SPARSE GRAPH CODES ON A MULTI-DIMENSIONAL CDMA PLATFORM FOR APPLICATION IN MOBILE WIRELESS COMMUNICATION SYSTEMS

J.D. Vlok* and L.P. Linde**

* *Defence, Peace, Safety and Security, Council for Scientific and Industrial Research, Meiring Naudé Road, Brummeria, Pretoria, South Africa*
** *Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Lynnwood Road, Pretoria, 0002, South Africa*

Abstract: This paper presents the uncoded and coded multiuser error performance results of a novel *super-orthogonal* four dimensional code division multiple access communication platform under additive white Gaussian noise and multipath fading channel conditions. The communication platform employs root-of-unity filtered constant envelope complex spreading sequences with zero cross-correlation properties. The uncoded communication platform employs multi-layered-modulation to improve spectral efficiency compared with existing multi-level modulation techniques, and the coded system uses the multiple dimensions to transmit redundancy to improve error performance. Three classes of sparse graph coding schemes are evaluated on the multi-dimensional communication platform. The channel codes include a three dimensional block-turbo-code with extended Reed-Muller constituent codes, low-density parity-check codes and repeat-accumulate codes.

Key words: Block Turbo codes (BTC), complex spreading sequences (CSS), channel modelling, log-likelihood ratio (LLR), low-density parity-check (LDPC) codes, multi-layered modulation (MLM), Multi-dimensional (MD), repeat-accumulate (RA) codes, sparse graph channel coding

1. INTRODUCTION

Multi-layered modulation (MLM) can be used to transmit parallel data streams with improved spectral efficiency compared with single-layer modulation, providing data throughput rates proportional to the number of modulation layers at performances equivalent to single-layer modulation. Alternatively, multiple layers can be used to transmit coded information to achieve improved performance at throughput rates equivalent to a single layer system.

An MLM four-dimensional (4D) code division multiple access (CDMA) building block employing super-orthogonal complex spreading sequences (SO-CSS) is used as MLM platform [1] for the evaluation of three classes of sparse graph codes. The communication platform including the transmitter, receiver and complex spreading sequences is discussed in Section 2. The encoding and decoding structures of the block-turbo-code and details regarding the construction of the constituent code's trellis structure and decoding algorithm are considered in Section 3. Section 4 briefly considers low-density parity-check (LDPC) and repeat-accumulate (RA) codes and Section 5 provides details regarding the channel simulator configuration. Finally Section 6 contains error performance results and Section 7 concludes the paper.

2. MODULATION PLATFORM

This section considers the basic 4D modulation platform [1] assuming perfect carrier synchronisation and spreading code lock.

2.1 Four Dimensional Transmitter

The basic 4D transmitter structure is essentially a Quadrature Phase Shift Keying (QPSK) modulator combined with spreading multipliers using the CSS $c(t) = c_r(t) + jc_i(t)$ as shown in Figure 1. The input sequence $d(t)$ is split up into four parallel data streams $d_1(t)$ to $d_4(t)$ by an serial-to-parallel (S/P) converter after which the upper two streams ($d_1(t)$ and $d_2(t)$) and the lower two streams ($d_3(t)$ and $d_4(t)$) are processed identically but separately. Two orthogonal spreading codes are used to spread each stream group as follows: data streams $d_1(t)$ and $d_2(t)$ are respectively multiplied by orthogonal spreading codes $c_r(t)$ and $c_i(t)$ to obtain $x_1(t)$ and $x_2(t)$ which are summed to obtain the in-phase component $y_1(t)$. The quadrature component $y_2(t)$ is obtained likewise from data streams $d_3(t)$ and $d_4(t)$. After modulating the in-phase and quadrature components onto quadrature carriers, the modulated components $z_1(t)$ and $z_2(t)$ are summed to produce the transmitted signal carrying 4 data bits per channel use.

$$\begin{aligned} s(t) &= \left[d_t(t)c_r(t) + d_2(t)c_i(t)\right]\cos\left(w_c t\right) \\ &\quad + \left[d_3(t)c_r(t) + d_4(t)c_i(t)\right]\sin\left(w_c t\right) \end{aligned} \tag{1}$$

