Interleaver-Division Multiple Access on the OR Channel

M. Griot, A. I. Vila Casado, W.-Y. Weng, H. Chan, J. Basak, E. Yablonovitch, I. Verbauwhede, B. Jalali, and R. D. Wesel

Dept. of Electrical Engineering, University of California

Los Angeles, CA 90095

{mgriot,avila,wenyen,herwin,juthika,eliy,ingrid,jalali,wesel}@ee.ucla.edu

Abstract— The OR channel provides an unusual opportunity where single-user decoding permits operation at about 70% of the full multiple-access channel sum capacity. The interleaver-division multiple access technique applied in this paper can approach that performance with turbo and LDPC solutions. The current paper focuses on very low latency codes with simple Viterbi decoding, intended for very high speed (gigabits per second) applications. Namely, it focuses on nonlinear trellis codes that provide about 30% of the full multiple-access sum capacity at high speeds and with very low latency. These trellis codes are designed specifically for the Z-channel that arises in a multiple-user OR channel when the other users are treated as noise. In order to optimize the sum-capacity of the OR-MAC, the trellis code transmits codewords with a ones density much less than 50%. Also, a union bound technique that predicts the performance of these codes is presented. Results from simulations and an award-winning FPGA implementation are shown.

I. Introduction

Completely uncoordinated transmission on the OR-MAC is theoretically possible with the same efficiency as TDMA if joint decoding is employed. Joint decoding can be simplified to sequential decoding if the ones densities of each transmitter are carefully controlled [1], but this level of coordination is not qualitatively different from assigning time slots.

These schemes are fully efficient in that one useful bit of information may be transmitted with every transmitted bit. However, completely uncoordinated transmissions using interleaver division multiple access (IDMA)[2][3] and simple decoding that treats other users as noise is an attractive alternative. Surprisingly, in the multiple-user OR channel, this relatively low-complexity approach can theoretically achieve about 70% of full efficiency.

This paper presents an uncoordinated multiple access system employing IDMA in which the other users are treated as noise. To allow decoding at very high speeds in the near future, this paper investigates trellis codes which operate in a range of 30% of full efficiency. In other work, we are also designing turbo solutions, which, with higher complexity and latency, approach the 70% limit.

Section II reviews uncoordinated multiple access in the OR channel. Section III presents the design of nonlinear trellis coded modulation (NL-TCM) for this application. Section IV presents a transfer function bound for NL-TCMs operating on

the Z-Channel. Section V presents performance results, and Section VI concludes the paper.

II. Uncoordinated Multiple Access in the OR Channel

In the multiple access OR channel, if all users transmit a zero, then the channel output is a zero. However, if even one user transmits a one, then the channel output is a one. The information-theoretic capacity region of this channel is the triangle of all rate pairs with a sum-rate (the sum of all the rates of the system) less than 1. As discussed above, this capacity may be achieved with time-division multiple access, joint decoding of all the transmitted sequences, or sequential decoding if transmitted ones densities are carefully controlled. All of these solutions require either coordination of all users or a very complex decoder that is not currently feasible at gigabit per second sum rates.

In very high rate applications where joint decoding and successive decoding are unavailable for complexity reasons, the other users must be treated as noise. This transforms the OR channel into the Z-Channel shown in Fig. 1. If we assume that all users have the same transmitted ones density, the transition probability is a function of the same transmitted ones density employed by the desired user. Though not required, we maintain this assumption throughout the paper.

The maximum theoretical sum-rate when treating the other users as noise (assuming each user employs the same ones density) decreases as the number of users increases. However, it is always lower bounded by $\ln 2 \simeq 0.6931$. This is a relatively small loss in rate for the substantial reduction in complexity. In order to be able to achieve this maximum theoretical sum-rate, the optimal ones density of each individual user decreases as the number of users increase. For example, the optimal density of ones is $p_1 \simeq 0.2864$ for 2 equal-rate users, $p_1 \simeq 0.1080$ for 6 equal-rate users, and $p_1 \simeq 0.0558$ for 12 equal-rate users.

