols are conditionally independent, from which (2) follows.

Consider, for example, the two cases $l = m$ and $l = 1$. When $l = m$, $k' = k'' = k$ and

$$z_{k,i}^L = C_1 p(\mathbf{y}_k^M|\mathbf{a}_k^m = i)$$

the conventional m-bit conditional likelihoods. When $l = 1$, $k' = k'' = \lfloor k/m \rfloor$ and

$$z_{k,i}^L = C_1 p(\mathbf{y}_{\lfloor k/m \rfloor}^M|\mathbf{a}_k^1 = i)$$

a binary likelihood.

III. Results

In this section we examine the performance degradation due to utilizing an l-bit decoder for M-ary PPM. All simulation results utilize a serially-concatenated-PPM code described in [1]. The code consists of the serial concatenation of a 4-state convolutional code, a binary interleaver, an accumulator, and PPM mapping. The binary ECC block length is $(N, K) = (15120, 7560)$. Codewords are transmitted over a Poisson channel with mean signal photons per pulsed slot n_s and mean noise photons per slot n_b. Decoding is performed iteratively, with decoding terminated if the correct codeword is found, or a maximum of 32 iterations is reached.

Figure 4 illustrates simulated performance on a Poisson channel with $n_b = 1$ for $l \in \{1, 2, 3, 4, 5, 6, 8\}$, $m = 8$. Performance improves as l approaches m. Figure 5 illustrates the signal power (n_s/M) in excess of capacity required to achieve a word error rate of $P_w = 10^{-3}$ using an l-bit decoder to decode m-bit PPM. For example, from Figure 4, the gap to capacity for $l = 4$ and $m = 6$ is approximately 1.4 dB. There is no loss relative to the m-bit decoder when l is divisible by m, since no marginalizations of the m-bit likelihoods are required. However, when $l > m$ and l does not divide m, e.g., when $l = 4, m = 3$, we see some loss since marginalizations of some of the m-bit likelihoods are required. Similarly, when $l < m$, we see slightly better performance when l divides m.

Figure 6 illustrates the average operations per information bit required for l-bit and m-bit decoders to attain a word error rate of $P_w = 10^{-3}$. All operations are included–not just those of iterative demodulation. We see that for a given l, the complexity of decoding remains constant (roughly constant, the iterations actually decrease with m yielding a small decrease in complexity), whereas the complexity of m-bit decoding grows exponentially with m. Memory requirements are reduced by a comparable amount. The complexity savings can be significant. For example, an $l = 4$ decoder decoding $m = 8$ requires 5.4 times fewer operations than the full $m = 8$ decoder, while suffering a loss of 0.66 dB.

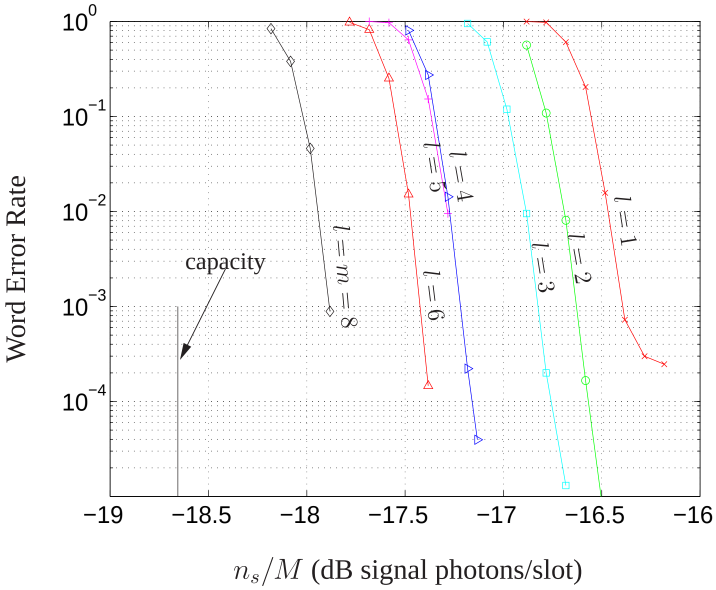


Figure 4. Word Error Rate of $m = 8$ ($M = 256$ PPM) SCPPM decoded with an l-bit decoder, $l \in \{1, 2, 3, 4, 5, 6\}$

IV. Comparisons with alternative approaches

In this section we compare the performance of the decoder with a front-end marginalizer with two other approaches that also allow one to implement variable duty-cycle coded modulation with little or no modification to the decoder hardware.