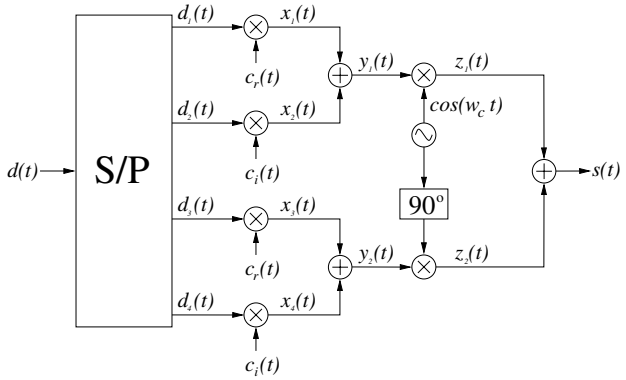

Figure 1: Four dimensional DSSS transmitter structure

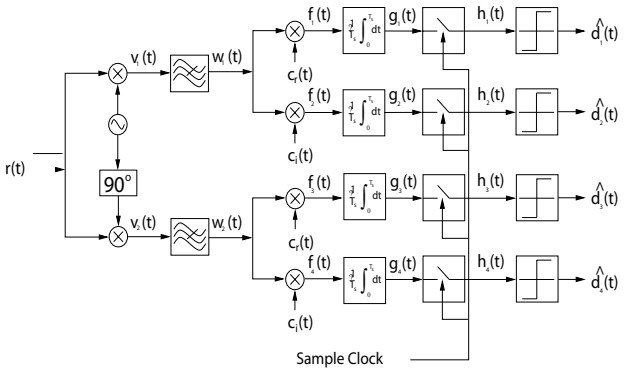

Figure 2: Four dimensional DSSS receiver structure

2.2 Four Dimensional Receiver

The 4D receiver is a correlation-type demodulator as shown in Figure 2. The received signal *r(t)* is demodulated and despread to obtain approximations $\hat{d}_1(t)$ to $\hat{d}_4(t)$ of the original data streams. The use of unique families of CSSs contributes to the significantly simplified MUI-free receiver structure.

2.3 Complex Spreading Sequences

The performance of the CDMA system presented in this section depends primarily on the correlation characteristics of the CSSs used (see [2] for a thorough correlation study). The base vectors of the multi-dimensional (MD) space are the CSSs and it is therefore important to employ CSSs with excellent cross correlation characteristics between the real and imaginary components. One class of CSSs fulfilling this condition is Zero Cross Correlation (ZCC) [3] Super-Orthogonal (SO; i.e. zero-cross correlation for all time shifts) Constant-Envelope (CE) CSSs derived from Park-Park-Song-Suehiro (PS) [4] CSSs using a non-linear root-of-unity (RU) filtering technique [5]. The CE characteristic of ZCC CSSs provides the output signal *s(t)* of the 4D modulator with near-constant instantaneous and constant average output power by virtue of the orthogonality of the basis functions. The primary advantage of the power efficiency is the modulator's ability to operate close to the 1 dB saturation point of the power amplifier of the transmitter improving the communication range and/or handset battery life.

3. BLOCK TURBO CODE

3.1 Encoding Structure

A 3D block turbo code (BTC) [6] with *(N,K) = (2K,K)* constituent block codes is used to form a *($4K^3$,K^3)* encoder structure shown in Figure 3, by mapping the code blocks $\bar{c}_1$ to $\bar{c}_4$ directly to the 4D transmitter inputs *$d_1(t)$* to *$d_4(t)$* by removing the S/P converter (see Figure 1). A 3D data block $\bar{x}$ is encoded by systematic row, column and depth encoders (1, 2 and 3 respectively) using row-to-column and row-to-depth block interleavers $\Pi_{1,2}$ and $\Pi_{1,3}$. The data and redundant information are transmitted simultaneously as a 4D signal $\bar{s} = [\bar{c}_0, \bar{c}_1, \bar{c}_2, \bar{c}_3]$ using SO-CSSs.

3.2 Decoding Structure

Correlation type demodulation (see Figure 2) is performed to calculate the soft value representing the kth; k = 1, . . . ,K^3, bit of the *n^{th}; n = 0, 1, 2, 3*, code block $\tilde{c}_n$ using the equation:

$$\tilde{c}_{n,k} = \sum_{l=1}^{L.N_s} \alpha_{k,l} r_{k,l} \psi_{n,l} \qquad (2)$$

With $\alpha_{k,l}$ and $r_{k,l}$ respectively the l^{th} instantaneous fading amplitude and received samples of the composite 4D signal containing the k^{th} group of parallel codebits. $\psi_{n,l}$ is the l^{th} sample of the orthonormal CSS base vector, distinguishing the n^{th} dimension of the 4D signal [1]. *L* is the CSS length and N_s the number of samples per bit. The 4D demodulator output $\bar{y} = [\tilde{c}_0, \tilde{c}_1, \tilde{c}_2, \tilde{c}_3]$ is iteratively decoded using 3 interconnected soft-input soft-output (SISO) decoding modules [6] as shown in Figure 4.

