Convolutional Coded DPIM for Indoor Optical Wireless Links

N. M. Aldibbiat, S. Rajbhandari and Z. Ghassemlooy

Optical Communications Research Group, NCRLab, School of Computing, Engineering & Information Sciences, Northumbria University, Ellison Building, Newcastle upon Tyne, NE1 8ST, UK.
Nawras@iee.org, Sujan.rajbhandari@unn.ac.uk, Fary.ghassemlooy@unn.ac.uk

Abstract: In this paper we evaluate the performance of digital pulse interval modulation (DPIM) with convolutional coding for different values of bit resolution and different sizes of guard bands. We show that convolutional coding_is an effective method of improving the error performance of DPIM over optical wireless links. The convolutional coded DPIM with a guard band of 2 slots has the advantage of fixed header pattern so that decoding is simple with no need for Viterbi decoding. It also achieves a code gain of more than 4 dB compared to standard DPIM with 2 guard slot at slot error rate of 10^{-4}.

I. Introduction

Optical wireless communications have attracted attentions from many researchers worldwide for their potential advantages over radio links. These advantages include the availability of huge unregulated bandwidth, theoretically 200 THz in the range of 700-1500 nm range [1]. Optical wireless links also offer high data rates, immunity to electromagnetic interference, relative security since it does not penetrate walls, and the ability to reuse the same wavelength in adjacent rooms [1-3].

With the emerging communication technology, size of signals such as video and high-resolution images is growing tremendously causing an increasing demand for modulation schemes with higher bit rate and lower bandwidth requirement. Anisochronous pulse modulation schemes such as digital pulse interval modulation (DPIM) is an alternative to on-off-keying (OOK) and pulse position modulation (PPM) for indoor infrared wireless links [4-5]. Although PPM offers the best performance and power efficiency, but it requires high transmission bandwidth and complex system implementation, whereas DPIM improves the transmission bit rate by removing the empty slots following the pulse as in PPM symbol [4], and it offers self synchronisation capabilities. To improve the performance of system error control coding may be used [5-6]. Error control coding improves the capacity of a channel by adding redundant information to the transmitted data. They are broadly classified into two groups: convolutional coding and block coding. Convolutional coding with Viterbi decoding is a forward error correction (FEC) technique suitable for a channel corrupted mainly by additive white Gaussian noise (AWGN) [7]. Convolutional coding is more efficient than block coding [7-9]. In Viterbi decoding, proposed by Viterbi in 1967 [9], the decoder examines the full received sequence of a given length, computes a metric for each path and makes a decision based on this metric and follows all paths until two paths converge on one node. The path with the higher metric is selected and called survivor.

.In this paper we investigate the performance of DPIM with convolutional coding. We consider guard bands of 0, 1 and 2 guard slots following each pulse. The guard band prevents two consecutive pulses and allows for a simple decoding at the receiver. We compare the simulation result with other modulation techniques. The paper is organized in the following order. A brief introduction of DPIM is presented in the Section II, followed by convolutional coding in section III. The symbol structure of convolutional coded DPIM is explained in section IV. The simulation results are presented in section V, and the conclusions are given in the final section.

II. Digital pulse interval modulation (DPIM)

DPIM is an anisochronous pulse time modulation technique in which data is encoded as a number of discrete time slots, between adjacent pulses. The symbol length is variable and is determined by the information content of the symbol. In order to avoid symbols in which the time between adjacent pulses is zero, an additional guard band of one or more slots may be added to each symbol immediately following the pulse. Detailed information about standard DPIM can be found in the literature [4-5]. In this paper we represent DPIM a guard band of 0, 1 or 2 guard slots, which are referred to as DPIM(0GS), DPIM(1GS) and DPIM(2GS), respectively.. Thus, a DPIM symbol which encodes M bits of data is represented by a pulse of constant power in one slot followed by k slots of zero power, where $1 \le k \le L+1$ and $L = 2^M$.

Each symbol of DPIM(0GS), DPIM(1GS) and DPIM(2GS) start with {1}, {1 0} or {1 0 0}, respectively followed by a number of empty slots according to the decimal value of the input data [3]. Here “1” represents a

Table 1: DPIM(0GS), DPIM(1GS) DPIM(2GS) and CC- DPIM(2GS) symbol structure

Decimal value of data	**DPIM(0GS)**	**DPIM(1GS)**	**DPIM(2GS)**	**CC- DPIM(2GS)**
0	1	10	100	**11 10 11**
1	10	100	1000	**11 10 11 00**
……	…..	……	……	......
n	10……0 (n)	10……0 (n+1)	10……00 (n+2)	**11 10 11** 00……00 (2n)

pulse and {0} represents an empty slot. Table 1 shows examples of DPIM symbol structures for different input data.