On the other hand, when maintaining equally likely ones and zeros ($p_1 = 0.5$) the maximum theoretical sum-rate rapidly decreases to zero with the number of users. For example, the maximum achievable sum-rate is less than 10% for 6 users and less than 1% for 10 users. Poor performance of the $p_1 = 0.5$ case demonstrates that codes with low ones densities are a requirement for this application.

One successful approach for uncoordinated multiple-access is Interleaver-Division Multiple-Access (IDMA) [2][3]. With
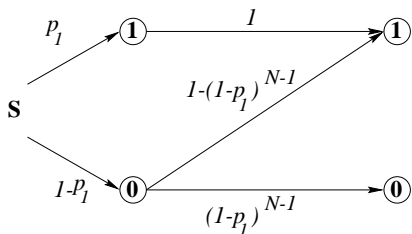
Fig. 1. Z-channel resulting from the OR-MAC channel when other users are treated as noise.

IDMA, every user has the same channel code, but each user's code bits are permuted using a randomly drawn interleaver, unique with extremely high probability. The receiver is assumed to know the interleaver of the desired user. With IDMA in the OR-MAC, a receiver should see the desired signal corrupted by a memoryless Z-channel. We performed simulations comparing an NL-TCM code under two channels: 1) a 6-user OR-MAC channel using IDMA and 2) the equivalent Z-channel that the receiver would see if the errors were not generated by codewords but by random errors. The performance was the same. Thus, in the context of IDMA, the remaining challenge is the design of a good code with the desired ones density.

III. NL-TCM WITH CONTROLLED ONES DENSITY

Papers appearing since the 1950's have addressed the problem of designing codes with $p_1 = 0.5$ for the Z-channel. See [4] for a unified account on such codes and [5] for the most recent advances in this field. Only recently there has been work on LDPC codes with an arbitrary density of ones, see [6] and [7]. This manuscript is the first to our knowledge to address the design of trellis codes with an arbitrary density of ones for the Z-Channel.

In this section, we present a design technique for trellis codes for the Z-channel with an arbitrary ones density p_1. Our goal is to maximize the minimum directional Hamming distance (a metric we'll define below) between codewords, and the rates considered will be of the form $1/N$. We use a conventional feed-forward encoder in order to determine the branches of the trellis, but instead of using generator polynomials to compute the output of each branch, a non-linear table-lookup will directly assign the output values.

A. Directional Hamming Distance

In the Z-Channel, a transmitted 1 will always induce a received 1. Thus, to make a decoding error, the decoder must see ones in all the bit positions where the incorrect codeword has ones. Let us define the directional Hamming distance $d_D(c_1, c_2)$ as the number of positions at which c_1 has a 0 and c_2 a 1.

Consider the Z-Channel with a probability of 0-to-1 transition less than 0.5. If two codewords c_1 and c_2 have different Hamming weights, the codeword with the smaller Hamming weight (denote it c_2) will never be incorrectly decoded by a maximum likelihood (ML) decoder when the code with the larger Hamming weight (c_1) is transmitted. Then, when considering the pairwise behavior of these two codewords, the only distance that matters is $d_D(c_2, c_1)$. On the other hand, if both codewords have the same Hamming weight, the directional Hamming distances are equal and errors can be made in either direction. Note that in any case, the directional distance that matters in such cases is the larger of the two.

Thus a proper definition for the pairwise design metric should be the maximum pairwise directional Hamming distance:

$$d(c_i, c_j) = d(c_j, c_i) = \max[d_D(c_i, c_j), d_D(c_j, c_i)] \quad (1)$$

This metric for the Z-Channel is well known, appearing in [4] and [5] among other papers.

B. Pessimistic definition of distance

Due to its non-linearity, this definition of distance cannot be applied branch-wise, since it is impossible to tell from an individual branch which codeword will end up having more Hamming weight. For that reason, we will use a pessimistic definition of distance for our trellis code design, considering both directional distances. Namely, the safest definition of branch-wise distance between any two branches b_i and b_j would be

$$d_p = \min[d_D(b_i, b_j), d_D(b_j, b_i)], \quad (2)$$

which is the pessimistic branch-wise metric that will be maximized in our design.

