Performance of Turbo Coded 64-QAM with Joint Source Channel Decoding, Adaptive Scaling and Prioritised Constellation Mapping

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

Abstract— **Turbo coded 64-QAM systems have been adopted by standards such as 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 proposes a combination of Joint Source Channel Decoding (JSCD), adaptive Sign Division Ratio (SDR) based scaling and prioritised constellation mapping, to improve the performance of Turbo coded 64-QAM. 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 64-QAM constellation and provides greater protection to the systematic bits of the Turbo encoder. Simulation results show that at Bit Error Rates (BERs) above 10^{-1}, the combination of these three techniques achieves an average gain of 2.5 dB over a conventional Turbo coded 64-QAM system. However, at BERs below 10^{-1}, the combination of only JSCD and SDR scaling provides an average gain of 1 dB.**

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

I. INTRODUCTION

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

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 [7] 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 [7] was developed by Jeanne *et.al* [8] and more recently Xiang and Lu [9] proposed a JSCD scheme for Huffman encoded multiple sources, which could exploit the a-priori bit probabilities in multiple sources. Also, Fowdur and Soyjaudah [10] 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 [11] used a fixed scale factor to improve the Max-Log-MAP Turbo decoding algorithm, while Gnanasekaran and Duraiswamy [12] proposed a modified MAP algorithm using a fixed scale factor. Interestingly though Lin *et.al* [13] proposed a scaling scheme that extended the Sign Division Ratio (SDR) technique of Wu *et.al* [14] 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 64-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* [15].

In contrast with previous works, which have mostly considered the schemes developed to improve the performance of Turbo codes independently, this paper analyses the performance of a Turbo coded 64-QAM scheme, which integrates three different techniques. Firstly, at the encoder side, prioritized constellation mapping [15] is performed so that the systematic bits output by the Turbo encoder are given the highest protection when they are mapped onto the 64-QAM constellation. The second technique employed is JSCD [7], [10], which exploits a-priori source statistics during Turbo decoding. The final technique used is adaptive extrinsic scaling based on the SDR criterion [13]. Significant performance gains are obtained for both iterative and non-iterative decoding 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.

 ISBN: 978-1-61208-263-9

II. SYSTEM MODEL

The complete transmission system is shown in Fig. 1.

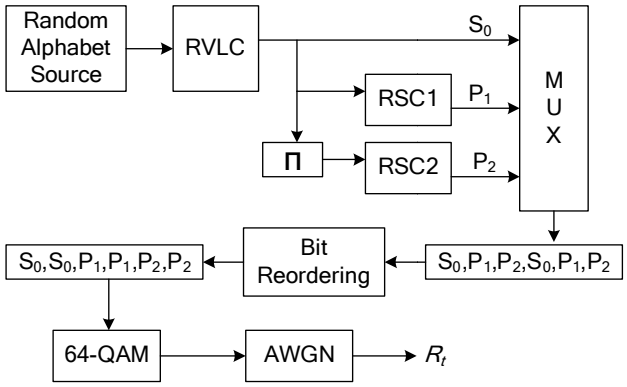

Fig. 1 Transmitter with prioritised constellation mapping.

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 [16]. The coded bits are fed to a Turbo encoder, which consists of a parallel concatenation of two Recursive Systematic Convolutional (RSC) encoders, RSC1 and RSC2, separated by an interleaver, $\prod$. The Turbo encoder generates a systematic stream, S_0 and two parity streams P_1 and P_2. To achieve prioritized constellation mapping, such that the systematic bits, S_0, are placed at the most strongly protected points on the 64-QAM constellation, bit reordering [15] must be performed after the multiplexing process. The bit reordering is performed on a group of six bits at a time since six bits are mapped onto one complex 64-QAM symbol.

From Fig. 1, it is observed that after bit re-ordering, the parity bits S_0 occupy the first two positions of the six bits that are mapped on one symbol of the 64-QAM constellation shown in Fig. 2. In this constellation, the bits found in the first two positions are most protected, while the bits found in the last two positions receive the lowest protection.