The output of SISO module *m* is a 3D cube $\Lambda_{E,m}$; *m = 1, 2, 3*, containing the extrinsic log-likelihood ratio (LLR) of each data bit x_k, computed using the equation [7]:

$$\Lambda_{E,m}(x_k) = \Lambda(x_k \mid r_m) - \Lambda_{A,m}(x_k) - \Lambda_c(r_m \mid x_k) \qquad (3)$$

with $\Lambda_{A,m}(x_k)$ the prior LLR of x_k and $\Lambda_c(r_m \mid x_k)$ the channel LLR at the input of SISO module *m* and $\mathbf{r}_m$ is the received code block consisting of $\Pi_{1,m}\tilde{c}_0$ and $\tilde{c}_m$. Each block interleaver $\Pi_{m,m'}$ performs a 3D rotation on the input code block to translate the format processable by SISO module *m* to *m*?. Each SISO module computes the posterior LLR $\Lambda(x_k \mid r_m)$ employing a bidirectional

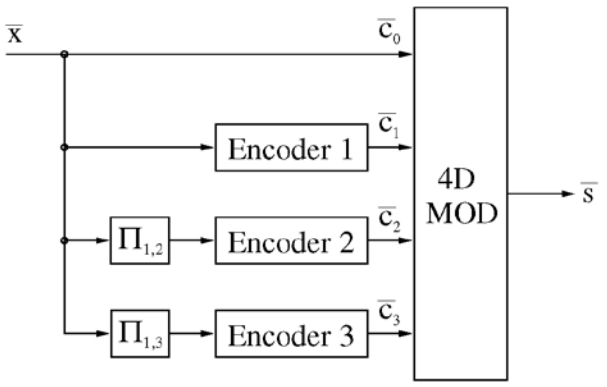

Figure 3: Encoder and 4D modulator structure

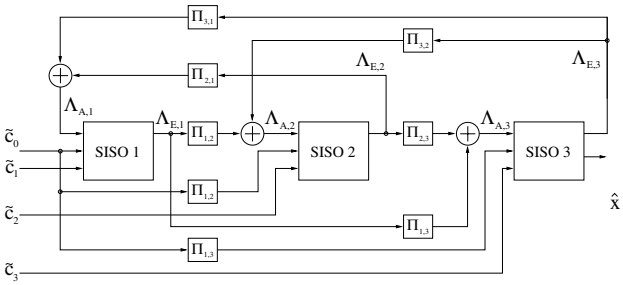

Figure 4: Three-dimensional block turbo decoder structure

Viterbi algorithm (VA) [8] with branch metric associated with x_k connecting states s and s' in the trellis:

$$BM_{s,s'}^{m,x_k} = \left(\tilde{c}_{n,k} - x_k\right)^2 - \frac{1}{2} x_k \Lambda_{A,m}(x_k) \qquad (4)$$

where n is determined by m, according to the structure shown in Figure 4. The estimated data block $\hat{x}$ is determined by performing hard decision quantisation on the posterior LLR of each data bit computed by SISO module 3.

3.3 Constituent Codes

Linear block codes namely binary Reed-Muller (RM) codes are used as constituent codes of the turbo coding scheme in this study. A soft-output bitwise decoding algorithm based on the trellis soft-output Viterbi algorithm (SOVA) is used to iteratively decode the BTC. Binary RM codes are chosen because of their relatively simple trellis structures.

Trellis Construction: The trellis structure of a linear block code can be constructed from its parity-check matrix [9]. The parity-check matrix of a linear *(N,K)* block code is given by:

$$H = \left[\bar{h}_1 \quad \bar{h}_2 \quad \cdots \quad \bar{h}_N\right] \qquad (5)$$

with $\bar{h}_i$ the i^{th} column vector of H with the index $i = 1, 2, \ldots, N$ also denoting the i^{th} trellis depth. Let $C_m = \left[c_{m,1}, \; c_{m,2}, \; \cdots, \; c_{m,N}\right]$ be the m^{th} valid codeword with $m = 1, 2, \ldots, 2^K$ and let the states in the trellis be denoted as $\bar{S}_{m,i}$ in column vector format like $\bar{h}_i$. By including $\bar{S}_{m,0}$, the trellis can be constructed using the following recursion formula [9, 10]:

$$\bar{S}_{m,0} = \bar{0} \qquad (6)$$

$$\bar{S}_{m,i} = \bar{S}_{m,i-1} + c_{m,i}, \bar{h}_i \qquad (7)$$

Note that the trellis construction formula given in Equation 7 requires knowledge of all valid codewords, though all information needed to construct the trellis is inherently available in H. A trellis construction procedure presented in [10] starts at $S_{m,0} = 0$ and considers both possible binary values for each code bit $c_{m,i} \in [0,1]$ resulting in multiple paths, valid and invalid, in the trellis. The invalid paths are then removed by an expurgation procedure, i.e. removing all paths from the trellis that don't end in the zero state in the trellis at $i = N$. An alternative trellis expurgation technique is presented in this section.

The parity check matrix of a systematic $RM_{1,3}$ or RM(8, 4, 4) code is given as:

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (8)$$

from which the unexpurgated forward trellis structure from trellis depth $i = 0$ to $i = N$ is constructed. Note that Equations 6 and 7 can be rewritten to describe the trellis in reverse order as:

$$\bar{S}_{m,N} = \bar{0} \qquad (9)$$

$$\bar{S}_{m,i} = \bar{S}_{m,i+1} + c_{m,i+1}, \bar{h}_{i+1} \qquad (10)$$

Equation 10 follows from the fact that $c_{m,i}\bar{h}_i + c_{m,i}\bar{h}_i = \bar{0}$ in Equation 7. The unexpurgated reverse trellis structure is constructed by moving from trellis depth $i = N$ to $i = 0$ using Equations 9 and 10. By combining the unexpurgated forward and reverse trellises, the expurgated trellis (Figure 5) is obtained directly. The expurgated trellis structure (shown as solid branches in Figure 5) is the union of branches of the unexpurgated forward and reverse trellises (shown as dotted branches in Figure 5 not ending in the zero state at either trellis depths $i = 0$ or $i = N$).

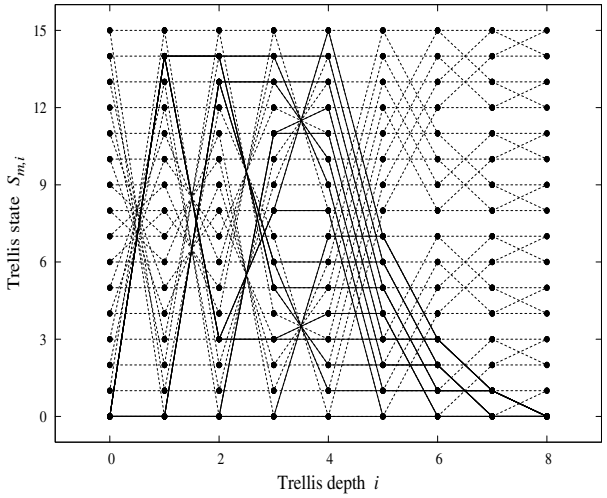

Figure 5: Unexpurgated (dotted branches) and expurgated (solid branches) trellis structure of the RM(8,4,4) code

Soft-Output Trellis Decoding: The reliability or posterior LLR of the transmitted information bits used in soft-output decoding algorithms can be expressed as [8, 11]:

$$\Lambda(x_k \mid r) = \ln\left[\frac{p(x_k = +1 \mid r)}{p(x_k = -1 \mid r)}\right] \quad (11)$$

$$= \ln\left(\sum_{l=1}^{L_t} e^{-PM_{l,k}^{x_k=+1}}\right) - \ln\left(\sum_{l=1}^{L_t} e^{-PM_{l,k}^{x_k=-1}}\right) \quad (12)$$

where L_t is the number of selected paths considered and $PM_{l,k}^{x=\pm 1}$ is the l^{th} path metric corresponding to $x_k = \pm 1$. Equation 12 follows from the fact that $p(x_k = \pm 1 | r)$ is the sum of all probabilities of occurrences of $x_k = \pm 1$ which correspond to the sum of probabilities of paths in the trellis with $x_k = \pm 1$.

To calculate the probability $p(x_k = \pm 1 | r)$ exactly, all possible paths in the trellis corresponding to $x_k = \pm 1$ must be considered and therefore $PM_{l,k}^{x=\pm 1}$ corresponds to the path metrics of all possible paths (not only ML paths) in the trellis.