With this branch-wise metric, codewords with equal Hamming weights produce larger values than codewords with different Hamming weights, so we will assign output values to the trellis branches with as similar Hamming weight as possible, preferably equal.

C. NL-TCM Code Design

As we mentioned before, the code design consists of assigning output values to the branches of the trellis. Those outputs have to maintain the desired average density of ones p_1. We will present an extension to the Ungerboeck's rules [8] intended to maximize the minimum pessimistic distance $d_{\min}$ using the pessimistic branch-wise metric introduced in section (III-B).

1) *Choosing the Hamming weight of the branches:* The first step in the design is to assign Hamming weights (W_h) to each of the branches. For a rate of $1/N$, using a 2^ν-state encoder ($B = 2^{\nu+1}$ branches), and an optimal ones density of p_1, there should be B_s branches with Hamming weight $s = \text{floor}(p_1 \cdot N)$ and $B_{s+1} = B - B_s$ branches with Hamming weight $s + 1$, where B_s should be chosen to minimize the deviation from the desired ones density:

$$\text{deviation} = |p_1 \cdot N - (B_{s+1} \cdot (s + 1) + B_s \cdot s)/B|.$$

2) *Choosing all branches to have distance of at least 1 between each other:* If possible, it would be desirable to have all branches have a pessimistic distance (d_p) of at least 1 between each other.

When all branches have the same weight $W_H = s$, there are $\binom{N}{s}$ branch labels with $d_p \geq 1$ between each other.
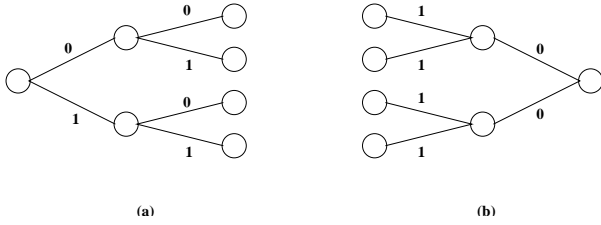
Fig. 2. (a) Four paths that start on the same state in two trellis sections. (b) Four paths that arrive to the same state in two trellis sections. Branches are labeled with the input bits that induce traversal of the branch.

For example, for $N = 17$ and $p_1 = 1/8$, and choosing all branches to have $W_H = 2$, we can have $\binom{17}{2} = 134$ different branches, so if we wanted to have all branch labels to be different so that $d_p \geq 1$ always, we could use a trellis with 64 states (represented by 6 bits) and 128 branches. The minimum number of branches that two paths can differ in a 2^ν-state encoder is $\nu + 1$. Thus, if each section adds at least 1 to the pessimistic distance, then $d_{\min} \geq \nu + 1$.

If repeated output values are necessary, often some branches can be strategically chosen to be equal while still maintaining $d_{\min} \geq \nu + 1$.

3) Ungerboeck's rule: Once we choose the weights of the branches, and the possible output values that will make all branches different (or almost all, allowing some to be equal as explained in the previous section), we have to assign output values to branches. One first idea is to apply Ungerboeck's idea of maximizing the distance between splits and merges[8]. The Hamming weights of the branches are at least s so it can be guaranteed that all splits and merges have a pessimistic distance of at least s between each other. Thus, applying Ungerboeck's rule a $d_{\min} \geq 2s + \nu - 1$ can be achieved.

4) Extending Ungerboeck's rule into the trellis: One can extend Ungerboeck's rule more deeply into the trellis, and maximize not only the distance between splits, and the distance between merges, but the distance between the 4 branches coming from a split in the previous trellis section, or the 8 branches coming from a split two sections before, and so on. One can do the same with the merges moving backwards in the trellis. Notice that by maximizing the distance between the 8 branches coming from a split two sections before, we are also maximizing the distance between all 4 branches coming from a split a trellis section before, and all splits. The same idea applies to the merges. If we move h sections from a split, and g sections from a merge, our new bound for the minimum distance is $d_{\min} \geq s \cdot (h+g) + \nu + 1 - (h+g)$. Fig. 2 shows the trellis branches involved with h=2 and g=2.

