Performance of LTE Turbo Codes with Joint Source Channel Decoding, Adaptive Scaling and Prioritised QAM Constellation Mapping

Tulsi Pawan Fowdur, Yogesh Beeharry, and K.M. Sunjiv Soyjaudah
Dept of Electrical and Electronic Engineering
University of Mauritius
Réduit, Mauritius
e-mail: p.fowdur@uom.ac.mu, yogesh536@hotmail.com, ssoyjaudah@uom.ac.mu

Abstract— **Turbo coded Quadrature Amplitude Modulated (QAM) systems have been adopted by standards such as Code Division Multiple Access (CDMA) 2000 and Long Term Evolution (LTE) to achieve high data rates. Although several techniques have been developed to improve the performance of Turbo coded QAM systems, combinations of these techniques to produce hybrids with better performances, have not been fully exploited. This paper investigates the performance of LTE Turbo codes with Joint Source Channel Decoding (JSCD), adaptive Sign Division Ratio (SDR) based scaling and prioritised QAM constellation mapping. JSCD exploits a-priori source statistics at the decoder side and SDR based scaling provides a scale factor for the extrinsic information as well as a stopping criterion. Additionally, prioritised constellation mapping exploits the inherent Unequal Error Protection (UEP) characteristic of the QAM constellation and provides greater protection to the systematic bits of the Turbo encoder. Simulation results show that with 16-QAM at Bit Error Rates (BERs) above 10^{-1}, the combination of these three techniques achieves an average gain of 1.7 dB over a conventional LTE Turbo coded 16-QAM system. For BERs below 10^{-1}, the combination of prioritized constellation mapping, JSCD and SDR scaling provides an average gain of 0.6 dB. The proposed scheme with 64-QAM gives a performance gain of 3 dB on average over the conventional LTE Turbo coded 64-QAM system over the whole E_b/N_0 range.**

Keywords- Turbo Code; QAM; JSCD; SDR; Priortised Mapping.

I. INTRODUCTION

This paper extends the work of [1] by considering both 16-QAM and 64-QAM with LTE Turbo codes. Since the inspection of Turbo codes by Berrou *et.al* in 1993 [2], several communication standards have adopted this powerful near Shannon limit error correcting code. For example, Turbo coded QAM systems have been widely exploited to achieve reliable transmission at high data rates in several standards such as Long Term Evolution (LTE) [3] [4], CDMA 2000 [5] and HomePlug Green PHY [6]. These systems have also been reported to be promising for IEEE 802.11a [7]. The major impact of Turbo codes has led to the emergence of several techniques such as Joint Source Channel Decoding (JSCD) [8] [9] [10] [11], extrinsic information scaling and iterative detection [12] [13] [14] [15] to improve its error performance and lower its decoding complexity. Moreover, certain characteristics of the 64-QAM constellation have also been exploited to improve the performance of Turbo coded QAM [16]. An overview of these techniques is given next.

JSCD essentially involves the use of a-priori source statistics and the exploitation of residual redundancy to enhance the channel decoding process. For example, Murad and Fuja [8] proposed a composite trellis, made up of a Markov source, a Variable Length Code (VLC), and a channel decoder's state transitions, to exploit a priori source statistics. A low complexity version of the technique in [8] was developed by Jeanne *et.al* [9] and more recently Xiang and Lu [10] proposed a JSCD scheme for Huffman encoded multiple sources, which could exploit the a-priori bit probabilities in multiple sources. Also, Fowdur and Soyjaudah [11] proposed a JSCD scheme with iterative bit combining, which incorporated two types of a-priori information, leading to significant performance gains. On the other hand, extrinsic information scaling aims at improving the Turbo decoder's performance by scaling its extrinsic information with a scale factor. For example, Vogt and Finger [12] used a fixed scale factor to improve the Max-Log-MAP Turbo decoding algorithm, while Gnanasekaran and Duraiswamy [13] proposed a modified MAP algorithm using a fixed scale factor. Interestingly though Lin *et.al* [14] proposed a scaling scheme that extended the Sign Division Ratio (SDR) technique of Wu *et.al* [15] to adaptively determine a scaling factor for each data block at every iteration. Finally, the Turbo decoding process can be further enhanced by exploiting the UEP characteristic of the QAM constellation to give more protection to the systematic bits of the Turbo encoder. This technique has been applied to LTE Turbo codes by Lüders *et.al* [16].

In contrast with previous works, which have mostly considered the schemes developed to improve the performance of Turbo codes independently, this paper builds upon the work of [1] and analyses the performance of LTE Turbo codes with QAM by integrating three different techniques. Firstly, at the encoder side, prioritized constellation mapping [16] is performed so that the systematic bits output by the Turbo encoder are given the highest protection when they are mapped onto the QAM constellation. The second technique employed is JSCD [8] [11], which exploits a-priori source statistics during Turbo decoding. The final technique used is adaptive extrinsic scaling based on the SDR criterion [14]. Significant performance gains are obtained for both LTE Turbo coded

16-QAM and 64-QAM systems with the combination of these three techniques.

The organization of this paper is as follows. Section II describes the complete system model. Section III presents the simulation results and analysis. Section IV concludes the paper and lists some possible future works.

II. SYSTEM MODEL

The complete transmission system is shown in Fig. 1. A random alphabet source is first generated with a non-uniform probability distribution and then encoded into bits with the Reversible Variable Length Code (RVLC) of [17].

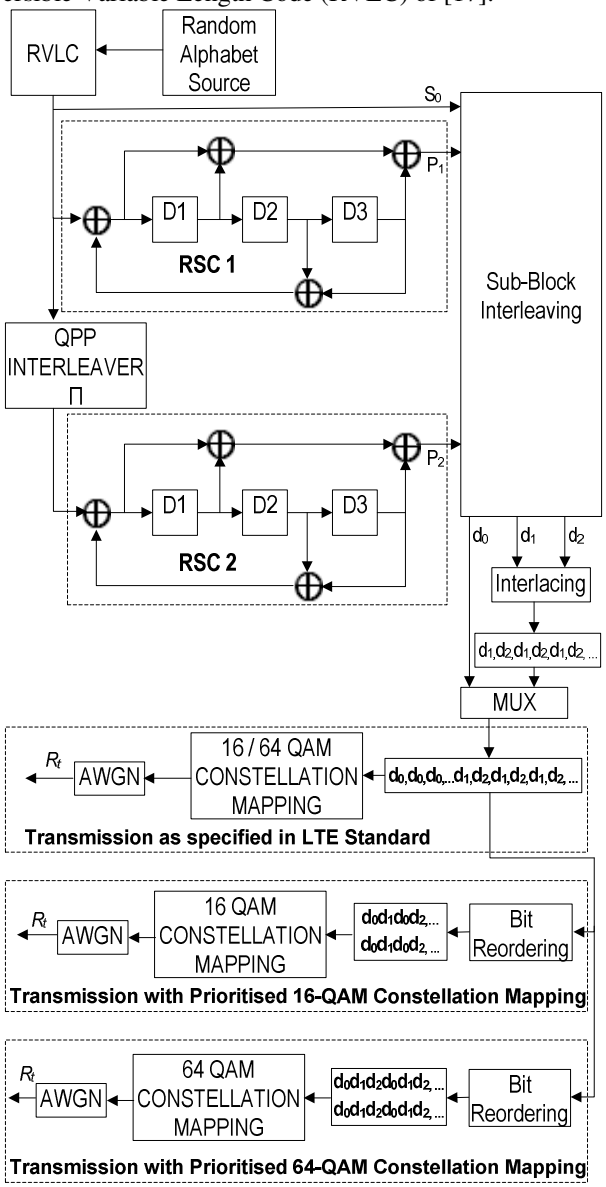

Fig. 1. Complete Transmission system

The coded bits are fed to an LTE Turbo encoder, which consists of a parallel concatenation of two Recursive Systematic Convolutional (RSC) encoders, RSC1 and RSC2, separated by an interleaver (Π). Fig. 2 shows the trellis diagram of the RSC encoders.