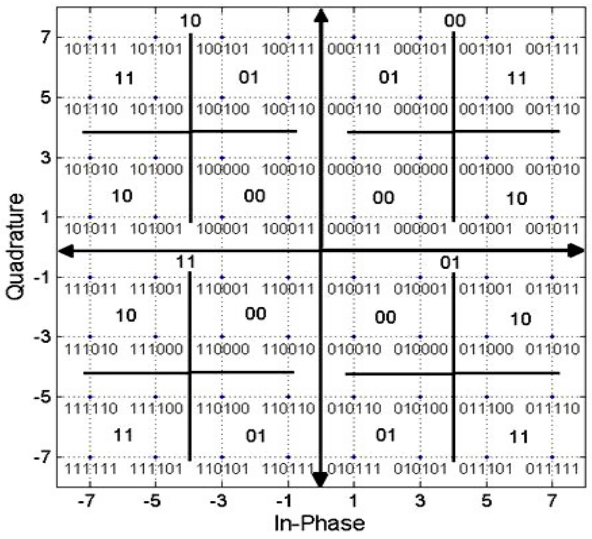

Fig. 2 64-QAM constellation with major and minor quadrants.

This can be explained by considering the four major and 16 minor quadrants of this constellation. The major quadrants are distinguished by the first two bits of the constellation point, for example, in the upper right major quadrant, the first two bits are 00. Hence, if the de-mapper only distinguishes between the four quadrants correctly, the first two bits are correctly de-mapped. Each major quadrant is divided into four minor quadrants, which are distinguished using the 3rd and 4th bits of the constellation points. Therefore, with bit ordering [15], 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 64-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. 3. The received symbols R_t are fed to a soft-output 64-QAM de-mapper to produce soft bits. These soft bits are then de-multiplexed and 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 [7] and [10]. This results into a composite trellis structure with which JSCD can be performed. With JSCD the computation of the branch transition probability is modified.

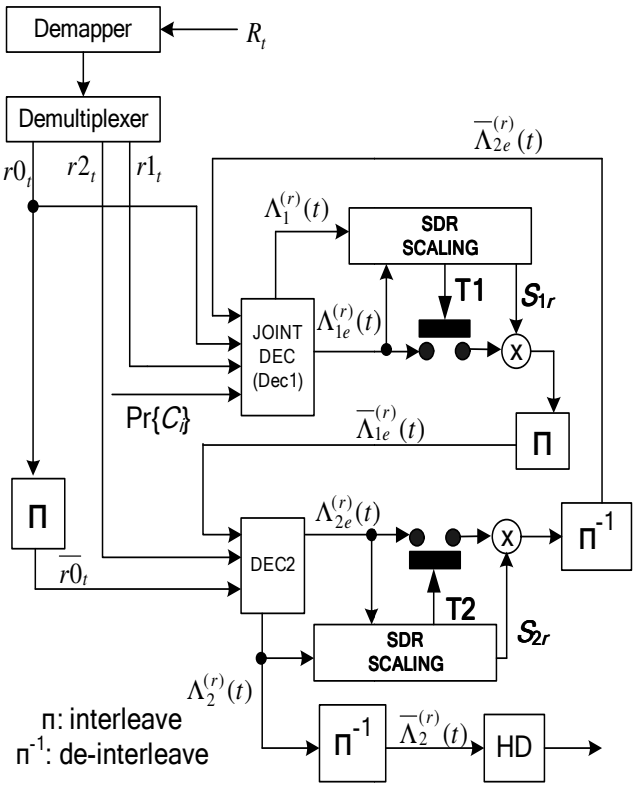

Fig. 3 Turbo decoding system with JSCD and SDR scaling.

Assuming that the Max-Log-MAP algorithm [17] is used, the branch metric probability for the joint decoder is computed as follows:

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

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 [10].

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. 4 [10].

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 [10]:

$$\Pr(M_t = IA, i = 0 \mid M_{t-1} = F) = PA + PB = P01 \quad (2)$$

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 in computing the branch metric probability as per equation (1) [10]. 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:

$$\overline{\alpha_t^1}(l) = \max\left(\overline{\alpha}_{t-1}^1(l') + \overline{\gamma_t^{1(i)}}(l',l) \right) for \;\; 0 \le l' \le M_j - 1 \quad (3)$$

TABLE I. RVLC CODEWORDS

Symbol	Probabilty	RVLC [16]
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

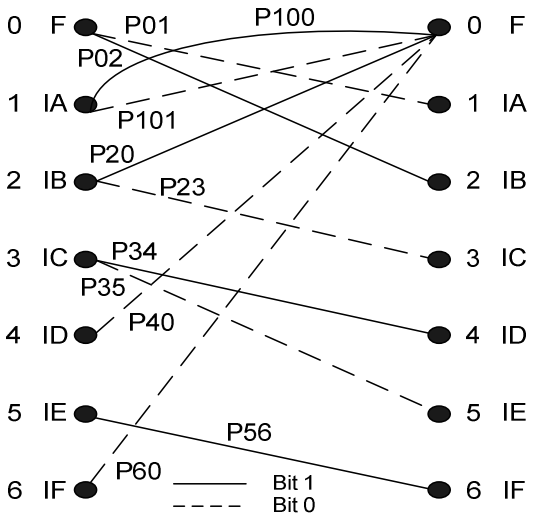

Fig. 4. Bit level trellis of RVLC decoder [10].