The largest possible values of h and g are given by the maximum number of output values that can have maximum distance between each other. Let us explain this in more detail.

First, compute the number of branches that need to have maximum distance between each other to cover h sections from a split and g sections backwards from a merge. From the splitting point of view, there will be groups of 2^h branches that need to have maximum distance between each other. From the merging point of view, there will be groups of 2^g. Each

branch belongs to one group of 2^h and one group of 2^g, so each branch has to have maximum pessimistic distance with $2^h + 2^g - 2$ other branches.

Second, compute the maximum possible number of branches of maximum pessimistic distance between each other. Using the example where $N = 17$ and all the branches have Hamming weight $s = 2$, it is possible to have floor($\frac{N}{s}$) = 8 branches with maximum distance between each other. Let us denote this number as T.

Then, the constraints are (1) $2^h \leq T$, (2) $2^g \leq T$ and (3) each branch has to belong to one group of 2^h and one group of 2^g. If one chooses h and g such that $2^h + 2^g - 2 \leq T$, all 3 constraints can be satisfied.

5) Designing for a very low target ones density: If the optimal ones density $p_1 \leq 1/2^{\nu+1}$, all the branches can be chosen to have maximum distance between each other (we can choose the branch labels so that for any particular position of the output, there is at most 1 branch that has a 1 in that position).

For such low densities of ones, the design becomes straightforward. Compute the Hamming weight of each branch as explained in section III-C.1, and for each branch, add ones in positions that aren't used in previous branches.

IV. TRANSFER FUNCTION BOUND FOR NL-TCM CODES

Ellingsen [9] provided a combinatorial expression for an upper bound on the BER of linear block codes over the Z-channel under ML decoding. For convolutional codes assuming binary PAM or QPSK, Viterbi [10] introduced an analytical technique using generating functions to provide a union bound on the BER of convolutional codes. The technique is based on a 2^ν-state diagram for the convolutional encoder. In the case of general trellis codes where high level constellations introduce nonlinearity, Biglieri [11][12] generalized Viterbi's algorithm by using the product state diagram with $2^{2\nu}$-states. Biglieri's algorithm can be applied to non-linear trellis codes over the Z-channel with modifications on the pairwise error probability measure.

Let α be the probability of a zero-to-one transition in the Z channel. Consider the whole binary transmitted sequence, X^n, and any other valid codeword, $\hat{X}^n$. The pairwise error probability of decoding X^n into $\hat{X}^n$ under ML decoding is

$$P_e(X^n \rightarrow \hat{X}^n) =$$
$$\begin{cases} \frac{1}{2} \cdot \alpha^{d_D(X^n, \hat{X}^n)} & , W_H(X^n) = W_H(\hat{X}^n) \\ \alpha^{d_D(X^n, \hat{X}^n)} & , W_H(X^n) < W_H(\hat{X}^n) \\ 0 & , W_H(X^n) > W_H(\hat{X}^n) \end{cases}$$

where $d_D(\cdot, \cdot)$ denotes the directional distance. If we consider the sequence pair $(X^n, \hat{X}^n)$, the error probability of transmitting one sequence and decoding the other is

$$P_e(X^n \rightarrow \hat{X}^n) + P_e(\hat{X}^n \rightarrow X^n)$$
$$= \alpha^{\max(d_D(X^n, \hat{X}^n), d_D(\hat{X}^n, X^n))} \quad (3)$$
$$\leq \frac{1}{2}[\alpha^{d_D(X^n, \hat{X}^n)} + \alpha^{d_D(\hat{X}^n, X^n)}]$$

Therefore, if $P_e(X^n \to \hat{X}^n)$ is replaced (not always upper-bounded) by $\frac{1}{2}\alpha^{d_D(X^n,\hat{X}^n)}$ for all the codewords X^n and $\hat{X}^n$, the transfer function bound technique can be readily applied to the NL-TCM to yield a valid upper bound because of the additive property of the directional distance.