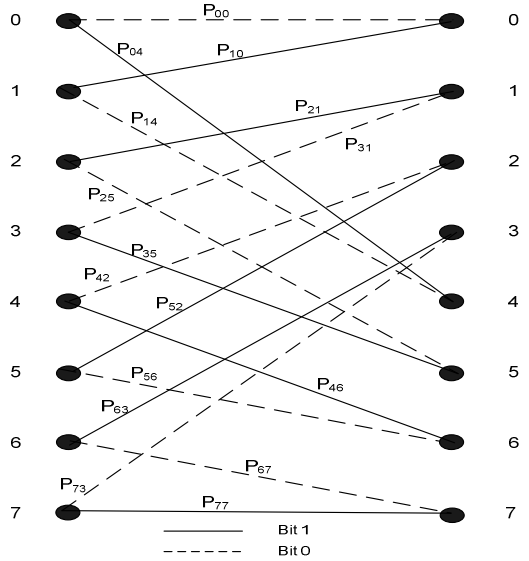

Fig. 2. Trellis diagram for LTE Turbo decoder

The interleaver used in the design of the LTE Turbo encoding system is a Quadratic Polynomial Permutation (QPP) interleaver. As the name suggests, QPP interleaver is based on a quadratic polynomial to obtain the interleaved permutation. The relationship between the output index z and input index $\pi(z)$ is expressed by the following equation [18]:

$$\pi(z) = (f_1.z + f_2.z^2)\ mod\ K \qquad (1)$$

where
f_1 and f_2 are the permutation coefficients,
K is the size of the information block.

After the encoding process, each of the three output streams S_0, P_1, and P_2 is interleaved with its own sub-block interleaver resulting in d_0, d_1, and d_2 respectively. Sub-block interleaving is performed as follows:

a. The input block is written row-wise in a rectangular matrix to form a R_s by C_s matrix.
b. The columns of this matrix are permuted.
c. The output is read column-wise from the permuted matrix.

The number of columns, C_s, is fixed to 32 and the number of rows, R_s, is (K / 32). The inter-column permutation is achieved using the Bit-Reversal-Order (BRO) pattern given below:
Q(j) = [0, 16, 8, 24, 4, 20, 12, 28, 2, 18, 10, 26, 6, 22, 14, 30, 1, 17, 9, 25, 5, 21, 13, 29, 3, 19, 11, 27, 7, 23, 15, 31]

Q(j) is the original column position of the j^{th} permuted column.

The multiplexing in the LTE Turbo coded standard is done such that the rearranged systematic bits, d_0, are placed

in the beginning followed by bit-by-bit interlacing of the two rearranged parity streams, d_1 and d_2, in order to form a single output buffer [18]. The LTE Turbo encoder generates a systematic stream, d_0 and two parity streams d_1 and d_2. To achieve prioritized constellation mapping, such that the systematic bits, d_0, are placed at the most strongly protected points on the QAM constellation, bit reordering [16] must be performed after the multiplexing process. The bit reordering is performed either on a group of four bits at a time since four bits are mapped onto one complex 16-QAM symbol or on a group of six bits at a time since six bits are mapped onto one complex 64-QAM symbol. The modulated symbols are then sent over an AWGN channel and the receiver obtains the corrupted symbols, R_t.

With 16-QAM, it is observed that after bit re-ordering, the systematic bits d_0 occupy the first and third positions of the four bits that are mapped onto one symbol of the 16-QAM constellation shown in Fig. 3. In this constellation, the bits found in the first and third positions are most protected, while the bits found in the second and fourth positions receive the lowest protection.

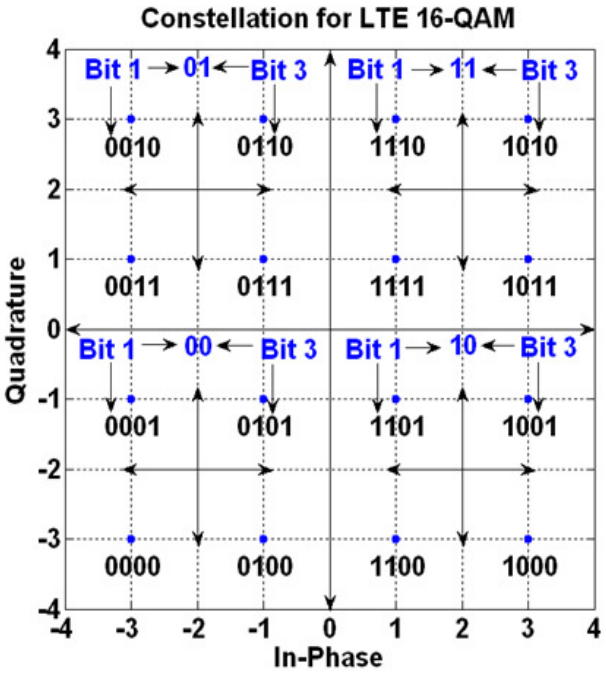

Fig. 3. 16-QAM constellation with major quadrants.

This can be explained by considering the four major quadrants of the 16-QAM constellation. The major quadrants are distinguished by the bits in the first and third positions in the 16-QAM constellation point. For example, in the upper right major quadrant of the 16-QAM constellation, the first and third bits are 11. Hence, if the de-mapper only distinguishes between the four quadrants correctly, the first and third bits of the 16-QAM constellation are correctly de-mapped.

With 64-QAM, after bit re-ordering, the systematic bits d_0 occupy the first and fourth positions of the six bits that are mapped onto one symbol of the 64-QAM constellation shown in Fig. 4. In this constellation, the bits found in the first and fourth positions are most protected, while the bits found in the third and sixth positions receive the lowest protection.

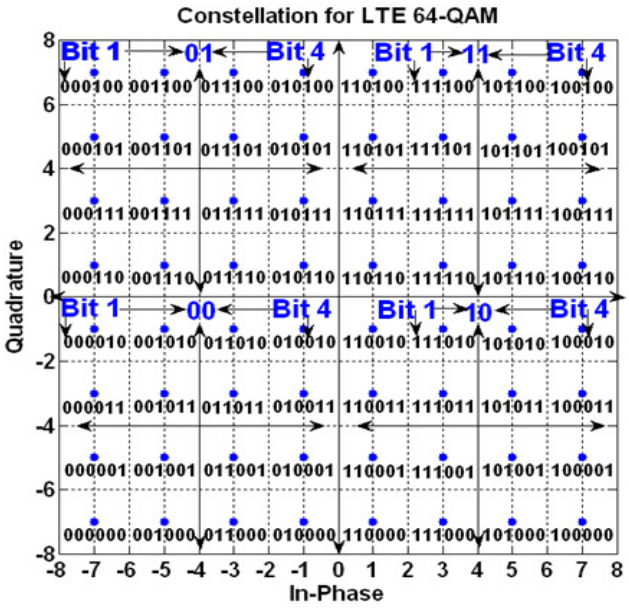

Fig. 4. 64-QAM constellation with major and minor quadrants.