III. Convolutional Coding

Convolutional coding is forward error control coding .Unlike block codes, which takes a block of data and adds some redundant bits; convolutional code works on the serial input [8] so that it is suitable for the modulation techniques having variable symbol length. Convolutional code is parameterised by the constraint length and code rate. The code rate, k/n, is a ratio of the number of bits into the convolutional encoder (k) to the number of output bits by the convolutional encoder (n) in a given encoder cycle. The constraint length parameter, *K*, denotes the "length" of the convolutional encoder. In this paper we use convolutional coding with a code rate of ½ and constraint length of 3. The generation matrix of g_1 = [111] and g_2 = [101] is use for the simulation results. The state transition of the encoder used for simulation is given in the fig.1.

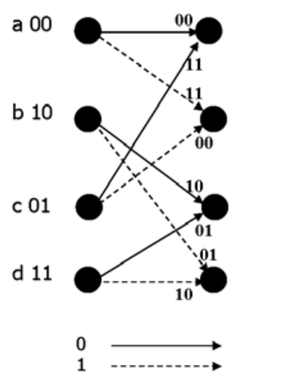

Fig. 1: State transition of a (3, 1, 2) convolutional encoder

The error correction capabilities of any error control coding depend upon the minimum Hamming distance from all zero sequence [8]. So the hamming distance needs to be as large as possible. The minimum Hamming distance of a convolutional encoder increases with an increase in the constraint length and it is tabulated for different code rate and constraint length [8]. So from the system performance view, the constraint length should be as large as possible but complexity of decoding increase exponentially with increase in the constraint length. So there is always compromise between the complexity and performance. It is difficult to ascertain exact Hamming distance of convolutional encoder so the error performance of convolutional coding is given in term of upper bound and the upper bound for error is given by [8] :

$$P_b < \left.\frac{\partial T(D,I)}{\partial I}\right|_{I=1,\, D=\sqrt{4P_{se}(1-P_{se})}} \tag{1}$$

where, *T*(*D*,*I*) is the transfer function of encoder and *Pse* is the bit error probability .

IV. Convolutional Coded DPIM

The DPIM (1GS) has a single pulse followed by at least an empty slot. Provided that the initial state of the encoder is ‘a’, the output sequence always start with {11 10 11} and the final state of the encoder after each symbol is ‘a’ except in the case when the input data has a decimal value of {0} which will produce output sequence {11 10}. But for the symbols preceded by decimal equivalent of ‘0’, the encoder initial state will not be in the state ‘a’ so that output will be different from {11 10 11…}. This situation can be avoided using DPIM (2GS). Every symbol in DPIM (2GS) has a pulse followed by at least two zeros. So the memory is cleared so that the encoder will always be in state ‘a’ at the starting of every symbol. The convolutional coded DPIM with two guard slots starts with {11 10 11} for any symbol, followed by a number of zeros depending upon the symbols as shown in Table 1. Therefore, it is easier to decode the convolutional coded DPIM (2GS) compared with DPIM (1GS) and hence the hardware complexity can be reduced. We will us the symbol CC-DPIM in further description to represent the convolutional coded DPIM.

We have developed an algorithm to decode CC-DPIM (2GS) based on determining the header of each symbol {11 10 11} by generating the header {11 10 11} and comparing it with the received sequences pair by pair. If the following sequence matches the structure {...00 11 10 11 00 ......}, it means transmission had no errors. If not then errors can be detected and corrected by comparing the current pair and a number adjacent pairs with a lookup table of all the allowed cases. The minimum number of pairs to make correct comparison is three pairs because the header of each symbol is {11 10 11}. If more than 3 pairs are taken into account, the error correction will be more accurate at the cost of slower decoding speed. The minimum distance of the convolutional encoder represented in fig. 1 is 5, so at least two errors can be corrected depending upon the decoding algorithm. The advantage of using the DPIM (2GS) over DPIM with no or one guard slot is the CC-DPIM decoding is much easier and there is not necessity of using the Viterbi algorithm.