Using the max-log approximation [7, 11] Equation 12 can be simplified as:

$$\Lambda(x_k \mid r) \approx \min_l \left\{PM_{l,k}^{x=-1}\right\} - \min_l \left\{PM_{l,k}^{x=+1}\right\} \quad (13)$$

$$= PM_{ML,k}^{x=-1} - PM_{ML,k}^{x=+1} \quad (14)$$

where $PM_{ML,k}^{x=\pm 1}$ is the path metric of the best candidate (ML) path with $x_k = \pm 1$ in each trellis depth.

The soft-output decoding algorithm used in this study is a SOVA-based or bidirectional soft-output decoder [7, 8]. The algorithm calculates an estimate of the posterior LLR defined in Equation 11 by performing a forward-backward Viterbi algorithm on an expurgated trellis structure such as the *RM(8, 4, 4)* trellis shown in Figure 5. The algorithm computes and stores the path metrics in each trellis depth by running through the trellis independently in both forward and reverse directions. $\Lambda(x_k \mid r)$ is then calculated using the forward and reverse calculated path metrics and the branch metrics corresponding to $x_k = \pm 1$ in each trellis depth.

The detailed forward-backward soft-output decoding algorithm used in this study is explained as follows:

- By running in the forward direction through the trellis according to Equation 7 starting at state $\overline{S}_{m,0}$ (see Equation 6), the forward path metrics are calculated as:

$$PM_{s,i}^{f} = \min_{s}\left(PM_{s',i-1}^{f} + BM_{s,s'}\right) \quad (15)$$

  where *(s, i)* indicates the trellis state $\overline{S}_{m,i}$ (as in Equations 6 and 7) in the i^{th} trellis depth and (*s*', *i*-1) indicates the trellis state in the previous trellis depth. $BM_{s,s'}$ is the branch metric of the branch connecting states *s* and *s*'. The path metric $PM_{s,i}^{f}$ is therefore the survivor path metric at state *s* up to trellis depth *i* in the forward direction. For each trellis coordinate *(s, i)* the survivor path metric should be stored.

- By running in the reverse direction through the trellis according to Equation 10 starting at state $\overline{S}_{m,N}$ (see Equation 9), the reverse path metrics are calculated as:

$$PM_{s,i}^{r} = \min_{s}\left(PM_{s',i+1}^{r} + BM_{s,s'}\right) \quad (16)$$

  where *(s, i)* indicates the trellis state $\overline{S}_{m,i}$ (as in Equations 9 and 10) in the i^{th} trellis depth and (*s*', *i*+1) indicates the trellis state in the next trellis depth. The path metric $PM_{s,i}^{r}$ is therefore the survivor path metric at state *s* up to trellis depth *i* in the reverse direction.

- For each trellis depth the best candidate path metric for $x_k = \pm 1$ is calculated using the equation:

$$PM_{ML,k}^{x=\pm 1} = \min_{x_k=\pm 1}\left(PM_{s,i}^{f} + PM_{s',i+1}^{r} + BM_{s,s'}^{x_k=\pm 1}\right) \quad (17)$$

where $BM_{s,s'}^{x_k=\pm 1}$ is the metric of the branch corresponding to $x_k = \pm 1$.

The posterior LLR $\Lambda(x_k \mid r)$ is finally calculated for each code bit using Equations 14 and 17.

The forward-backward decoding algorithm is used in each SISO decoding module in Figure 4 to calculate $\Lambda(x_k \mid r)$ with $k = 1, \ldots, N$ for each code bit in the received codeword. The extrinsic LLR $\Lambda_E(x_k)$ with $k = 1, \ldots, K$ is calculated for each information bit and passed to other SISO modules in the decoder using the posterior LLR $\Lambda(x_k \mid r)$ and Equation 3.

4. OTHER SPARSE GRAPH CODES

The two other classes of sparse graph channel coding schemes considered in this study are LDPC and RA codes, and are briefly considered in this section.

4.1 LDPC codes

Low-density parity-check (LDPC) codes [12] have received much research interest [13] and are powerful error-correcting codes outperforming turbo codes if the code length is large and the parity check matrix is designed correctly. In this study approximately regular LDPC codes constructed by random parity check matrix generation are used. Systematic sparse matrix encoding [14] is performed by writing the parity-check matrix as:

$$H = \begin{bmatrix} A & B \end{bmatrix} \tag{18}$$

with a non-singular sub-matrix **A** *(M × M)* and sub-matrix **B** *(M × K)*. The codeword can be written as $\bar{c} = [\bar{p} \quad \bar{x}]$ with $\bar{p}$ *(1×M)* the parity vector and $\bar{x}$ *(1×K)* the systematic information vector. From the code's null space and Equation 18 the following equation is obtained:

$$A\bar{p}^T + B\bar{x}^T = 0 \tag{19}$$

from which the parity vector can be calculated as:

$$\bar{p}^T = A^{-1}B\bar{x}^T \tag{20}$$

Decoding is performed using Pearl's belief propagation message-passing (MP) algorithm [15] on the code's factor graph. The algorithm takes as input the channel LLR $\Lambda_c(r_m \mid x_k)$ (see Equation 3) and provides as output an approximation of the posterior probability $p(t_n|r)$ for every transmitted bit.