This can be explained by considering both the four major and 16 minor quadrants of the 64-QAM constellation. The major quadrants are distinguished by the bits in the first and fourth positions in the 64-QAM constellation point. The first and fourth bits are 11 in the upper right major quadrant of the 64-QAM constellation. Hence, if the de-mapper only distinguishes between the four quadrants correctly, the first and fourth bits of the 64-QAM constellation are correctly de-mapped. Each major quadrant of the 64-QAM constellation is divided into four minor quadrants, which are distinguished using the 2nd and 5th bits of the constellation points. Therefore, with bit ordering [16], the systematic bits S_0 receive the highest protection while the second parity bits, P_2, receive the lowest. Since the systematic bits of a Turbo encoder have the greatest impact on its performance, the re-ordering scheme improves the performance of the Turbo decoder. The modulated QAM symbols are then transmitted over a complex Additive White Gaussian Noise (AWGN) channel and the corresponding received sequence is denoted by R_t.

The complete system model for the receiver is shown in Fig. 5. The received symbols R_t are fed to a soft-output QAM de-mapper to produce soft bits. These soft bits are then ordered and de-multiplexed. The parity soft bits are de-interlaced. The systematic and parity soft information are sub-block de-interleaved and the soft bits are sent for Turbo decoding. The first Turbo decoder is modified so that it can

incorporate a-priori source statistics by combining the trellis of the Turbo decoder with the trellis of the RVLC decoder as described in [8] and [11]. This results into a composite trellis structure with, which JSCD can be performed. The joint trellis is obtained by merging the trellises of Fig. 2 and Fig. 7. The RVLC codewords are as shown in Table I. The corresponding RVLC decoder state diagram is given in Fig. 6 from, which the bit-level trellis shown in Fig. 7 is obtained. With JSCD the computation of the branch transition probability is modified.

Fig. 5. Turbo decoding system with JSCD and SDR scaling.

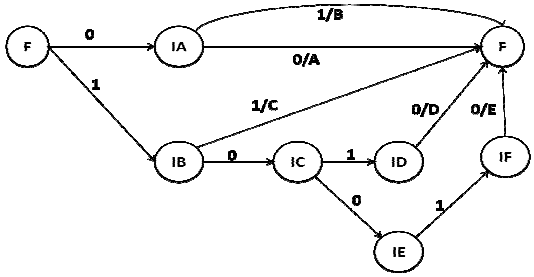

Fig. 6. RVLC decoder state diagram

Assuming that the Max-Log-MAP algorithm [19] is used, the branch metric probability for the joint decoder is computed as follows:

$$\overline{\gamma_t^{1(i)}}(l',l) = \log\left[p_t^1(i).\exp\left(-\frac{[r0_t - x0_t]^2 + [r1_t - x1_t]^2}{2\sigma^2} \right).\Pr\{C_i\} \right]$$

$$= \log\left[p_t^1(i)\right] - \left(\frac{[r0_t - x0_t]^2 + [r1_t - x1_t]^2}{2\sigma^2} \right) + \log[\Pr\{C_i\}] \quad (2)$$

where

$\overline{\gamma_t^{1(i)}}(l',l)$ is the branch transition probability from state l' to l of bit i (i = 0 or 1) at time instant t,

$p_t^1(i)$ is the a-priori probability of bit i derived from the channel extrinsic information and input to the joint (first) decoder,

$\Pr\{C_i\}$ is the a-priori probability of bit i obtained from source statistics,

$r0_t$ and $r1_t$ are the de-mapped soft bits corresponding to the bipolar equivalent of the transmitted systematic bits, $x0_t$ and first parity bits, $x1_t$. σ^2 is the noise variance [11] [19].

With the joint decoder, the a-priori statistics, $\Pr\{C_i\}$ can be incorporated into the Turbo decoding process. The derivation of the a-priori source statistics for the RVLC source given in Table I is now explained. The RVLC decoder's bit-level trellis is shown in Fig. 7 [11] [19].

TABLE I. RVLC CODEWORDS

Symbol	Probability	RVLC [17]
A	0.33 (PA)	00
B	0.30 (PB)	01
C	0.18 (PC)	11
D	0.10 (PD)	1010
E	0.09 (PE)	10010

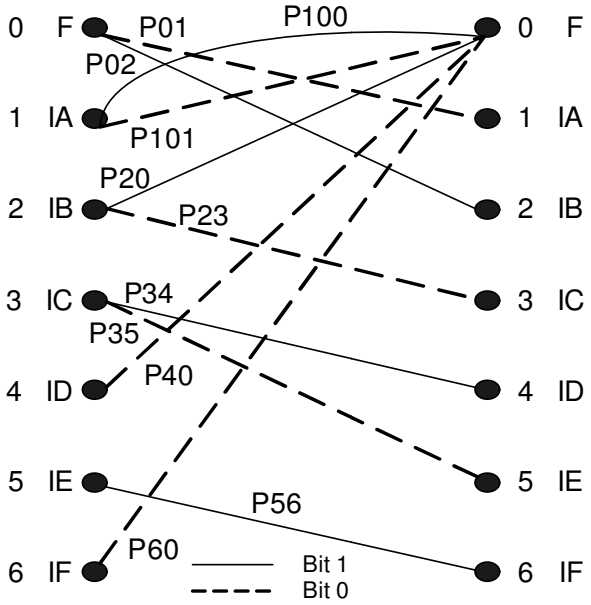

Fig. 7. Bit level trellis of RVLC decoder [11].

From the bit level trellis, the probability of the transition from state $M_{t-1} = l'$ to $M_t = l$, where l', $l \in (F,IA,IB,IC,ID,IE,IF)$, given an input bit i at time instant t, can be derived for all possible state transitions. For example, the probability of the transition from the final state F to the intermediate state IA, is given by [11]:

$$\Pr\left(M_t = IA, i = 0 \mid M_{t-1} = F\right) = PA + PB = P01 \quad (3)$$

For simplicity, the state transition probability for any state corresponding to bit *i* is denoted as $\Pr\{C_i\}$ and the joint decoder exploits this probability by computing the branch metric probability given by (2) [11]. The forward recursive variable, $\overline{\alpha_t^1}(l)$, at time *t* and state *l* is computed as follows for a joint decoder with M_j states:

$$\overline{\alpha_t^1}(l) = \max\left(\overline{\alpha}_{t-1}^1(l') + \overline{\gamma_t^{1(i)}}(l', l)\right) for \;\; 0 \le l' \le M_j - 1 \quad (4)$$

The number of states of the joint decoder, M_j is greater than the number of states, M_s of the second decoder (DEC2), because the joint decoder is obtained by merging the states of the RVLC decoder with the states of the Turbo decoder as described in [11]. A joint trellis is obtained by merging the trellis of the turbo decoder and the bit-level trellis of the RVLC decoder. The backward recursive variable, $\overline{\beta_t^1}(l)$, is computed as follows:

$$\overline{\beta_t^1}(l) = \max\left(\overline{\beta}_{t-1}^1(l') + \overline{\gamma_t^{1(i)}}(l, l')\right) for \;\; 0 \le l' \le M_j - 1 \quad (5)$$