A. Guard-Time

The duty-cycle of the modulation may be varied without changing the PPM order by following each PPM symbol by d unpulsed, or *guard-time* slots. Suppose you have a parent code with throughput T bits/sec and average power threshold $P_{av} = n_s/M$ photons/slot. Adding d guard-time slots while keeping n_s fixed has no impact on performance, since the guard-time slots are effectively ignored in the decoder. However, this reduces the average power to $P'_{av} = n_s/(M + d) = P_{av}M/(M + d)$ and the throughput to $T' = TM/(M + d)$. Adding guard-time allows one to vary the duty cycle without changing the encoding or decoding, since M-ary symbol likelihoods are used. Symbol and codeword synchronization algorithms would be modified to account for the guard-time, and the guard-time periods themselves may be used by the receiver for symbol synchronization.

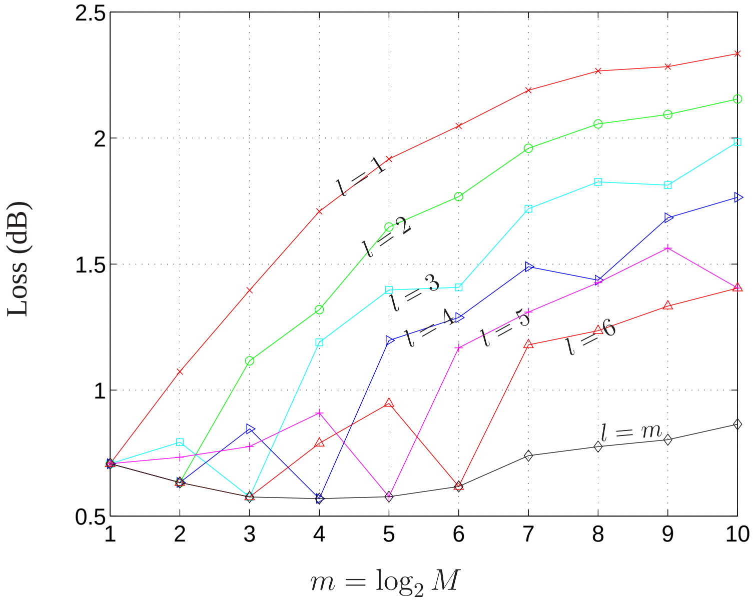


Figure 5. Signal power in excess of capacity required to achieve $P_w = 10^{-3}$ as a function of (m, l)