Table 2: Lookup table of all correct DPIM(2GS) convolutional symbols for sequences of 5 pairs

i-1	**i**	**i+1**	**i+2`**	**i+3**	**i-1**	**i**	**i+1**	**i+2`**	**i+3**
00	00	00	00	00	11	00	00	00	00
00	00	00	00	11	11	00	00	00	11
00	00	00	11	10	11	00	00	11	10
00	00	11	10	11	11	00	11	10	11
00	11	10	11	00	11	11	10	11	00
00	11	10	11	11	11	11	10	11	11

To show how errors can be detected and corrected, we assume the following pairs: {00 **00** 10 11 00} are received. Here comparing this sequence with the standard {00 **11** 10 11 00} makes it easy to detect that an erasure error occurred changing the second pair from the expected {11} to {00}. Table 2 shows a lookup table of all correct DPIM(2GS) convolutional symbols for the case of comparing sequences of 5 pairs (i.e. 10 slots).

Unlike Viterbi decoding where there necessity of viewing large number of received bits before making decisions and probability of having many paths with same Hamming distance from the received sequences, there is only two paths to choose in DPIM(2GS) system making decoding a lot easier. Moreover the decision can be made in every 10 received bits saving a lot of memory. Hence the decoding is much easier because of finite number of possible path to select from and also small viewing window is enough for making decision.

The probability of error for the convolutional DPIM (2GS) depends upon the slot error rate of the uncoded DPIM (2GS). The detailed analysis of slot error probability of DPIM is found in [3]. So the slot error rate of the convolutional DPIM can be calculated using (1) by replacing P_{se} by slot error probability of DPIM system.

V. **Results and Analysis**

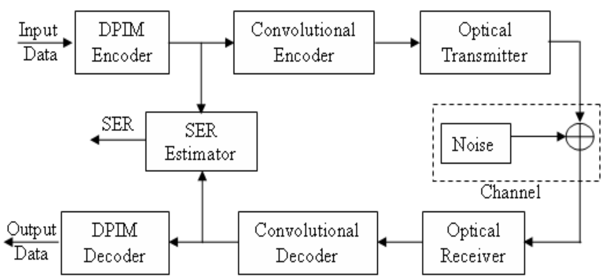

Fig. 2: Block diagram of simulation system

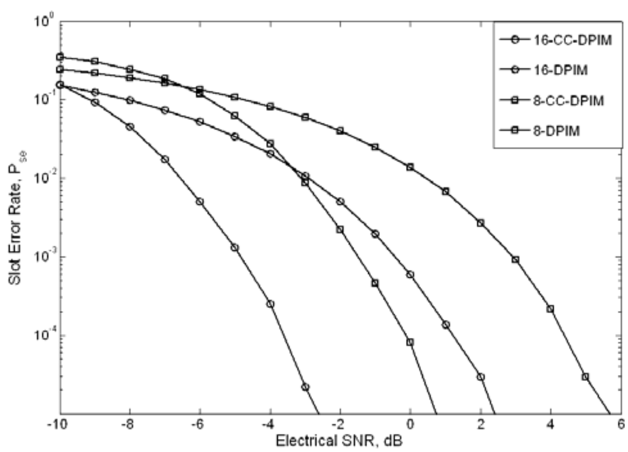

Fig. 3: The SER against the SNR for standard

The block diagram of the system used in the simulation is given in Fig. 2. The input binary data are converted to DPIM symbols, which then are passed to a convolutional encoder. Optical transmitter converts the electrical signal to optical signal. Then the optical signal is transmitted through a channel with white Gaussian noise. The optical receiver (photodiode) received the signal and passes it to the matched filter which is followed by a sampler and threshold detector to regenerate the signal which then is decoded using a convolutional decoder. The DPIM decoder converts the output of the convolutional decoder to DPIM signal which will be converted to binary data using a DPIM decoder. The SER estimator compares the transmitted and decoded DPIM symbols and calculates the slot error rate.

Because of computational power, we generate a 10^5 binary random bits and convert them to DPIM symbol according to the number of guard slots allocated. We assume that the channel is ideal with no bandwidth limitation; therefore, we simulate it by adding a white Gaussian noise to the transmitted signal. The noise has a one-sided