As in [11], the product state diagram consists of state pairs, (s_e, s_r), where s_e is the encoder state and s_r the receiver state. Following Biglieri's notation, the product states can be divided into two sets, the good states denoted by S_G and the bad states denoted by S_B defined as

$$S_G = \{(s_e, s_r) \mid s_e = s_r\}, \ S_B = \{(s_e, s_r) \mid s_e \neq s_r\}$$

By suitably renumbering the product states, we get the transition matrix

$$S(W,I) = \left[\begin{array}{c|c} S_{GG}(W,I) & S_{GB}(W,I) \\ \hline S_{BG}(W,I) & S_{BB}(W,I) \end{array}\right] \quad (4)$$

Where the $N \times N$ matrix $S_{GG}(W,I)$ accounts for the transitions between good product states, the $N \times (N^2 - N)$ matrix $S_{GB}(W,I)$ accounts for the transition from good product states to bad product states, and so forth. N is the number of encoder states 2^ν. For each transition in the product state diagram, $S_1 \to S_2$, the branch is labelled by

$$p(S_1 \to S_2)W^{d_D(x_e,x_r)}I^{d_H(u_e,u_r)} \quad (5)$$

where u_e and x_e denote the input and output word for the encoder states respectively. Similar for the received states. $d_H(\cdot,\cdot)$ denotes the Hamming distance. Then the transfer function $T(W,I)$ becomes

$$T(W,I) = p_s\{S_{GG} + S_{GB}(I - S_{BB})^{-1}S_{BG}\}\mathbf{1} \quad (6)$$

where $p_s = [\frac{1}{N}\frac{1}{N}\cdots\frac{1}{N}]$ is the probability distribution of the encoder states and $\mathbf{1} = [11\cdots1]^T$. The BER bound is computed as

$$BER \leq \frac{1}{2} \cdot \frac{1}{k} \cdot \left.\frac{\partial T(W,I)}{\partial I}\right|_{W=\alpha,I=1} \quad (7)$$

V. Performance Results

We have tested the NL-TCM performance in an uncoordinated OR multiple access channel, where every user treats the others as noise, and all users transmit with the same density of ones p_1, as explained in Section II.

A. NL-TCM for 6-user OR-MAC

Fig. 3 shows the BER of various 64-state NL-TCM codes designed to work in a 6-user OR-MAC channel, along with their theoretical transfer function bounds. The densities of ones are close to 0.108, the density needed to achieve capacity for a 6-users OR-MAC when treating other users as noise. Simulations have been programmed in C for rate-1/17 NL-TCM code with $p_1 = 2/17$, rate-1/18 NL-TCM code with $p_1 = 1/8$ and rate-1/20 NL-TCM code with $p_1 = 1/8$.
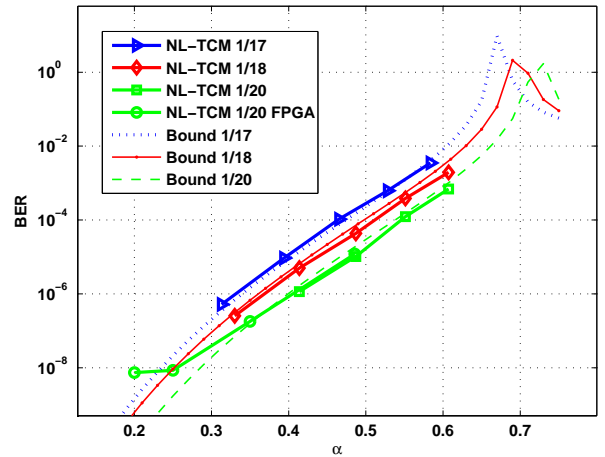

Fig. 3. BER of NL-TCM codes versus crossover probability (α)

It can be observed that that the transfer function bounds are tight in all three cases. The transfer function bound is an upper bound on the expectation assuming an infinite decoder depth, and it is not unexpected for these simulations to be slightly above the bound, since the margin is within the usual variation of a simulation around the expectation.

