Low-Density Parity-Check Code Design for GSM

Andre M. Mc Donald and Jan C. Olivier
Department of Electrical, Electronic and Computer Engineering, University of Pretoria, South Africa.
Tel: 27-12-420-2060
Fax: 27-12-362-5000
E-mail: andre.mcdonald@up.ac.za

***Abstract*— We present a technique for designing low-density parity-check (LDPC) code matrices for the MCS-1 to MCS-4 coding schemes of GSM. We demonstrate that these codes have significant performance gains over randomly constructed LDPC codes in the GSM setup. The new codes also show significant gains in performance over the conventional MCS-1 to MCS-4 GSM coding schemes when compared in terms of the bit-error rate.**

I. Introduction

With over 2 billion mobile connections and 82% of the global mobile market [1], GSM is arguably the most popular and widespread cellular communication technology available today. The application of modern, capacity-approaching, iterative coding/equalization techniques to the GSM platform is therefore a worthwhile pursuit. While publications do exist on the application of turbo-codes [2] and turbo-equalization [3] to the GSM standard [4]–[6], there is, to the best of the authors' knowledge, no literature on the application of low-density parity-check (LDPC) block codes [7], [8] to the GSM platform. It was demonstrated that properly constructed LDPC codes operate within a fraction of a decibel of the capacity of certain channels [9], but with a decoding complexity that grows only linearly in the block length.

There are two possible reasons for the lack of interest in applying LDPC codes to the GSM platform. Firstly, the performance of LDPC codes improves with an increase in the block-length of the code [10] - it is therefore unlikely that a significant performance gain will be possible in GSM, which has relatively short frames. Secondly, for a fair comparison with the existing GSM system, LDPC-based receivers and transmitters are constrained to use a single generator matrix parity-check matrix pair, and to employ puncturing for achieving higher code rates. The performance of LDPC codes degrades significantly under random puncturing, however [11] - special attention has to be paid when designing these LDPC codes. The design of punctured LDPC codes under the assumption of infinite-length code blocks were considered in [12], [13], while references [11], [14] consider the case of short code blocks (which applies to GSM). The techniques presented in references [11], [14] are useful for designing codes of moderate rate, but are unable to produce codes with very high rates (as is required for Modulation and Coding Schemes MCS-3 and MCS-4 of GSM) [14].

We propose a new technique for designing punctured LDPC codes for the GSM platform. Instead of starting with a low-rate LDPC code and selecting which codeword bits to puncture, as is commonly done when designing punctured codes, we start with a high-rate punctured code, and progressively 'unpuncture' codeword bits to obtain lower-rate codes. The 'unpunctured' codeword bits are intelligently selected as to improve the performance of the code. We demonstrate that these new codes have significant performance gains over randomly constructed LDPC codes in the GSM setup, especially for the lower-rate coding schemes. We further demonstrate that the newly proposed coding schemes improve on the traditional MCS-1 to MCS-4 GSM coding schemes in terms of the bit-error rate.

The remainder of this paper is set out as follows. In section II we present details on the integration of LDPC codes with the GSM standard. Section III presents the new design technique for punctured LDPC codes. In section IV we present simulation results for the original GSM codes, randomly constructed LDPC codes and the newly proposed LDPC codes. Several conclusions are drawn in section V.

II. LDPC codes and the GSM standard

The convolutional code of the GSM standard [15] has a rate of 1/3 and a constraint length of 7. Puncturing is employed to obtain the desired code rates of the four modulation and coding schemes (MCS-1 to MCS-4). The code is applied to both the header and data payload in each Radio Link Control / Media Access Control (RLC/MAC) block to produce coded data[1]. The coded data is punctured and interleaved to produce four radio burst blocks per RLC/MAC block, which are ultimately transmitted.

The length of the uncoded data payload in each RLC/MAC block depends on the coding scheme that is applied to the block - table I lists the four GSM coding schemes and the lengths of the relevant blocks. It is observed that GSM punctures each coded data block in such a manner that a total of 372 coded bits remain, regardless of which coding scheme is employed.