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 [10]. The backward recursive variable, $\overline{\beta_t^1}(l)$, is computed as follows:

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

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

$$\Lambda_1^{(r)}(t) = \max\left(\overline{\alpha}_{t-1}^1(l') + \overline{\gamma_t^{1(1)}}(l',l) + \overline{\beta}_t^1(l) \right) \quad (5)$$
$$- \max\left(\overline{\alpha}_{t-1}^1(l') + \overline{\gamma_t^{1(0)}}(l',l) + \overline{\beta}_t^1(l) \right) for \;\; 0 \le l' \le M_j - 1$$

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

$$\Lambda_{1e}^{(r)}(t) = \Lambda_1^{(r)}(t) - \frac{2}{\sigma^2} r0_t - \overline{\Lambda}_{2e}^{(r-1)}(t) \quad (6)$$

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:

$$S_{1r} = \frac{1}{N} \sum_{t=1}^{N} f\left(\Lambda_{1e}^{(r)}(t), \Lambda_1^{(r)}(t) \right) \quad (7)$$

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 [13], 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 [13]. The a-priori probability, $p_t^2(i)$, is computed as follows and sent to decoder 2:

$$p_t^2(i)=\left\{\begin{array}{l}\dfrac{\exp(\overline{\Lambda}_{1e}^{(r)}(t))}{1+\exp(\overline{\Lambda}_{1e}^{(r)}(t))}\ for\ \ i=1\\ \dfrac{1}{1+\exp(\overline{\Lambda}_{1e}^{(r)}(t))}\ for\ \ i=0\end{array}\right\} \quad (7)$$

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

$$\overline{\gamma_t^{2(i)}}(l',l\ )=\log\left[p_t^2(i)\right]-\left(\frac{[\overline{r0}_t-x0_t]^2+[r2_t-x2_t]^2}{2\sigma^2}\right) \quad (8)$$

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:

$$\overline{\alpha_t^2}(l)=\max\left(\overline{\alpha}_{t-1}^2(l')+\overline{\gamma_t^{2(i)}}(l',l)\right)for\ \ 0\le l'\le M_s-1 \quad (9)$$

$$\overline{\beta_t^2}(l)=\max\left(\overline{\beta}_{t-1}^2(l')+\overline{\gamma_t^{2(i)}}(l,l')\right)for\ \ 0\le l'\le M_s-1 \quad (10)$$

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

$$\Lambda_2^{(r)}(t)=\max\left(\overline{\alpha}_{t-1}^2(l')+\overline{\gamma_t^{2(1)}}(l',l)+\overline{\beta}_t^2(l)\right)-\max\left(\overline{\alpha}_{t-1}^2(l')+\overline{\gamma_t^{2(0)}}(l',l)+\overline{\beta}_t^2(l)\right)for\ \ 0\le l'\le M_j-1 \quad (11)$$

$$\Lambda_{2e}^{(r)}(t)=\Lambda_2^{(r)}(t)-\frac{2}{\sigma^2}\overline{r0}_t-\overline{\Lambda}_{1e}^{(r)}(t) \quad (12)$$

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

$$S_{2r}=\frac{1}{N}\sum_{t=1}^{N}f\left(\Lambda_{2e}^{(r)}(t),\Lambda_2^{(r)}(t)\right) \quad (13)$$

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 per equation (7) 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 Turbo coded 64-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 Turbo coded 64-QAM schemes are compared:

Scheme 1 – The Turbo coded 64-QAM system with JSCD, adaptive scaling and prioritised constellation mapping. The encoding and decoding frameworks are given in Fig. 1 and Fig. 3, respectively.

Scheme 2 - This scheme only uses prioritised constellation mapping. The encoding is as per Fig. 1, but the decoding does not include JSCD or adaptive scaling.