4.2 RA codes

Repeat-Accumulate (RA) [15] codes are extremely simple Turbo-like [16] capacity-approaching codes, which allow linear-time encoding with performance results comparable to Turbo and LDPC codes. This study only considers the original non-systematic regular RA codes. RA codes can be decoded using a LLR MP algorithm on the code's factor graph, similar to decoding LDPC codes.

5. CHANNEL AND SIMULATION PLATFORM

AWGN and small-scale effects typically encountered in wide band mobile wireless communication systems, including frequency selective fading, are considered in this study. A channel model based on Clarke's model presented in [10, 17] (with a more accurate Doppler filter based on the design given in [18] - see [2] for a complete description of the channel simulator) is developed as part of the simulation platform to test and evaluate the performance of the CDMA system.

A multipath fading channel simulator with three statistically independent communication paths per user where each path has an unique Rician factor, Doppler spread and relative time delay similar to the model presented in [17] is used in the simulations. Each user's multipath channel has an exponentially decaying power delay profile as is typical in mobile wireless communications.

Shown in Figure 6 is the coded simulation platform. The data sequence *x(t)* originating from a random data generator, is encoded and then passed through an interfacing module to adapt the signal for transmission by the 4D transmitter. The code interface is an S/P module at the transmitter and a P/S module at the receiver. The channel encoder encodes a serial data sequence (or block) into a serial coded sequence and the 4D transmitter takes as input four parallel bit sequences given as four parallel estimated sequences at the output of the 4D receiver. It is therefore necessary to interface the encoder output to the transmitter input and receiver output to the decoder input. In the turbo coded system, the channel coding and interfacing modules are integrated into a single module by virtue of the turbo encoder and decoder structures. The channel state information (CSI, including the fading amplitude and phase) is extracted from the reference user's multipath fading channel simulator and used in the 4D demodulator. All other users' signals and AWGN are added to the reference user's signal before demodulation.

6. PERFORMANCE RESULTS

6.1 Block-Turbo Code

The uncoded and coded 4D communication systems are evaluated under AWGN and multipath fading conditions.

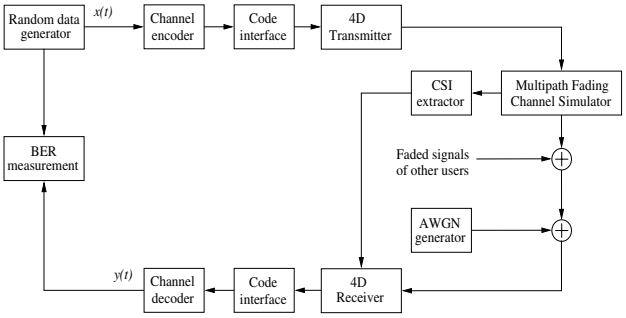

Figure 6: Coded simulation platform

Extended *RM* (12, 6) constituent codes are used to form a (864, 216) code with CSS parameters $L = 63$ and $N_s = 8$. Figures 7 and 8 show the AWGN and multipath fading error performance results respectively.

AWGN results of the BTC: Although the 4D uncoded system has data throughput equivalent to 16-QAM, the system maintains BPSK error performance under AWGN channel conditions, achieving a 4 dB gain relative to 16-QAM. The 3D BTC coded system has data throughput equivalent to BPSK and achieves a 6 dB coding gain over BPSK (about 5 dB from the Shannon limit) at P_e(bit) = 1×10^{-5}, after 5 decoding iterations. The results shown in Figure 7 are relevant for 1 to 5 users (the performance for 1 to 5 users lies on the same decoding iteration curve) with the number of users depending on the CSS length.

Multipath fading results of the BTC: Although a slight performance degradation (approximately 0.5 dB for E_b/N_0 larger than 8 dB) for 1 to 5 users is observed in multipath fading conditions for the uncoded system, practically identical multi-user results are achieved by employing the BTC. After 1 decoding iteration a 3 dB gain above the BPSK AWGN curve is evident and after 10 iterations an additional 2.5 dB is gained.