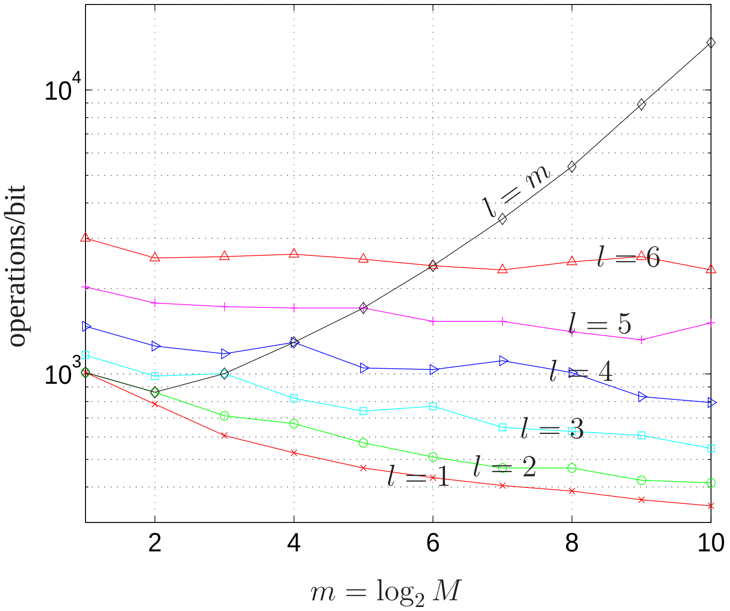


Figure 6. Average operations/bit to attain $P_w = 10^{-3}$

B. Variable slot widths

As the available average power decreases, the optimum peak-to-average power of the link increases. In practice, physical constraints limit the achievable peak power of a laser transmitter, constraining the range of duty cycles that may be implemented while utilizing all available average power at a given slotwidth. For a fixed slotwidth and ECC rate, this limits the range of signal power over which one can close a link. However, by varying the slotwidth, the operable region can be increased by increasing the average signal photons per pulsed slot[2]. One can show that at low average power, the capacity of the link is invariant to the slotwidth for a fixed duty cycle, see, e.g., [3]. Hence, by increasing the slotwidth, while retaining the same peak power and duty cycle, the operable region can be increased. The decoder is essentially blind to the slotwidth, and no changes are required to the hardware to accommodate a change in the slotwidth. Slot and symbol synchronization algorithms may remain unchanged if larger effective slots are constructed from a baseline minimum slotwidth, as long as the signal strength is sufficiently large.

C. Comparisons

Figure 7 illustrates the achievable power efficiency in photons/bit for each of the described approaches to implementing a variable duty cycle coded modulation on the Poisson channel. The noise background is fixed at $\lambda_b = 1$ photon/nsec, the ECC rate at $1/2$, and the minimum slotwidth for all cases is $T_s = 1$ nsec. The best performance is achieved by choosing the minimum slotwidth, choosing the optimum PPM order for that signal power, and using an $l = m$ decoder. Performance is also illustrated for a decoder with front-end marginalizer $l = 4$, for baseline $M = 16$ plus varying duration guard-time, and baseline $M = 16$ with varying slotwidth. The minimum slotwidth is used for all but the variable slotwidth system. The $l = 4$ decoder has a small loss relative to the $l = m$ decoder. The required bits/photon is fixed for the guard-time decoder, since adding guard-time does not change the power efficiency. The variable slotwidth decoder is relatively inefficient, but, as noted, does not require an increase to the peak to average power ratio.

Figure 8 illustrates the increase in signal photon rate relative to the M-PPM system (minimum slotwidth, optimum PPM order, $l = m$ decoder) required to achieve $P_w = 10^{-3}$ as a function of the data rate in Mbps. The data rate is varied by changing M for the $l = 4$ bit decoder, d for the guard-time decoder, and T_s for the variable-slotwidth decoder. $M = 16$ for the guard-time and variable slotwidth decoders. Losses are greatest with the variable slotwidth. Varying losses are observed with the $l = 4$ marginals decoder, bounded by 0.7 dB over the range observed. Losses are greater with the guard-time decoder. Note that, without changes to the decoder, neither the variable slotwidth decoder nor the guard-time decoder allow operation at rates above that achieved with the baseline modulation–125 Mbps for $M = 16$, ECC rate $1/2$ and slotwidth 1 nsec.

[2] Mars Lasercom Terminal To Ground Terminals Interface Control Document, MLCD-ICD2, Release 2.1a, February 1, 2005