Also, in order to prove that NL-TCM codes are feasible today for very high speeds (optical speeds), a hardware simulation engine was built on the Xilinx Virtex2-Pro 2V20 FPGA. The simulator implemented on this device had equivalent gate count of 360K gates and is able to simulate the rate 1/20 code at 70Mbps. This implementation, demonstrating the power of trellis-coded interleaver-division multiple access in an optical fiber local area network, received first prize at the 2006 student design contest [13] sponsored by the ACM Design Automation Conference and the IEEE International Solid State Circuits Conference.

Results for rate-1/20 NL-TCM code obtained in the FPGA testbed, are also shown in Fig.3. Due to design constraints, the hardware Viterbi decoder has implementation differences comparing to the software simulation. The two main differences are the traceback depth and the maximum path metric. Both of these values are set to infinity for software simulation. The hardware implementation has a traceback depth of 35 and a path distance metric of 20. These differences cause the deviation from the theoretical bound at low bit error rates. Future versions of the simulator will address these issues to provide better performance.

Finally, Fig. 4 shows the BER of these codes in terms of the number of users present in an OR-MAC.

B. NL-TCM for 100-user OR-MAC

As explained in section III-C.5, the design of NL-TCM codes becomes straight-forward for low enough density of ones. As an example, we have designed NL-TCM codes for the 100-user OR-MAC case. In that case, the optimal $p_1 \simeq 0.00691$. We designed codes for rates $1/400$ and $1/360$. Results are shown in Table I.
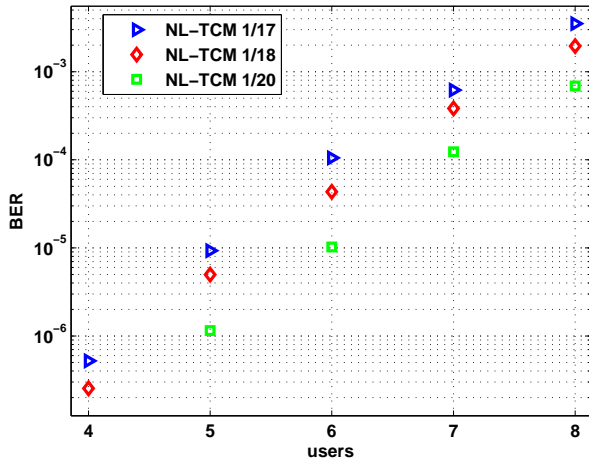
Fig. 4. BER of NL-TCM codes vs. number of users

TABLE I

BER OF NL-TCM FOR 100-USER OR-MAC

Rate	Sum-rate	p_1	α	BER
1/360	0.2778	0.006944	0.49837	$4.54 \cdot 10^{-6}$
1/400	0.25	0.006875	0.49489	$9.45 \cdot 10^{-7}$

C. Concatenation of NL-TCM code with a Block Code

In applications where a very low BER is required, the rate of the NL-TCM channel code would have to be very low. A better solution is found taking into account the distribution of the erred bits in a transmitted stream after the NL-TCM decoding. A high rate block code that can correct a small number of symbol errors can be attached as an outer code, dramatically lowering the BER.

A concatenation of the rate-1/20 NL-TCM code with a (255 bytes, 247 bytes) Reed-Solomon code has been tested for the 6-user OR-MAC scenario. The rate of this code is $(247/255) \cdot (1/20) \simeq 0.0484$. Results are shown in Table II. In this example, we are achieving almost 30% of full capacity, with a BER on the order of 10^{-10}. As an example, the OR channel can be used as a simple communications model that describes the multiple-user local area network optical channel with non-coherent combining. With the concatenation of NL-TCM codes with a Reed Solomon Code we are achieving a good part of the capacity (taking into account the very low latency of the code), with a suitable BER for optics, and a feasible complexity for today's technology at optical speeds.