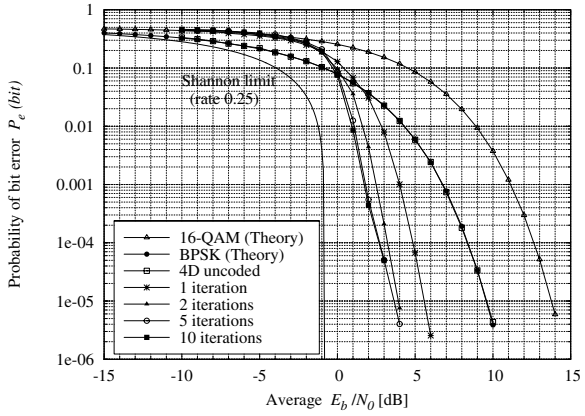

Figure 7: AWGN performance results for 1 to 5 CDMA users

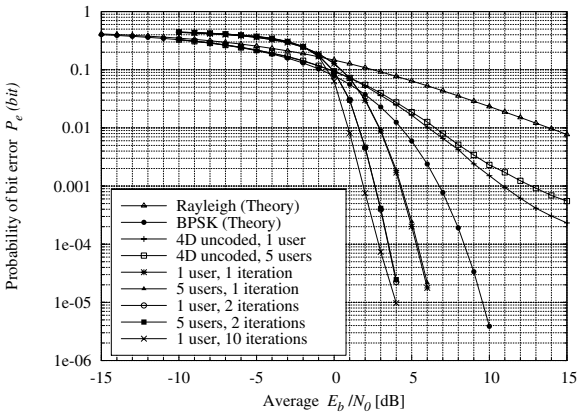

Figure 8: Multipath fading performance results for 1 to 5 CDMA users

6.2 Comparison of sparse graph codes

Comparing AWGN performance: Figure 9 shows the simulated AWGN error performance results of the uncoded 4D system, the 3D turbo code with the *RM* (8,4,4) constituent code (which is effectively a (256, 64) code) for 10 decoding iterations, the LDPC (256,64) code for 100 decoding iterations and the RA code for 100 decoding iterations. All curves shown are the single user error performance curves (which are identical to the performance curves for up to 5 CDMA users) using ZCC CSSs.

Comparing Multipath Fading Performance: Figure 10 shows the simulated multipath fading error performance results of the uncoded 4D system for 1 and 5 users, the 3D turbo code with the RM (8,4,4) constituent code for 1 and 5 users with 10 decoding iterations, the LDPC (256,64) code and the RA (256,64) code. The curves for the LDPC and RA codes shown in Figure 10 represent the performance curves for 1 to 5 users with 100 decoding iterations each.