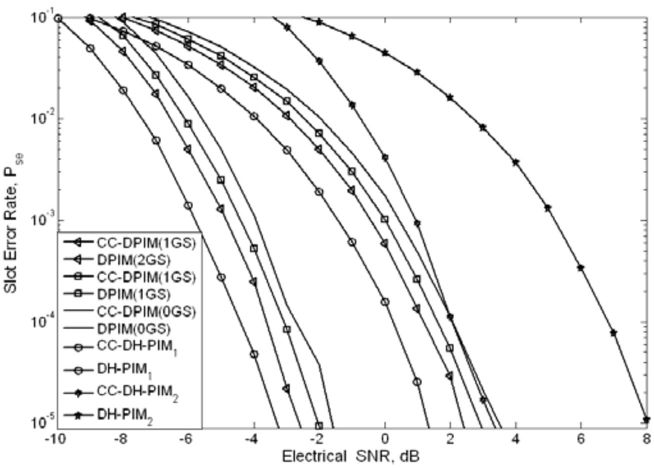

Fig. 4: Comparison of CC-DPIM (2GS) with theCC-DH-PIM, CC-DPIM (1GS) and CC-DPIM (0GS)for M=4..

power spectral density of $\eta = 2q_e I_b$, where q_e is the charge of electron and the background noise current I_b is set to 200 μA [11]. The simulation results of the slot error rate (SER) of CC-DPIM(2GS) against electrical signal to noise ratio (SNR) is shown in Fig. 3 for data rate of 1 Mbps for different bit resolutions. CC-DPIM(2GS) shows a very good improvement in slot error rate compared with standard DPIM(2GS). . CC-DPIM(2GS) has a code gain of more than 4dB at SER of 10^{-4} for all the cases compared to the standard DPIM(2GS).

Figure 4 compares the performance CC-DPIM(2GS) with convolutional coded DH-PIM, DPIM(0GS) and DPIM(1GS). CC-DPIM(2GS) shows an improvement over the CC-DPIM(1GS) and CC-DPIM(0GS). A CC-DPIM(2GS) provides an improvement of about 1 dB compared with CC-DPIM(0GS) and 0.5 dB compared with CC-DPIM(1GS) for slot error rate of 10^{-4}. The performance of CC-DPIM(2GS) is very close to the CC-DH-PIM_1, 1dB more SNR is required to get same SER for previous one. Therefore, the CC-DPIM(2GS) provides a good improve in the performance of the system and it also can reduce the hardware complexity by reducing the number of trellis paths and length of decision window.

IV. Conclusions

DPIM with convolutional coding has been investigated using different guard bands. CC-DPIM(2GS) showed improvement in the system performance over CC-DPIM(0GS) and CC-DPIM(1GS). A look-up table for decoding the CC-DPIM(2GS) has been presented. It is observed that code gain of almost 1dB is achieved compared to CC-DPIM(0GS) and gain of more than 0.5 dB compared to DPIM(1GS). The DPIM(2GS) can reduce the hardware complexity greatly because it limits the possible trellis path to 2, and hence reduce the cost of system.

Reference:

[1] J.M. Kahn and H. R. Barry,” Wireless infrared communications”, *Proc. of the IEEE*, 85(2), Feb 1997, pp. 265-298.

[2] F. Parand, G. E. Faulkner, D. C. O'Brie, “Cellular tracked optical wireless demonstration link*”, IEE Proceedings on Optoelectronics,* vol. 150(5), pp. 490 – 496, Oct. 2003.

[3] U. Sethakaset and T. A. Gulliver,” Differential Amplitude Pulse-Position Modulation for Indoor Wireless Optical Communications”, EURASIP *Journal on Applied Signal Processing,* pp. 3–11, 2005.

[4] A. R. Hayes: “Digital pulse interval modulation for indoor optical wireless communication systems", PhD thesis, Sheffield Hallam University, UK, 2002.

[5] N. M. Aldibbiat: "Optical wireless communication systems employing dual header pulse interval modulation (DH-PIM)", PhD thesis, Sheffield Hallam University, UK, December 2001.

[6] H. Park and J. R. Barry ,”Trellis-Coded Multiple-Pulse-Position Modulation for Wireless Infrared Communications” , *IEEE transactions on communications*, vol. 52, no. 4, April 2004.

[7] A. M. Michelson and A. H. Levesque: “Error Control Techniques for Digital Communication*”, John Wiley & Sons*, 1985.

[8] J. G. Proakis,”*Digital Communications* “, McGraw-Hill, Inc., New York 1995, pp 470-492.

[9] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, IT-13, April 1967, pp. 260-269.

[10] M. J. N. Sibley,”Optical Communications”,*The Macmillan Press LTD*, second edition, pp191-196, 1995.

[11] A. J. C. Moreira, A. M. R. Tavares, R. T. Valadas and A.M. D Oliveira Duarte: “Modulation methods for wireless infrared transmission systems: performance under ambient light noise and interference”, Proc. SPIE, USA, Oct. 1995, Vol. 2601, pp. 226-237.