Block codes are inflexible in terms of the block lengths to which they are applied - these codes need to be applied to fixed-length uncoded blocks. The implementation of a LDPC block code in the GSM standard, which requires the use of a single code for all the coding schemes, but with uncoded data payloads of unequal length, is therefore more challenging.

[1]We disregard the coding issue of the header in this paper, and use the original GSM code to encode headers in each case.

TABLE I
GSM CODING SCHEMES MCS-1 TO MCS-4, AND BLOCK LENGTHS

	MCS-1	MCS-2	MCS-3	MCS-4
Uncoded data payload (input to encoder)	196 bits	244 bits	316 bits	372 bits
Coded, unpunctured data block	588 bits	732 bits	948 bits	1116 bits
Coded, punctured data block	372 bits	372 bits	372 bits	372 bits

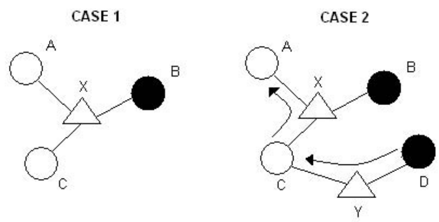

Fig. 1. Two example Tanner graphs. Circles represent variable nodes, and triangles represent check nodes. Erased (or punctured) variable nodes are not filled.

Our approach consists of designing a single rate 1/3 'parent' LDPC code, which is punctured to be rate-compatible with MCS-4. The block lengths that are associated with this code are identical to those of MCS-4, as indicated in table I. The parent code is then extended to the remaining MCS coding schemes.

For MCS-3, each 316-bit uncoded data payload is first padded to a length of 372 bits (the MCS-4 uncoded payload length) with data that is known to both the transmitter and receiver. This 'padded' payload is encoded using a systematic generator matrix of the LDPC parent code. The resulting codeword is subsequently punctured to produce 372 bits. The resulting scheme is rate-compatible with the MCS-3 scheme, and overcomes the input-length issue of the block code.

The padded bits are known a-priori at the receiver - the decoder simply inserts the correct values of these bits into the received data. As the receiver has prior knowledge of the padded data, these bits need not be transmitted. The padded bits are therefore automatically punctured in the systematic part of each codeword prior to transmission.

The extension of the parent code to the MCS-2 and MCS-1 coding schemes is very similar to the extension of the parent code to MCS-3. For MCS-2 we simply append more bits to the uncoded data payload - a total of 128 padded bits is added to produce a padded block of 372 bits. For MCS-1, a total of 176 padded bits are appended to each uncoded data block.

The challenge of designing a parent LDPC block code for this 'padding'-approach to error correction in GSM lies in guaranteeing good performance over both low (MCS-1) and very high (MCS-4) code rates. Several puncturing techniques in the literature cannot produce LDPC codes with very-high code rates, as is required for the design of the parent code [11], [14]. The code must also be designed to use the padded bits in an effective manner - this issue is not considered in other puncturing techniques. These issues are addressed in the next section.

III. DESIGN OF PUNCTURED LDPC CODES

A technique for designing punctured LDPC codes for the GSM standard is presented in this section. We start by discussing several design issues of punctured LDPC codes, and then specify the structure of the parent LDPC code parity-check matrix.

Note that factor graphs [16] will be used to illustrate the design issues - these graphs represent the parity-check matrix of a block code visually.

A. Design issues associated with punctured LDPC codes

The design of punctured codes may be likened to the design of finite-length codes for the binary erasure channel [10], if we assume that the punctured variable nodes are equivalent to erasures[2]. One problem with the design of codes for the erasure channel is that of stopping sets, which are subsets of erased variable nodes that are unrecoverable regardless of the number of iterations of belief propagation [11].

Fig. 1 (case 1) illustrates how a stopping set may occur in a factor graph. If more than one erased variable node (nodes A and C in fig. 1, case 1) are connected to a check node, it is impossible to 'recover' the value of either one of these variable nodes via the specific check node, despite the number of un-erased variable nodes that may also be connected to that check node. This fact can easily be confirmed from the check node to variable node message-passing equation of belief-propagation decoding [7]. If the graph is of such a nature that one of the erased variable nodes cannot be recovered from any of the check nodes to which it is connected, regardless of the number of iterations, this variable node is part of a stopping set.