The Log-Likelihood Ratio (LLR), $\Lambda_1^{(r)}(t)$ at iteration *r* and time *t* for the joint decoder is computed as follows:

$$\Lambda_1^{(r)}(t) = \max\left(\overline{\alpha}_{t-1}^1(l') + \overline{\gamma_t^{1(1)}}(l', l) + \overline{\beta}_t^1(l)\right) \quad (6)$$
$$- \max\left(\overline{\alpha}_{t-1}^1(l') + \overline{\gamma_t^{1(0)}}(l', l) + \overline{\beta}_t^1(l)\right) for \;\; 0 \le l' \le M_j - 1$$

The extrinsic information $\Lambda_{1e}^{(r)}(t)$ at iteration *r* and time *t* for the joint decoder is computed as follows:

$$\Lambda_{1e}^{(r)}(t) = \Lambda_1^{(r)}(t) - \frac{2}{\sigma^2} r0_t - \overline{\Lambda}_{2e}^{(r-1)}(t) \quad (7)$$

where $\overline{\Lambda}_{2e}^{(r-1)}(t)$ is the de-interleaved extrinsic information obtained from the second decoder at iteration *r*-1.

The extrinsic information, $\Lambda_{1e}^{(r)}(t)$ and the LLR, $\Lambda_1^{(r)}(t)$ are then sent to a SDR scaling mechanism, which computes a scale factor S_{1r} as follows:

$$S_{1r} = \frac{1}{N} \sum_{t=1}^{N} f\left(\Lambda_{1e}^{(r)}(t), \Lambda_1^{(r)}(t)\right) \quad (8)$$

where $f\left(\Lambda_{1e}^{(r)}(t), \Lambda_1^{(r)}(t)\right)$=1 if $\Lambda_{1e}^{(r)}(t)$ and $\Lambda_1^{(r)}(t)$ have the same sign, otherwise $f\left(\Lambda_{1e}^{(r)}(t), \Lambda_1^{(r)}(t)\right)$=0. *N* is the frame size in bits.

When S_{1r} takes its maximum value of 1.0, the switch T1 is opened, the iterative decoding process is stopped and a hard decision is made on $\Lambda_1^{(r)}(t)$. However, when $S1r$ is less than one, T1 remains closed and the extrinsic information $\Lambda_{1e}^{(r)}(t)$ is scaled with S_{1r} and interleaved to obtain $\overline{\Lambda}_{1e}^{(r)}(t)$. Hence, the SDR scaling mechanism acts both as a stopping criterion and a scale factor generator. The mechanism is derived from the one proposed in [14], but, in this work $\Lambda_{1e}^{(r)}(t)$ and $\Lambda_1^{(r)}(t)$ are used to compute the scale factor and not $\overline{\Lambda}_{2e}^{(r-1)}(t)$ and $\Lambda_{1e}^{(r)}(t)$. Another difference is that in this work only the extrinsic information has been scaled and not the soft channel inputs, as was the case in [14]. The a-priori probability, $p_t^2(i)$, is computed as follows and sent to decoder 2:

$$p_t^2(i) = \left\{ \begin{array}{l} \dfrac{\exp(\overline{\Lambda}_{1e}^{(r)}(t))}{1 + \exp(\overline{\Lambda}_{1e}^{(r)}(t))} \; for \;\; i = 1 \\ \dfrac{1}{1 + \exp(\overline{\Lambda}_{1e}^{(r)}(t))} \; for \;\; i = 0 \end{array} \right\} \quad (9)$$

The branch metric probability for the second decoder is computed as follows:

$$\overline{\gamma_t^{2(i)}}(l', l \;\;) = \log\left[p_t^2(i)\right] - \left(\frac{[\overline{r0}_t - x0_t]^2 + [r2_t - x2_t]^2}{2\sigma^2} \right) \quad (10)$$

where $r2_t$ is the de-mapped soft bits corresponding to the bipolar version of the transmitted second parity bits $x2_t$ and $\overline{r0}_t$ is the interleaved counterpart of $r0_t$.

The forward and backward recursive variable, $\overline{\alpha_t^2}(l)$ and $\overline{\beta_t^2}(l)$ at time *t* and state *l* are computed as follows:

$$\overline{\alpha_t^2}(l) = \max\left(\overline{\alpha}_{t-1}^2(l') + \overline{\gamma_t^{2(i)}}(l', l)\right) for \;\; 0 \le l' \le M_s - 1 \quad (11)$$

$$\overline{\beta_t^2}(l) = \max\left(\overline{\beta}_{t-1}^2(l') + \overline{\gamma_t^{2(i)}}(l, l')\right) for \;\; 0 \le l' \le M_s - 1 \quad (12)$$

The LLR, $\Lambda_2^{(r)}(t)$ and extrinsic information, $\Lambda_{2e}^{(r)}(t)$ at iteration *r* and time *t* are computed as follows:

$$\Lambda_2^{(r)}(t) = \max\left(\overline{\alpha}_{t-1}^2(l') + \overline{\gamma_t^{2(1)}}(l', l) + \overline{\beta}_t^2(l)\right) - \max\left(\overline{\alpha}_{t-1}^2(l') + \overline{\gamma_t^{2(0)}}(l', l) + \overline{\beta}_t^2(l)\right) for \;\; 0 \le l' \le M_j - 1 \quad (13)$$

$$\Lambda_{2e}^{(r)}(t) = \Lambda_2^{(r)}(t) - \frac{2}{\sigma^2}\overline{r0}_t - \overline{\Lambda}_{1e}^{(r)}(t) \quad (14)$$

The scale factor S_{2r} is computed as follows:

$$S_{2r} = \frac{1}{N}\sum_{t=1}^{N} f\left(\Lambda_{2e}^{(r)}(t), \Lambda_2^{(r)}(t)\right) \quad (15)$$

where $f\left(\Lambda_{2e}^{(r)}(t), \Lambda_2^{(r)}(t)\right)$=1 if $\Lambda_{2e}^{(r)}(t)$ and $\Lambda_2^{(r)}(t)$ have the same sign. Finally, the a-priori probability, $p_t^1(i)$, is computed as given by (9) but using $\overline{\Lambda}_{2e}^{(r)}(t)$. If S_{2r} = 1.0, T2 is opened to stop the iterative decoding process and a hard decision, (HD) is made on $\overline{\Lambda}_2^{(r)}(t)$.

The combination of prioritized constellation mapping, JSCD and adaptive scaling certainly lead to an enhanced LTE Turbo coded QAM system, but at the cost of greater computational complexity and delay. The complexity increase due to the bit re-ordering scheme is negligible and may even be integrated with the multiplexer. JSCD on the other hand leads to the greatest increase in complexity and delay because as mentioned previously the joint decoder is obtained by merging the states of the RVLC decoder with the states of the Turbo decoder. The number of computations involved in computing S_{1r} and S_{2r} to perform adaptive scaling also increase the delay. However, this is compensated by the faster convergence achieved with the use of the scale factor and the possibility of stopping the iterative decoding process once convergence is achieved. This prevents the decoder from performing unnecessary iterations.