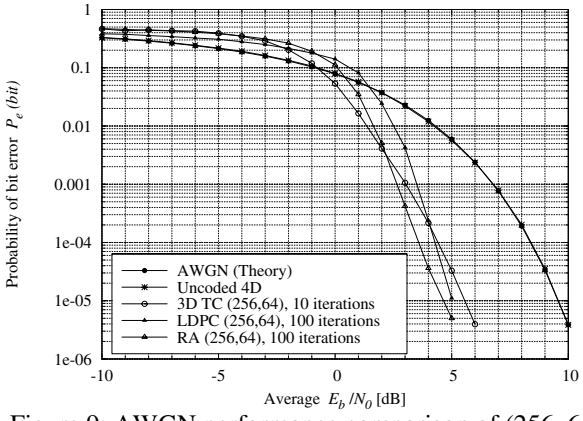

Figure 9: AWGN performance comparison of (256, 64) coding schemes

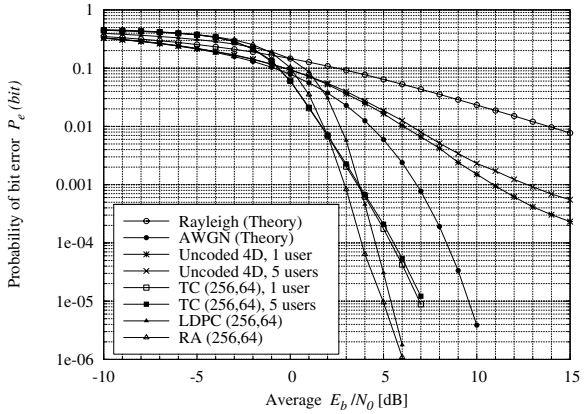

Figure 10: Multipath performance comparison of (256, 64) coding schemes

7. CONCLUSION

This paper presented the error performance results of the uncoded and coded 4D communication system presented in [1], showing BPSK error performance for all CDMA users under AWGN channel conditions. The three coding schemes were adapted to the 4D system and show comparable error performance results (within 2 dB), though at low E_b/N_0 values (around 0 dB) the turbo code performs a little better than the other two coding schemes and at 3 dB and higher the LDPC and RA codes outperform the turbo code. The RA code shows the best overall error performance. It is also clear that the turbo code requires less decoding iterations than the other coding schemes, though different decoding structures and algorithms (with different levels of complexity) are employed for each scheme. The 3D block-turbo-code (with RM(12,6) constituent codes) yielded an approximate 4 to 6 dB coding gain in AWGN and 3 to 6 dB coding gain in multipath fading channel conditions over BPSK at P_e(bit) = 10^{-5}, with decoding complexity of each constituent decoder equivalent to the bidirectional VA.

8. REFERENCES

[1] L. P. Linde and J. D. Vlok, "Power and spectrally efficient four dimensional super-orthogonal WCDMA building block for next generation wireless applications," *IEEE Communications Letters*, vol. 10, no. 7, pp. 519–521, July 2006.

[2] J. D. Vlok, "Sparse graph codes on a multi-dimensional WCDMA platform," Master's dissertation, University of Pretoria, South Africa, 2007.

[3] I. Pryra, L. P. Linde, and S. A. Swanepoel, "New family of constant envelope root-of-unity filtered complex spreading sequences with zero cross-correlation properties," *Proceedings of 6th AFRICON IEEE Conference*, vol. 1, pp. 299–304, October 2002.

[4] N. Suehiro, "A signal design without co-channel interference for approximately synchronized CDMA systems," *IEEE JSAC*, vol. 12, pp. 837–841, June 1994.

[5] L. P. Linde and M. Lötter, "Spread spectrum modulator and method," South African Complete Patent 96/0355, January 17, 1996, earliest priority claimed: ZA 94/9014, 14 November 1994.

[6] B. Yin and M. R. Soleymani, "Design and implementation of three dimensional block turbo codes," *Proceedings of 2003 IEEE Canadian Conference on Electrical and Computer Engineering - CCECE2003,* Montreal, Canada, pp. 1625–1628, May 2003.

[7] R. H. Morelos-Zaragoza, *The Art of Error Correcting Coding*, John Wiley & Sons Limited, Chichester, UK, first edition, 2004.

[8] W. Feng and B. Vucetic, "A list bidirectional soft output decoder of turbo codes," *Proceedings of the International Symposium on Turbo Codes and Related Topics*, pp. 288–292, September 1997.

[9] L. R. 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, no. 2, pp. 284–287, March 1974.

[10] L. Staphorst, "Viterbi decoded linear block codes for narrowband and wideband communication over mobile fading channels," Master's dissertation, University of Pretoria, South Africa, 2005.

[11] L. Hanzo, T. H. Liew, and B. L. Yeap, *Turbo-Coding, Turbo Equalisation and Space-Time Coding for transmission over Fading Channels*, John Wiley & Sons Limited, Chichester, UK, first edition, 2002.

[12] R. G. Gallager, "Low-density parity-check codes," Ph.D. thesis, M.I.T., Cambridge, Massachusetts, 1963.

[13] T. J. Richardson and R. L. Urbanke, "The Renaissance of Gallager's Low-Density Parity-Check codes," *IEEE Communications Magazine*, vol. 41, no. 8, pp. 126–131, August 2003.

[14] R. M. Neal, "Sparse matrix methods and probabilistic inference algorithms - Part I: Faster encoding for low density parity check codes using sparse matrix methods," IMA Programs on Codes, Systems and Graphical Models, Tech. Rep., August 1999.

[15] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, Cambridge, UK, first edition, 2005.

[16] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for Turbo-Like codes," *Proceedings of 36th Allerton Conference on Communication, Control, and Computing*, pp. 201–210, September 1998.

[17] L. Staphorst and L. P. Linde, "Evaluating Viterbi decoded Reed-Solomon block codes on a complex spreaded DS/SSMA CDMA system: Part II -

Channel model, evaluation and results," *Proceedings of AFRICON IEEE Conference*, pp. 335–340, 2004.

[18] C. Komninakis, "A fast and accurate Rayleigh fading simulator," *Proceedings of GLOBECOM Global Telecommunications IEEE Conference*, vol. 6, pp 3306–3310, December 2003.