Stopping sets may be avoided for the graph in fig. 1 by connecting a new check node (fig. 1 case 2, check node Y) to one of the erased variable nodes. By adding an un-erased variable node (node D) to this new check node, we guarantee that the erased variable node (node C) will be recovered in a single iteration of belief propagation, as indicated by the arrows. After the second iteration, the remaining erased variable node (node A) is also recovered. We have found that a randomly constructed factor graph typically has no stopping sets if we ensure that a sufficient number of erased variable nodes are recovered in the first few iterations of belief-propagation decoding. The absence of stopping sets is therefore likely provided that a sufficient number of erased nodes are connected (one at a time) to one or more un-erased variable nodes via check nodes. The problem with applying this design approach to punctured codes is that an unpunctured

[2] A key difference between the two design techniques is that the designer has control over the location of the erasures (punctures) when designing the punctured code, but not when designing the code for the erasure channel.

variable node may be unreliable (i.e. the prior log-likelihood ratio of the variable node is 0, and is thus equivalent to an erasure) and therefore stop the recovery of the punctured variable node connected to the same check node. We therefore introduce the notion of using the padded variable nodes, which are always perfectly reliable, for recovering some of the punctured variable nodes of the graph (recall that the values of padded variable nodes are known a-priori by the decoder). By using the padded nodes, we may guarantee that belief-propagation decoding will progress regardless of the reliability of a few variable nodes that may instead be responsible for recovering the values of several punctured variable nodes.

The alternative to using padded variable nodes for avoiding stopping sets is to simply design the code in such a manner that a sufficient number of check nodes have only one connection to a punctured variable node. For codes with rates close to 1 (with many punctured nodes) this approach has a drawback - for these codes, the systematic variable nodes of each codeword are left unpunctured, as it is unlikely that a few unpunctured non-systematic variable nodes would result in the recovery of any punctured systematic bits. In this way, we can at least recover the systematic part of the codeword if these bits are not corrupted. In an attempt to reduce the number of stopping sets in these high-rate codes, we require that some check nodes contain only one connection to a single, punctured, non-systematic variable node. As we are using the same code for both high rate (many punctured nodes) and low rate (few punctured nodes) transmission in the GSM system, this approach results in a low-rate code with fewer connections between systematic variable nodes and non-systematic variable nodes, which reduces the overall performance of the low-rate code. The approach of using the padded variable nodes for recovery of the punctured nodes is therefore the better alternative.

B. *Structure of the parent LDPC code parity-check matrix*

Having discussed the design issues of punctured LDPC codes and the approach of using padded bits to reveal punctured variable nodes, we implement this approach by designing the parity-check matrix of the parent LDPC code. Note that all systematic codeword bits remain unpunctured (as to avoid problems with the very-high rate MCS-4 coding scheme), with the exception of the padded systematic bits for each of the coding schemes. To maintain rate-compatibility with the four coding schemes, all non-systematic codeword bits are punctured, with the exception a few randomly-selected non-systematic codeword bits. The number of randomly-selected non-systematic bits that are left unpunctured equals the number of padded bits for each of the MCS coding schemes.

The parity-check matrix $\mathbf{H}_{m,n}$ of the parent code may be expressed as[3]

$$\mathbf{H} = \begin{pmatrix} \mathbf{I}_{\Delta,\Delta} & \mathbf{0}_{\Delta,m-\Delta} & \mathbf{I}_{\Delta,\Delta} & \mathbf{0}_{\Delta,k-\Delta} \\ \mathbf{A}_{m-\Delta,\Delta} & \mathbf{B}_{m-\Delta,m-\Delta} & \mathbf{C}_{m-\Delta,\Delta} & \mathbf{D}_{m-\Delta,k-\Delta} \end{pmatrix} \tag{1}$$

[3]Matrices are printed in bold. The subscript indicates the dimensions of each matrix.