VI. CONCLUSIONS

This paper addressed the problem of designing codes for the Z-channel along with an IDMA-based architecture that allows uncoordinated multiple access in the OR-MAC. In this architecture, the same code is used by every user, each of which randomly picks an interleaver to permute its coded bits.

TABLE II

BER OF RS+NL-TCM FOR 6-USER OR-MAC

Rate	Sum-rate	p_1	α	BER
0.0484	0.29	0.125	0.4652	$2.48 \cdot 10^{-10}$

For each user, every other user is treated as noise, in which case it 'sees' a Z-channel. We have addressed the problem of designing codes for this channel. These codes are required to have a relatively low ones density, requiring non-linear codes. In this work, we also required them to have low complexity to be computationally feasible at very high speeds today.

Non-linear trellis codes satisfy both requirements. A design criteria for NL-TCM codes was introduced, and bounds on their performance were computed. Furthermore, by concatenating these codes with high-rate block codes we can achieve a good part of the capacity of the channel with a very low BER and a very fast decoder. An important feature of this solution is that the achieved sum-rate remains basically unchanged as the number of users increases as shown in section V for the 6-user and 100-user case. This result makes this solution especially attractive compared to coordinated solutions for a large number of users, where coordination becomes an issue.

REFERENCES

[1] A. Grant, B. Rimoldi, R. Urbanke, and P. Whiting. Rate-Splitting Multiple Access for Discrete Memoryless Channels. *IEEE Trans. on Inform. Theory*, 47(3):873–890, Mar. 2001.

[2] Li Ping, K. Y. Wu, and L. Liu. A Simple, Unified Approach to Nearly Optimal Multiuser Detection and Space-time Coding. In *ITW 2002*, India, October 2002.

[3] Li Ping, Lihai Liu, and W. K. Leung. A Simple Approach to Near-Optimal Multiuser Detection: Interleaver-Division Multiple-Access. In *IEEE Wireless Comm. and Networking Conference*, pages 391–396, 2003.

[4] T. Kløve. Error correcting codes for the asymmetric channel. In *Dept. Mathematics, Univ. Bergen, Bergen, Norway, Tech. Rep.1809-0781*, 1995.

[5] Fang-Wei Fu, San Ling, and Chaoping Xing. New Lower Bounds and Constructions for Binary Codes Correcting Asymmetric Errors. In *IEEE Trans. on Inform. Theory*, volume 49, pages 3294–3299, March 2003.

[6] Amir Bennatan and David Burshtein. On the Application of LDPC Codes to Arbitrary Discrete-Memoryless Channels. In *IEEE Trans. on Inform. Theory*, volume 50, pages 417–438, March 2004.

[7] Edward A. Ratzer and David J.C. MacKay. Sparse Low-Density Parity-Check Codes for Channels with Cross-Talk. In *Proc. ITW 2003*, Paris, France, April 2003.

[8] G. Ungerboeck. Channel Coding with Multilevel Phase Signals. In *IEEE Trans. on Inform. Theory*, volume 28, pages 52–67, 1982.

[9] P. Ellingsen, S. Spinsante, and O. Ytrehus. Maximum Likelihood Decoding of Codes on the Asymmetric Z-channel. In *the 10th IMA International Conference on Cryptography and Coding 2005*.

[10] A. J. Viterbi. Convolutional codes and their performance in communication systems. *IEEE Trans. on Comm.*, 19, Oct 1971.

[11] E. Biglieri. High-Level Modulation and Coding for Nonlinear Satellite Channels. *IEEE Trans. on Comm.*, volume 32, May 1984.

[12] Y. J. Liu, I. Oka, and E. Biglieri. Error probability for digital transmission over non-linear channels with application to tcm. volume 36, Sep 1990.

[13] H. Chan, A. I. Vila Casado, J. Basak, M. Griot, W. Weng, R. Wesel, B. Jalali, E. Yablanovitch, and I. Verbauwhede. Demonstration of Uncoordinated Multiple Access in Optical Communications. In *the Student Design Contest organized jointly by the 2006 ACM Design Automation Conference and the 2006 IEEE International Solid-State Circuits Conference.* .