III. Simulation Results and Analysis

The performances of the following four LTE Turbo coded QAM schemes are compared:

Scheme 1 – The LTE Turbo coded QAM system as specified in the standard with conventional decoding. No adaptive scaling and prioritised constellation mapping are used. The encoding framework is as given in Fig. 1. The decoding is as shown in Fig. 5 but without JSCD and adaptive scaling.

Scheme 2 - This scheme uses prioritised constellation mapping with highest priority given to the systematic bits. The encoding is as shown in Fig. 1. The decoding is similar to that of Scheme 1.

Scheme 3 – This scheme uses JSCD and adaptive scaling with the LTE Turbo coded QAM as specified by the standard. The decoding framework is given in Fig. 5.

Scheme 4 – This scheme is an LTE Turbo coded QAM system with prioritised constellation mapping, JSCD and SDR scaling.

In all simulations, a random alphabet source with the probability distribution given in Table I has been used. After generating the alphabets, they are grouped into packets of size P = 64 symbols. The packets are then Reversible Variable Length Coded to obtain an RVLC bit-stream as shown in Fig. 1. Normally, the length in bits, L, of each packet is transmitted as side-information because L is different for each packet. The packetization is important to prevent error propagation [11]. The RVLC bit-streams of all packets are grouped into blocks of 4096 bits since an interleaver size of 4096 bits has been used in the simulations. The parameters for the LTE Turbo code used are as follows [20]:

Generator: G = [1, g1/g2], where g1 = 15 and g2 = 13 in Octal.
Interleaver size, N = 4096 bits.
QPP Interleaver parameters: f_1= 31 and f_2 = 64
Maximum number of iterations, I = 12.
Code-rate = 1/3 and channel model: Complex AWGN.

The graphs of Bit Error Rate (BER) and Levenshtein Error Rate (LER) as a function of E_b/N_0 have been plotted over an E_b/N_0 range: 0 dB ≤ E_b/N_0 ≤ 11 dB in steps of 0.5 dB for 16-QAM and 64-QAM. The Levenshtein distance denoted by $d_L(u_1, u_2)$ between two symbol sequences u_1 and u_2, which are not necessarily of equal lengths, is defined as the number of insertions, deletions, and substitutions necessary to transform one sequence into the other [21] [22] [23]. The LER is a more appropriate measure for Variable Length Coded (VLC) symbol sources than the Symbol Error Rate (SER) because the length of the transmitted and received symbols may not necessarily be the same. Hence, LER provides a fairer measure than the SER for VLC coded symbols. E_b/N_0 is the ratio of the bit energy, E_b, to the noise power spectral density, N_0. The performance analysis has been made for both iterative and non-iterative decoding. Non-iterative curves were given to analyse the baseline turbo coding performance as was done in [16].

Fig. 8 shows the graph of BER against E_b/N_0 for iterative decoding with 16-QAM. In the low E_b/N_0 range (0 dB ≤ E_b/N_0 ≤ 4 dB) it is observed that the LTE Turbo coded 16-QAM system with JSCD, adaptive scaling, and prioritised constellation mapping (Scheme 4) provides the best performance with an average gain of 1.7 dB for BER > 10^{-1} over the conventional LTE Turbo coded system (Scheme 1). At an E_b/N_0 of 1 dB, Scheme 2, which uses only prioritised constellation mapping, outperforms Scheme 1 by 1dB. It is to be noted from a theoretical point of view the performance of the system for a BER > 10^{-1} is important only because it is revealing a new characteristic of the system whereby it is

seen that significant gains can be obtained in this BER region using the proposed technique. However, from a practical point of view the performance of the system for BER > 10^{-1} is not really relevant. In the high E_b/N_0 range (4.5 dB $0 \leq E_b/N_0 \leq 11$ dB), it is observed that prioritised constellation mapping is not really beneficial. For instance, Scheme 2 provides the worst performance. A possible explanation is that with iterative decoding, in the high E_b/N_0 range (4.5 dB $0 \leq E_b/N_0 \leq 11$ dB), convergence takes place. As such, giving more protection to the systematic bits does not provide further gains. Also, since lower protection has been given to the parity bits, this can lead to performance degradation. Over this E_b/N_0 range, Scheme 4 that uses prioritised constellation mapping with JSCD and adaptive scaling provide the best performance with an average gain of 0.6 dB in E_b/N_0 over Scheme 1.

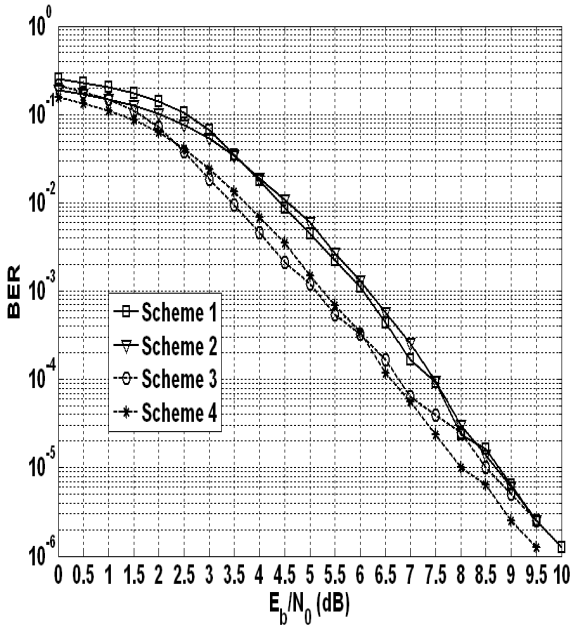

Fig. 8. Iterative BER performance with N = 4096 and 16-QAM.

Fig. 9 shows the graph of LER against E_b/N_0 for iterative decoding with 16-QAM. In the low E_b/N_0 range (0 dB $\leq E_b/N_0 \leq 4$ dB) it is observed that the LTE Turbo coded 16-QAM system with JSCD, adaptive scaling, and prioritised constellation mapping (Scheme 4) provides the best performance with an average gain of 1.5 dB for LER > 10^{-1} over the conventional LTE Turbo coded system (Scheme 1). At an E_b/N_0 of 1 dB, Scheme 2, which uses only prioritised constellation mapping outperforms Scheme 1 by 0.6 dB. In the high E_b/N_0 range (4.5 dB $0 \leq E_b/N_0 \leq 11$ dB), Scheme 2 provides the worst performance and Scheme 4, which uses prioritised constellation mapping with JSCD and adaptive scaling provides the best performance with an average gain of 0.8 dB in E_b/N_0 over Scheme 1.

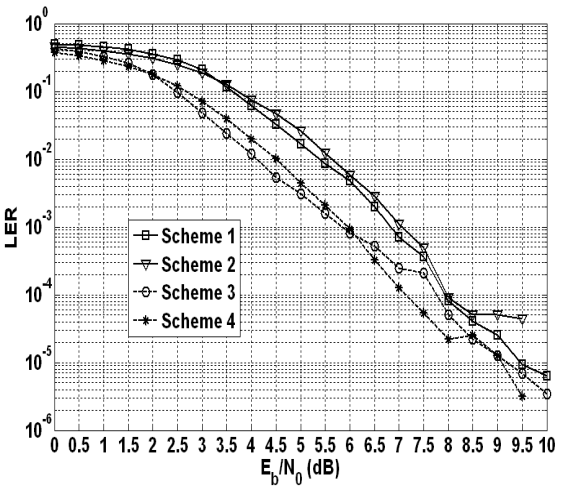

Fig. 9. Iterative LER performance with N = 4096 and 16-QAM.