where $\mathbf{I}$ is the identity-matrix, $\mathbf{0}$ is the all-zero matrix, and $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ denote matrices with certain structures. Each codeword is n bits long, with k systematic bits and $m = n-k$ non-systematic bits, while Δ is a scalar design parameter. Observe that matrix $\mathbf{H}$ consists of 8 matrices in two rows and four columns - we refer to these rows and columns as m-rows (matrix-rows) and m-columns (matrix-columns) to distinguish them from the ordinary rows and columns of the matrix. Matrix $\mathbf{A}$ of (1) is expressed as

$$\mathbf{A} = \begin{pmatrix} \mathbf{I}_{\Delta,\Delta} \\ \mathbf{I}_{\Delta,\Delta} \\ \mathbf{0}_{m-3\Delta,\Delta} \end{pmatrix} \tag{2}$$

while matrix $\mathbf{B}$ of (1) is expressed as

$$\mathbf{B} = \begin{pmatrix} \mathbf{E}_{m-2\Delta,m-\Delta} \\ \mathbf{I}_{\Delta,\Delta}\ \mathbf{0}_{\Delta,m-2\Delta} \end{pmatrix} \tag{3}$$

Matrix $\mathbf{C}$ of (1) is expressed as

$$\mathbf{C} = \begin{pmatrix} \mathbf{0}_{m-2\Delta,\Delta} \\ \mathbf{I}_{\Delta,\Delta} \end{pmatrix} \tag{4}$$

and matrix $\mathbf{D}$ of (1) is expressed as

$$\mathbf{D} = \begin{pmatrix} \mathbf{F}_{m-2\Delta,k-\Delta} \\ \mathbf{0}_{\Delta,k-\Delta} \end{pmatrix} \tag{5}$$

Matrices $\mathbf{E}$ and $\mathbf{F}$ are randomly-constructed binary matrices with E_r (F_r) average non-zero bits per row, and E_c (F_c) average non-zero bits per column, respectively. The parameters E_r and F_r are selected when designing the code - these parameters imply certain average column weights. Non-zero bits are placed randomly within the $\mathbf{E}$ and $\mathbf{F}$ matrices, and are divided as evenly as possible between the rows and columns, thereby minimizing the variation from the mean row and column weights.

We now proceed by motivating the design structure of the parity-check matrix. The matrix is designed under the assumption that the m non-systematic bits of each codeword correspond to the first and second m-columns of $\mathbf{H}$, while the k systematic bits correspond to the third and fourth m-columns of $\mathbf{H}$. We also assume that maximum overlap between the padded systematic bits and the third m-column of $\mathbf{H}$ occurs. To preserve these properties, special consideration needs to be taken when constructing the systematic generator matrix from the parity-check matrix using Gaussian elimination.

Recall that Gaussian elimination iteratively constructs the $m \times m$ identity-matrix in the left side of $\mathbf{H}$ using row combining and column exchanges, and ultimately produces a generator matrix with the property that the systematic bits are contained in the final k bits of each codeword that it produces. By eliminating the column exchanges during Gaussian elimination, we would be able to preserve the structure of the parity-check matrix so that the systematic and non-systematic bits fall within the proper columns of $\mathbf{H}$. While our technique cannot eliminate all column exchanges during the construction of the generator matrix, we can guarantee that the padded bits of the third m-column of $\mathbf{H}$ are not moved during the first Δ

iterations of Gaussian elimination. This is achieved by placing the identity-matrix in the top-left corner of $\mathbf{H}$ (1), and the all-zero matrix in the first m-row, second m-column of $\mathbf{H}$. We have found that, in general, very few column exchanges occur during the remaining iterations of the algorithm.

The structure of $\mathbf{H}$ is chosen so that Δ systematic, padded variable nodes are connected to Δ non-systematic, punctured variable nodes via Δ check nodes. We therefore place the $\Delta \times \Delta$ identity matrix in the bottom rows of matrices $\mathbf{B}$ (3) and $\mathbf{C}$ (4) to establish these connections. To ensure that the punctured non-systematic nodes are 'revealed' in a single iteration, we avoid connecting any other variable nodes to the Δ check nodes by placing all-zero matrices in matrices $\mathbf{A}$ (2), $\mathbf{B}$ (3) and $\mathbf{D}$ (5).