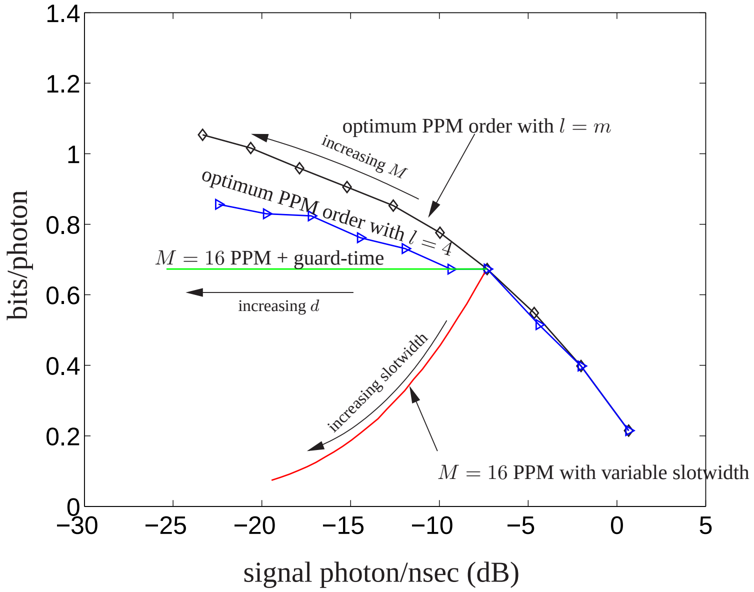


Figure 7. Achievable efficiencies for various approaches to scalable coded modulation

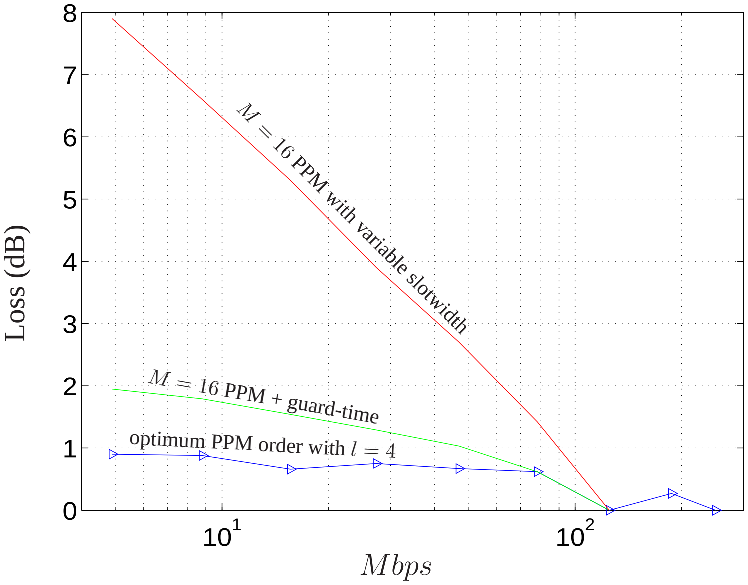


Figure 8. Losses for for various approaches to scalable coded modulation

V. Conclusions

We have demonstrated an architecture that allows a single hardware decoder to decode a wide range of coded modulations with PPM modulation. This greatly simplifies the implementation of a system that allows a variable duty cycle. It not only reduces the implementation complexity of the decoder, but can lead to simplifications of other components of an optical communications system. For example, a large channel interleaver may be introduced to mitigate fading on the channel [4]. The interleaver interleaves symbol likelihoods and would need to be modified to operate with varying symbol durations if the PPM order is changed. By mapping all symbol likelihoods to a common length, the channel interleaver design can be fixed. Other components of the system have similar reductions in complexity. The approach can be extended to coded modulations where the hardware implementation of the code is matched to a wide range of modulations and iterative demodulation is allowed in the decoder.

References

[1] B. Moision and J. Hamkins, “Coded modulation for the deep space optical channel: Serially Concatenated PPM,” *IPN Progress Report*, vol. 42-161, 2005.

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

[3] B. Moision and J. Hamkins, “Deep-space optical communications downlink budget: Modulation and Coding,” in *IPN Progress Report*, vol. 42-154, Aug. 2003.

[4] R. J. Barron and D. M. Boroson, “Analysis of capacity and probability of outage for free-space optical channels with fading due to pointing and tracking error,” in *Proceedings of SPIE* (G. S. Mecherle, ed.), vol. 6105, Proceedings of the SPIE, March 2006.