Fig. 10 shows the graph of average number of iterations versus E_b/N_0 over the range 4.5 dB $\leq E_b/N_0 \leq$ 12 dB, when 16-QAM is used. Schemes 3 and 4, which employ SDR based scaling with a stopping criterion, show a progressive decrease in the number of iterations required as the E_b/N_0 increases. For example at an E_b/N_0 of 8.5 dB, Scheme 4 consumes seven iterations less than Schemes 1 and 2. However, at the same E_b/N_0 of 8.5 dB, Scheme 4 consumes 1.5 iterations more than Scheme 3 due to performance loss as a result of using prioritised constellation mapping after convergence. The number of iterations required by Schemes 1 and 2 remains fixed at 12 throughout the E_b/N_0 range.

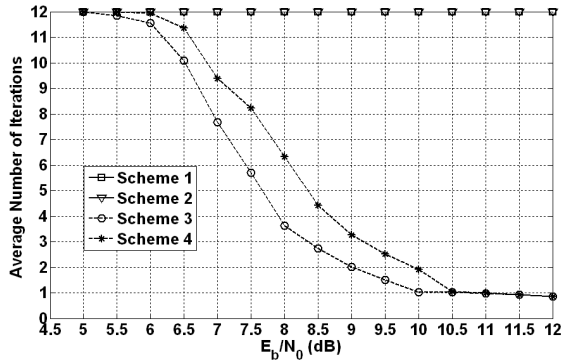

Fig. 10. Average number of iterations vs E_b/N_0 for N = 4096 and 16-QAM.

Fig. 11 shows the graph of BER against E_b/N_0 for non-iterative decoding with 16-QAM. In the low E_b/N_0 range (0 dB $\leq E_b/N_0 \leq$ 4 dB), it is observed that Scheme 4 outperforms all the other schemes with an average gain of 1.5 dB over Scheme 1 and 0.8 dB over Scheme 2. However, with non-iterative decoding, convergence does not take place and the use of prioritized constellation mapping does not lead to degradation at BERs below 10^{-1}. This is observed in the high E_b/N_0 range (4.5 dB $0 \leq E_b/N_0 \leq 11$ dB) whereby Scheme 4 outperforms Scheme 1 by 0.8 dB on average.

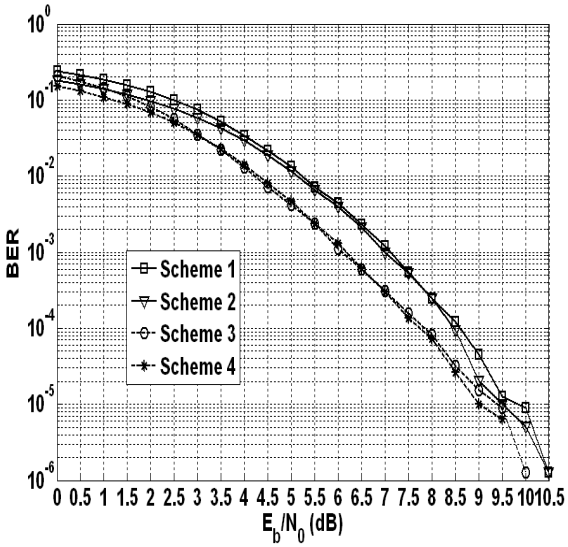

Fig. 11. Non-Iterative BER performance with N = 4096 and 16-QAM.

Fig. 12 shows the graph of LER against E_b/N_0 for non-iterative decoding with 16-QAM. In the low E_b/N_0 range (0 dB $\leq E_b/N_0 \leq$ 4 dB), it is observed that Scheme 4 outperforms all the other schemes with an average gain of 1.5 dB over Scheme 1 and 1 dB over Scheme 2. However, the use of prioritized constellation mapping does not lead to degradation at BERs below 10^{-1} since with non-iterative decoding, convergence does not take place. This is observed in the high E_b/N_0 range (4.5 dB $0 \leq E_b/N_0 \leq 11$ dB) whereby Scheme 4 outperforms Scheme 1 by 1 dB on average.

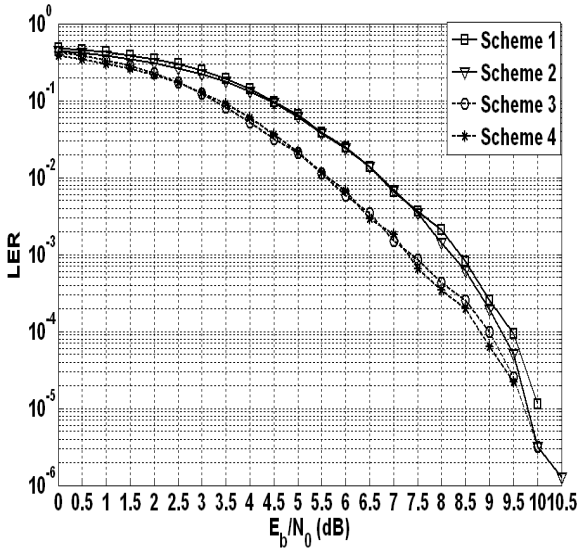

Fig. 12. Non-Iterative LER performance with N = 4096 and 16-QAM.

Fig. 13 shows the graph of BER against E_b/N_0 for iterative decoding with 64-QAM. In the low E_b/N_0 range (0 dB $\leq E_b/N_0 \leq$ 4 dB), it is observed that Scheme 4 outperforms all the other schemes with an average gain of 4.5 dB over Scheme 1 and 1 dB over Scheme 2 at BERs above 10^{-1}. However, with 64-QAM, convergence does not take place and the use of prioritized constellation mapping does not lead to degradation at BERs below 10^{-1}. This is observed in the high E_b/N_0 range (4.5 dB $0 \leq E_b/N_0 \leq 11$ dB) whereby Scheme 2 outperforms Scheme 1 by 2 dB on average and Scheme 4 outperforms Scheme 1 by 3 dB on average. The combination of LTE-Turbo codes with 16-QAM did not lead to performance improvements in the high E_b/N_0 range when prioritized constellation mapping was used. However, with 64-QAM, prioritized constellation mapping provides an improvement over the whole E_b/N_0 range. In [1], when 64-QAM was combined with a 4-state turbo code with g1 = 5 and g2 = 7 in octal, a degradation was obtained due to convergence. However, with LTE Turbo code, this was observed with only 16-QAM. With 64-QAM, prioritised constellation mapping still provided the gain especially when combined with JSCD and SDR scaling. One possible explanation is that convergence depends on both the free distance of the turbo code and the modulation scheme. Since LTE Turbo codes have a higher free distance than the turbo code in [1], a gain was observed for 64-QAM throughout the E_b/N_0 range.