The identity matrices of $\mathbf{A}$ (2), as well as the $\mathbf{E}$ (3) and $\mathbf{F}$ (5) matrices connect the systematic variable nodes with the non-systematic variable nodes in a random fashion, as is done in the construction of random LDPC code ensembles. The matrices are sparse in order to increase the effectiveness of belief-propagation decoding, and to avoid short cycles in the graph. Observe that MCS-4 has no padded bits, and that we may need to infer the values of the variable nodes in the third m-column of $\mathbf{H}$ (1) using the non-systematic variable nodes. We establish these connections by placing the identity matrix in the top-left corner of $\mathbf{H}$ and two identity matrices in $\mathbf{A}$ (2). These identity matrices connect the third m-column of $\mathbf{H}$ to the variable nodes of the second m-column of $\mathbf{H}$. These connections are intended to increase the likelihood of recovering the variable nodes of the third m-row of $\mathbf{H}$ should these nodes be corrupted by noise.

After having constructed the parity-check matrix for chosen values of Δ, E_r and F_r, we obtain the systematic generator matrix using Gaussian elimination. We then proceed by selecting the puncturing and padding patterns for each of the coding schemes of GSM. The puncturing patterns were specified previously - for the padding patterns, we simply select the appropriate number of bits from the systematic bits that correspond to the third m-column of $\mathbf{H}$ (1). Any remaining padded bits that need to be selected (as occurs when Δ is less than the number of padded bits) are randomly chosen from the remaining systematic bits.

We conclude this section by elaborating on the encoding and decoding processes. For MCS-4, we simply code the systematic data block (372 bits long) using the generator matrix. At the decoder, no padded bit values are substituted into the received codeword, and conventional belief-propagation decoding is carried out. For MCS-1 to MCS-3, the coding/decoding process is somewhat different. At the encoder, we pad each systematic data block prior to coding with known data to obtain a block of 372 uncoded bits. Coding is performed by multiplying this 'padded' block with the generator matrix, and subsequently puncturing the codeword. At the receiver, we modify the soft-input to the decoder by providing perfect-reliability information to the systematic padded bits of each codeword (these bits are effectively 'unpunctured'). The belief-propagation algorithm is then used to recover the codeword.

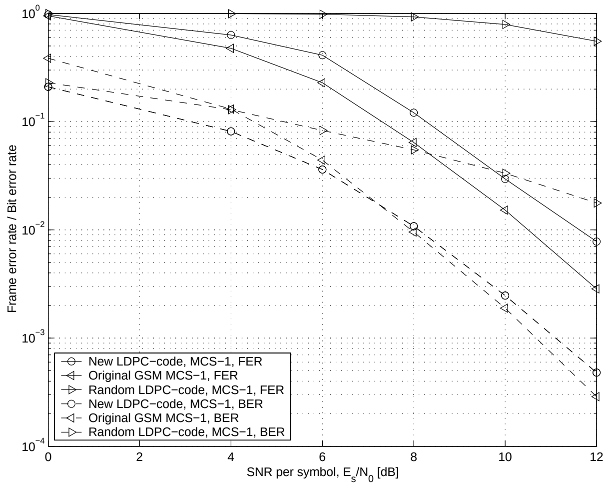

Fig. 2. Simulation results for the MCS-1 coding schemes. Solid lines represent frame-error rates, and dashed lines represent bit-error rates.

IV. Results

This section contains bit-error rate (BER) and frame-error rate (FER) simulation results for the various coding schemes. All simulations were carried out on a software implementation of the GSM physical layer [15], with coding modules replaced as needed for the new coding schemes.

The coding schemes considered here are the original GSM MCS-1 to MCS-4 coding schemes, the newly-proposed LDPC-based MCS-1 to MCS-4 coding schemes, and randomly constructed LDPC coding schemes at rates compatible to those of MCS-1 to MCS-4. The original GSM coding schemes were implemented with puncturing mode 1 [15], as well as a standard Viterbi channel decoder. The new LDPC-based coding schemes were constructed using design parameters of $\Delta = 10$, $E_r = 2.03$ and $F_r = 2$. The randomly constructed LDPC MCS-4 coding scheme uses the same block lengths as the original GSM MCS-4 coding scheme - the parent code was constructed randomly with 3 non-zero bits per parity-check matrix column, and with non-zero bits evenly distributed between rows. Padded bits were added to the uncoded bits to obtain the MCS-1 to MCS-3 rate-compatible random LDPC coding schemes, while all non-systematic codeword bits were punctured to maintain rate compatibility with the coding schemes. All LDPC-based schemes used the iterative belief-propagation decoder [7].

Each simulation used the standard 12-tap typically-urban multipath fading channel profile as specified in the GSM standard [15], with a mobile speed of 50 km/h at the 900 MHz carrier frequency (i.e. a maximum Doppler spread of 41.6 Hz). A 2-tap Max-Log-MAP equalizer was used to reduce the distortion caused by multipath fading. An additive white Gaussian noise source was considered, with no co-channel interference. Interleaving, as specified in the GSM specification, was applied to all data frames.

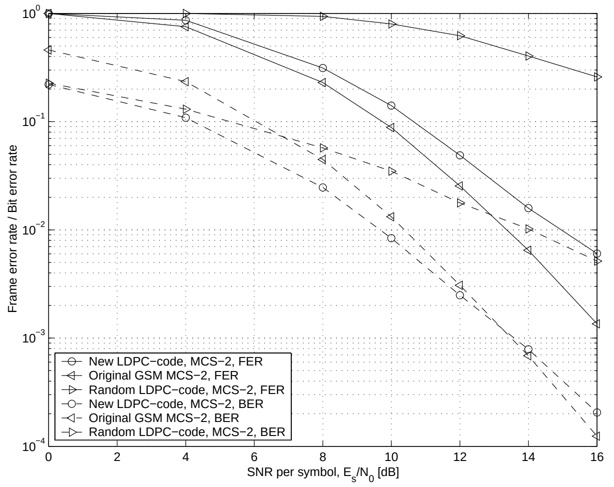

Fig. 3. Simulation results for the MCS-2 coding schemes. Solid lines represent frame-error rates, and dashed lines represent bit-error rates.

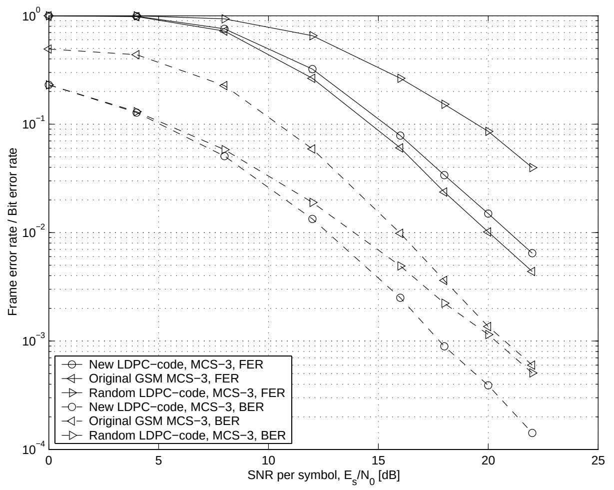

Fig. 4. Simulation results for the MCS-3 coding schemes. Solid lines represent frame-error rates, and dashed lines represent bit-error rates.

The bit-error/frame-error rate graphs for the coding schemes based on the MCS-1 to MCS-4 code rates are presented, respectively, in figs. 2 to 5. Several observations are made regarding the frame-error rates of the coding schemes. Firstly, the random MCS-1 LDPC coding scheme is far inferior to both the new MCS-1 LDPC coding scheme and the original GSM MCS-1 coding scheme (with an observed maximum coding loss of around 7.8 dB to the new LDPC code). This observation confirms that the design of the new LDPC-based coding scheme is sound. For the coding schemes based on the rates of MCS-2 to MCS-4, the FER performance of the random LDPC coding schemes approaches that of the new LDPC coding schemes and the original GSM coding schemes as the code rate increases - for MCS-4, the FER performances of the three schemes are virtually identical. The new LDPC coding schemes therefore seem to perform better at lower coding rates, when compared to the random LDPC codes.