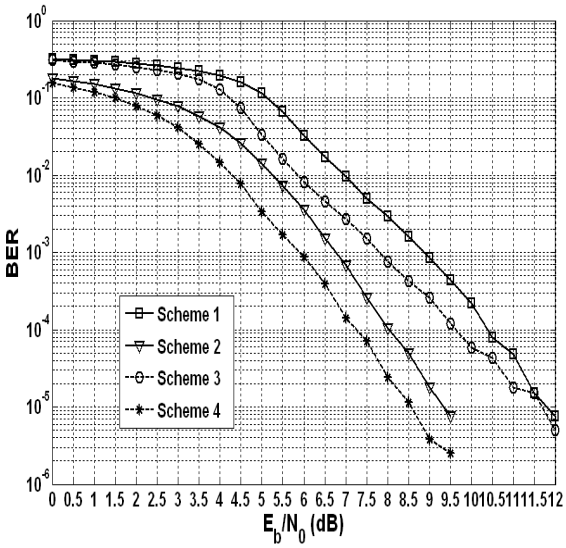

Fig. 13. Iterative BER performance with N = 4096 and 64-QAM.

Fig. 14 shows the graph of LER against E_b/N_0 for iterative decoding with 64-QAM. In the low E_b/N_0 range (0 dB $\leq E_b/N_0 \leq$ 4 dB), it is observed that Scheme 4 outperforms all the other schemes with an average gain of 4.5 dB over Scheme 1 and 1.5 dB over Scheme 2. However, with 64-QAM, convergence does not take place and the use of prioritized constellation mapping does not lead to degradation at LERs below 10^{-1}. This is observed in the high E_b/N_0 range (4.5 dB $0 \leq E_b/N_0 \leq 11$ dB) whereby scheme 2 outperforms Scheme 1 by 1 dB on average and Scheme 4 outperforms Scheme 1 by 2 dB on average over this range of E_b/N_0. Also, it can be noted that Scheme 4 outperforms Scheme 3 by 2.7 dB on average in the low E_b/N_0 range and by almost 2.2 dB in the high E_b/N_0 range.

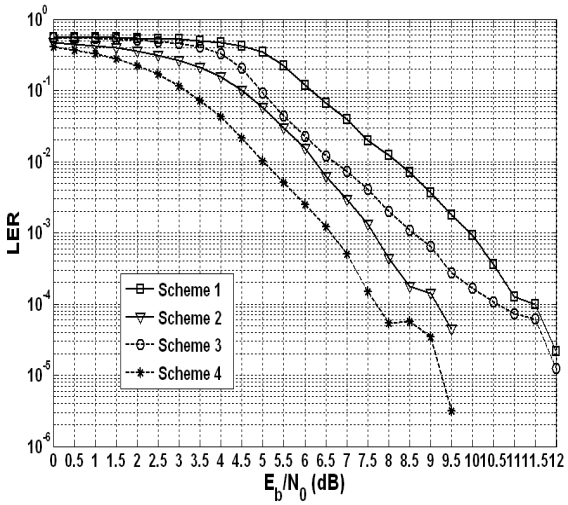

Fig. 14. Iterative LER performance with N = 4096 and 64-QAM.

Fig. 15 shows the graph of average number of iterations versus E_b/N_0 over the range 4.5 dB $\leq E_b/N_0 \leq$ 12 dB, when 64-QAM is used. Schemes 3 and 4, which employ SDR based scaling with a stopping criterion, show a progressive decrease in the number of iterations required as the E_b/N_0 increases. For example at an E_b/N_0 of 10 dB, Scheme 4 consumes nine iterations less than Schemes 1 and 2. However, it is very interesting to note that at the same E_b/N_0 of 10 dB, Scheme 4 also consumes 8 iterations less than Scheme 3, which is not the case when 16-QAM is used. The number of iterations required by Schemes 1 and 2 remains fixed at 12.

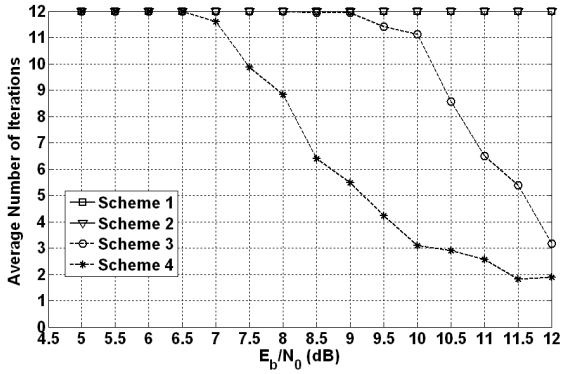

Fig. 15. Average number of iterations vs E_b/N_0 for N = 4096 and 64-QAM.

Fig. 16 shows the graph of BER against E_b/N_0 for non-iterative decoding with 64-QAM. In the low E_b/N_0 range (0 dB $\leq E_b/N_0 \leq$ 4 dB), it is observed that Scheme 4 outperforms all the other schemes with an average gain of 4.5 dB over Scheme 1 and 0.5 dB over Scheme 2. However, with non-iterative decoding, convergence does not take place and the use of prioritized constellation mapping does not lead to degradation at BERs below 10^{-1}. This is observed in the high E_b/N_0 range (4.5 dB $0 \leq E_b/N_0 \leq 11$ dB) whereby Scheme 4 outperforms Scheme 1 by 4.5 dB on average and Scheme 2 by 0.7 dB on average over this range of E_b/N_0. Also, it can be noted from Fig. 16 that Scheme 4 outperforms Scheme 3 by 3.5 dB on average in both the low E_b/N_0 range and the high E_b/N_0 range.

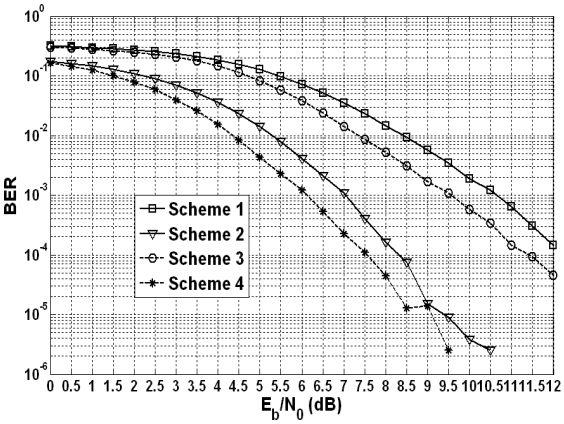

Fig. 16. Non-Iterative BER performance with N = 4096 and 64-QAM.

Fig. 17 shows the graph of LER against E_b/N_0 for non-iterative decoding with 64-QAM. In the low E_b/N_0 range (0 dB $\leq E_b/N_0 \leq$ 4 dB), it is observed that Scheme 4 outperforms all the other schemes with an average gain of 4.5 dB over Scheme 1 and 1 dB over Scheme 2. However, the use of prioritized constellation mapping does not lead to degradation at LERs below 10^{-1}. This is observed in the high E_b/N_0 range (4.5 dB $0 \leq E_b/N_0 \leq 11$ dB) whereby Scheme 4 outperforms Scheme 1 by 4.5 dB on average and Scheme 2 by 1 dB on average over this range of E_b/N_0. Also, it can be noted from Fig. 17 that Scheme 4 outperforms Scheme 3 by 3.5 dB on average both in the low E_b/N_0 range and the high E_b/N_0 range.