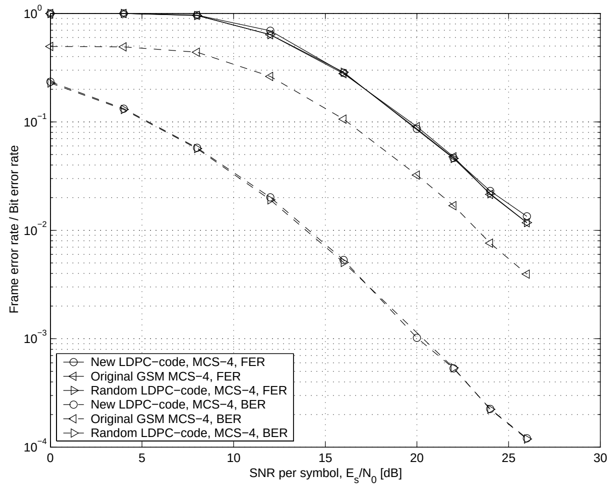

Fig. 5. Simulation results for the MCS-4 coding schemes. Solid lines represent frame-error rates, and dashed lines represent bit-error rates.

Secondly, the frame-error rate of the new MCS-1 coding scheme is larger than that of the original GSM MCS-1 coding scheme. This implies that the new-technique is not effective when transmitting data that is required to be lossless (i.e. data for which the frame-error rate is more important than the bit-error rate, such as text). This observation also holds for the MCS-2 and MCS-3 coding schemes, but with a penalty that decreases with increasing rate. At the MCS-4 code rate, there is no observable difference between the frame error rates of the new LDPC code and the original GSM code. Table II summarizes the various gains and crossing points of the coding schemes.

GSM is not a data-services standard - the higher-throughput EDGE standard was developed for this purpose. GSM is primarily intended for delivering voice services, in which some loss of data is acceptable. With voice data, the frame error rate is less important, as we cannot re-transmit corrupted frames (this will cause random delays in the speech). We therefore shift the discussion from the frame-error rate to the bit-error rate, which is more important for voice quality in GSM.

The bit-error rate curves of figs. 2 to 5 suggest that even the random LDPC coding scheme outperforms the GSM coding scheme at lower SNRs per symbol. The BER performance of the newly proposed LDPC coding schemes is significantly better than that of the random LDPC coding schemes, especially at lower rates. The new LDPC scheme also outperforms the original GSM coding schemes, especially at low SNRs - the SNR range over which this observation holds becomes larger with an increase in the code rate (i.e. the BER crossing points of the original GSM codes and the new LDPC codes occurs at a much higher relative SNR when switching from MCS-1 to MCS-2). For the MCS-3 and MCS-4 coding schemes, we observe that the new LDPC coding schemes outperform the original GSM coding schemes over the entire range of SNR. The coding gain (in terms of the BER) of the new LDPC code is approximately 9 dB over that of the original GSM code for

TABLE II
GAINS AND CROSSING POINTS ASSOCIATED WITH THE ERROR RATES OF THE CODING SCHEMES

	MCS-1	MCS-2	MCS-3	MCS-4
New LDPC coding gain over random LDPC coding (FER)	7.83 dB at 0.553 FER	7.525 dB at 0.259 FER	4.38 dB 3.97×10^{-2} FER	0 dB
New LDPC coding loss over original GSM coding (FER)	1.21 dB at 7.80×10^{-3} FER	1.91 dB 6.04×10^{-3} FER	0.93 dB 6.45×10^{-3} FER	0 dB
Crossing point between new LDPC code and original GSM coding scheme (BER)	7.24 dB at 1.71×10^{-2} BER	13.21 dB at 1.24×10^{-3} BER	None	None
Coding gain of the new-LDPC code over the original GSM codes in terms of BER (for voice data)	0.9 dB at 8.1×10^{-2} BER	1.7 dB at at 8.1×10^{-2} BER	> 2.64 dB for entire SNR range considered	> 9 dB for entire SNR range considered

MCS-4, and 2.6 dB for MCS-3.