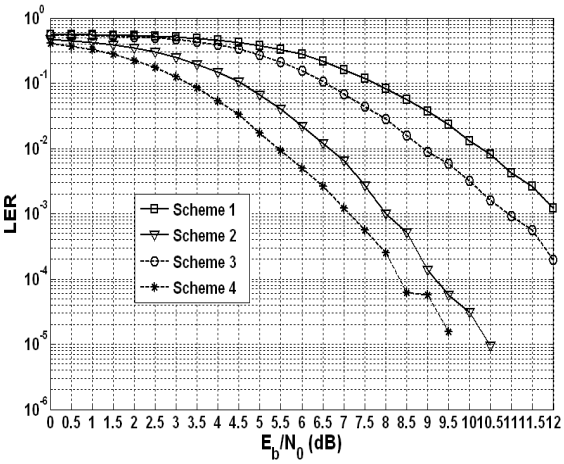

Fig. 17. Non-Iterative LER performance with N = 4096 and 64-QAM.

IV. CONCLUSION AND FUTURE WORKS

This paper proposed an efficient LTE Turbo coded QAM scheme with JSCD, adaptive scaling, and prioritised constellation mapping. At the encoder side a re-ordering mechanism is used to map the systematic bits of the Turbo

encoder on the most strongly protected points of the QAM constellation. To enhance the decoding process, JSCD is used to incorporate a-priori source statistics and adaptive SDR based scaling is also performed. When 16-QAM is used, at BERs above 10^{-1}, the proposed scheme provides a significant performance gain of 1.7 dB over a conventional LTE Turbo coded scheme. For BERs below 10^{-1}, the use of prioritised constellation mapping with conventional decoding degrades performance as a result of convergence. Hence, for BERs below 10^{-1}, it is preferable to use JSCD and SDR scaling with prioritised constellation mapping with 16-QAM, which achieves a gain of 0.6 dB on average over a conventional LTE Turbo coded scheme. However, with 64-QAM, the proposed scheme outperforms a conventional LTE Turbo coded scheme at all BERs because there is no performance degradation due to prioritised constellation mapping. The proposed scheme provides a significant average performance gain of 3 dB over a conventional LTE Turbo coded scheme. Overall, the combination of prioritised constellation mapping with JSCD and SDR based scaling appears promising for LTE Turbo coded QAM systems.

Several interesting future works can be envisaged from the scheme proposed in this work. For, e.g., the prioritised constellation mapping scheme could be optimised so that performance gains could be obtained in the high E_b/N_0 range (4.5 dB $0 \leq E_b/N_0 \leq 11$ dB) also when 16-QAM is used with conventional decoding. A more complex channel, such as selective fading, could be considered for the performance analysis of Scheme 4. Puncturing could be used to obtain higher coding rates, e.g., ½ to extend the analysis of the four schemes presented.

ACKNOWLEDGMENT

The authors would like to thank the University of Mauritius for providing the necessary facilities for conducting this research as well as the Tertiary Education Commission of Mauritius.

REFERENCES

[1] Tulsi Pawan Fowdur, Yogesh Beeharry, and K. M. Sunjiv Soyjaudah, "Performance of Turbo Coded 64-QAM with Joint Source Channel Decoding, Adaptive Scaling and Prioritised Constellation Mapping," CTRQ, 6th International Conference on Communciation Theory, Reliability and Quality of Service, Venice, Italy, April 2013, pp. 35-41.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," IEEE International Conference on Communications, (ICC), Geneva, May 1993, vol. 2, pp. 1064-1070.

[3] S. Sesia, I. Toufik, and M. Baker, LTE – The UMTS Long Term Evolution: From Theory to Practice, John Wiley & Sons, Ltd, 2009, ISBN: 978-0-470-69716-0.

[4] 3GPP, "Technical Specifications Rel.8.", 2009.

[5] 3GPP2 C.S0024-B, "CDMA 2000 High Rate Packet Data Air Interface Specification" Version 1.0, May 2006. Available Online: http://www.3gpp2.org/Public_html/specs/C.S0024-B_v1.0_060522.pdf [Accessed: November 2012].

[6] J. Zyren, "Home Plug Green PHY Overview," Technical Paper, Atheros Communications, 2010.

[7] I. Lee, C.E.W. Sundberg, S. Choi, and W. Lee, "A modified medium access control algorithm for systems with iterative decoding," IEEE Transactions on Wireless Communications, vol. 5, no. 2, 2006, pp. 270-273.

[8] A.H. Murad and T.E. Fuja, "Joint source‑channel decoding of variable length encoded sources," Proceedings of the Information Theory Workshop (ITW). Killarney, Ireland, June. 1998, pp. 94‑95.

[9] M. Jeanne, J.C. Carlach, and P. Siohan, "Joint source‑channel decoding of variable length codes for convolutional codes and turbo codes," IEEE Trans Commun, vol. 53, no. 1, 2005, pp. 10‑15.

[10] W. Xiang and P. Lu, "Bit-Based Joint Source-Channel Decoding of Huffman Encoded Markov Multiple Sources," Journal of Networks, vol. 5, no. 4, 2010, pp. 443-450.

[11] T.P. Fowdur and K.M.S. Soyjaudah "Performance of joint source–channel decoding with iterative bit combining and detection," Ann. Telecommun. vol. 63, 2008, pp. 409-423.

[12] J. Vogt and A. Finger, "Improving the MAX-Log-MAP Turbo decoder," Electr. Lett., vol. 36, no. 23, Nov. 2000, pp. 1937-1939.

[13] T. Gnanasekaran and K. Duraiswamy, "Performance of Unequal Error Protection Using MAP Algorithm and Modified MAP in AWGN and Fading Channel," Journal of Computer Science, vol. 4, no. 7, 2008, pp. 585-590.

[14] Y. Lin, W. Hung, W. Lin, T. Chen, and E. Lu, "An Efficient Soft-Input Scaling Scheme for Turbo Decoding," IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing Workshops, vol. 2, 2006, pp. 252-255.

[15] Y. Wu, B. Woerner, and J. Ebel, "A Simple Stopping Criterion for Turbo Decoding," IEEE Commun. Lett., vol. 4, no. 8, Aug. 2000, pp. 258-260.

[16] H. Lüders, A. Minwegen, and P. Vary, "Improving UMTS LTE Performance by UEP in High Order Modulation," 7th International Workshop on Multi-Carrier Systems & Solutions (MC-SS 2009), Herrsching, Germany, 2009, pp. 185-194.

[17] Y. Takishima, M. Wada, and H. Murakami, "Reversible variable length codes," IEEE Trans. on Commun., vol. 43, 1995, pp. 158–162.

[18] Farooq Khan, "LTE for 4G Mobile Broadband Air Interface Technologies and Performance," Cambridge University Press, 2009.

[19] B. Vucetic and J. Yuan, Turbo Codes: Principles and Applications, Kluwer Academic Publishers, 2000.

[20] 3GPP TS 36.212 version 9.2.0: "Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and Channel Coding," May 2010.

[21] V. I. Levenshtein, "Binary codes capable of correcting spurious insertions and deletions of ones," Probl. Perdachi Inform., vol. 1, no. 1, pp. 12-25, 1965.

[22] V. I. Levenshtein, "Binary codes with correction of deletions, insertions and substitutions of symbols," Dokl. Akad. Nank. SSSR, vol. 163, pp. 845-848, 1965.

[23] T. Okuda, E. Tanaka, and T. Kasai, "A method for the correction of the garbled words based on the Levenshtein metric," IEEE Trans. Comput., vol. C-25, no. 2, pp. 172-176, Feb. 1976.