We may therefore state that the new LDPC coding scheme results in better quality voice at higher code rates, and for low to medium SNRs at lower code rates.

V. Conclusion

We conclude that the new LDPC coding scheme results in better quality voice (lower bit-error rate) than the voice carried over mobiles that use the original GSM coding schemes. This gain in performance is significant, as voice is the key data type to be carried by GSM. The gain occurs at low to moderate SNRs for the MCS-1 and MCS-2 coding schemes, and over the entire range of SNRs for the MCS-3 and MCS-4 coding schemes - a huge 9 dB gain was observed over the original MCS-4 GSM coding scheme.

The reported gain in SNR may potentially be exploited in the design of mobiles - the mobile would require significantly less energy to transmit voice at the same quality as with the original GSM coding schemes. This results in cheaper mobiles (we could use cheaper amplifiers than before and still have acceptable voice quality), with longer battery life.

References

[1] (2007) The gsmworld website. [Online]. Available: http://www.gsmworld.com/

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting and decoding: Turbo-codes (l)," in *Proc. IEEE International Conference on Communications (ICC'93)*, May 1993, pp. 1064–1070.

[3] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo-equalization," *European transactions on Telecommunications*, vol. 6, pp. 507–511, 1995.

[4] F. Burkert, G. Caire, J. Hagenauer, T. Hindelang, and G. Lechner, "Turbo decoding with unequal error protection applied to GSM speech coding," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM'96)*, Nov. 1996, pp. 2044–2048.

[5] G. Bauch and V. Franz, "Iterative equalization and decoding for the GSM-system," in *Proc. IEEE Vehicular Technology Conference (VTC'98)*, May 1998, pp. 2262–2266.

[6] J. Bajcsy, C.-V. Chong, D. Garr, J. Hunziker, and H. Kobayashi, "On iterative decoding in some existing systems," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 883–890, May 2001.

[7] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399–431, Mar. 1999.

[8] R. G. Gallager, "Low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 8, pp. 21–28, Jan. 1962.

[9] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, Feb. 2001.

[10] T. Richardson and R. Urbanke. (2007, Apr.) Modern coding theory. [Online]. Available: http://lthcwww.epfl.ch/mct/index.php

[11] J. Ha, J. Kim, D. Klinc, and S. W. McLaughlin, "Rate-compatible punctured low-density parity-check codes with short block lengths," *IEEE Trans. Inform. Theory*, vol. 52, pp. 728–738, Feb. 2006.

[12] J. Ha, J. Kim, and S. W. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. IT-50, pp. 2824–2836, Nov. 2004.

[13] J. Ha and S. W. McLaughlin, "Optimal puncturing distributions for rate-compatible low-density parity-check codes," in *Proc. Int. Symp. Information Theory*, June 2003, p. 233.

[14] N. Vellambi, R. Badri, and F. Fekri, "Rate-compatible puncturing of finite-length low-density parity-check codes," in *Proc. IEEE International Symposium on Information Theory (ISIT'2006)*, July 2006, pp. 1129–1133.

[15] *Digital cellular communication systems (Phase 2+) Transmission and reception*, 3GPP TS 05.05 Std. GSM 05.05 version 8.5.0, 1999.

[16] R. Koetter, "Factor graphs and iterative algorithms," in *Proc. IEEE Information Theory and Networking Workshop*, June 1999, p. 28.

Andre M. Mc Donald Andre Mc Donald received the B.Eng (Computer) and B.Eng (Hons, Electronics) summa cum laude from the University of Pretoria in 2004 and 2005. He is currently pursuing his Master's degree in Electronic Engineering at the University of Pretoria.

Jan C. Olivier Jan Olivier received the B.Eng (Electronics), M.Eng (Electronics) and Ph.D. (Electronics) summa cum laude from the University of Pretoria in 1985, 1986 and 1990. He was appointed as a Professor at the University of Pretoria (2003), where he is Director of the Telkom Centre of Excellence in Telecommunications Engineering (CETEIS).