Scheme 3 – This scheme uses JSCD and adaptive scaling and its decoder is similar to Scheme 1. However, prioritised constellation mapping is not performed, as such, the bit re-ordering block of the encoder shown in Fig. 1 is omitted.

 ISBN: 978-1-61208-263-9

Scheme 4 – This scheme is a conventional Turbo coded 64-QAM system without 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. The RVLC bit-streams of all packets are grouped into blocks of 4056 bits since an interleaver size of 4056 bits has been used in the simulations. The parameters for the Turbo code used are as follows:

Generator: G = [1, g1/g2], where g0 = 7 and g1 = 5 in Octal.
Interleaver size, N = 4056 bits.
Maximum number of iterations, I = 12.
Code-rate = 1/3 and channel model: Complex AWGN.

The graphs of Bit Error Rate (BER) as a function of Eb/No have been plotted separately over a low Eb/No range: 0 dB ≤ Eb/No ≤ 3 dB and a high Eb/No range: 3.5 dB 0 ≤ Eb/No ≤ 6.5 dB in steps of 0.5 dB. Eb/No is the ratio of the bit energy, Eb to the noise power spectral density, No. It is to be noted that the transition from the low Eb/No range to the high Eb/No range is essentially a continuity from 3 dB to 3.5 dB and up to 6.5 dB. The performance analysis has also been made for both iterative and non-iterative decoding.

Fig. 5 shows the graph of BER against Eb/No for iterative decoding over the low Eb/No range. It is observed that the Turbo coded 64-QAM system with JSCD, adaptive scaling and prioritised constellation mapping (Scheme 1) provides the best performance with an average gain of 2.5 dB for BER > 10^{-1} over the conventional Turbo coded system (Scheme 4). At an Eb/No of 1 dB, Scheme 1 also provides a gain of about 1.5 dB over Scheme 3, which does not employ prioritised constellation mapping.

Moreover, Scheme 2, which uses only prioritised constellation mapping outperforms Scheme 3 by 1dB at an Eb/No = 1 dB. It is to be noted from a theoretical point of view the performance of the system for a BER > 10^{-1} is important 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.

Fig. 6 shows the graph of BER against Eb/No for iterative decoding over the high Eb/No range. In this range it is observed that prioritised constellation mapping is not beneficial. For example, Scheme 2 provides the worst performance while the performance of Scheme 1 is comparable to that of Scheme 4. A possible explanation is that with iterative decoding, in the high Eb/No range, 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 Eb/No range, Scheme 3 which uses only JSCD and adaptive scaling provides the best performance with an average gain of 1 dB in Eb/No over Scheme 1 and Scheme 4. It is to be noted that Scheme 3 outperforms Scheme 1 over this high Eb/No range because Scheme 1 suffers from a performance loss, which results from the use of prioritised constellation mapping after convergence.

Fig. 7 shows the graph of average number of iterations versus Eb/No over the range 3 dB ≤ Eb/No ≤ 6.5 dB. Scheme 1 and Scheme 3, which employ SDR based scaling with a stopping criterion, show a progressive decrease in the number of iterations required as the Eb/No increases. For example at an Eb/No of 5.5 dB, Scheme 3 consumes six iterations less than Scheme 2 and Scheme 4. However, Scheme 1 consumes on average 1.5 iterations more than Scheme 3 due to performance loss as a result of using prioritised mapping after convergence. The number of iterations required by Schemes 2 and 4 remains fixed at 12.

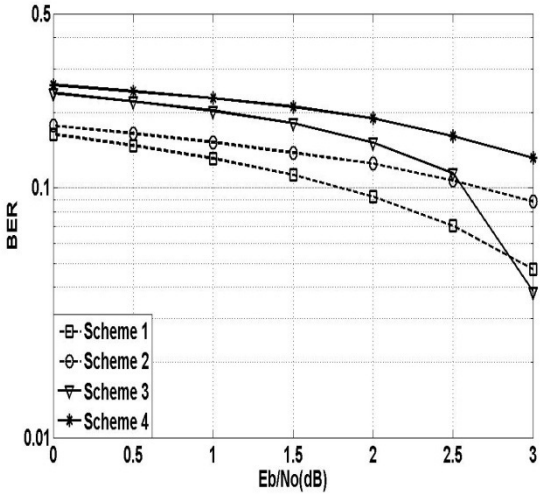

Fig. 5. Iterative low Eb/No performance with N = 4056.

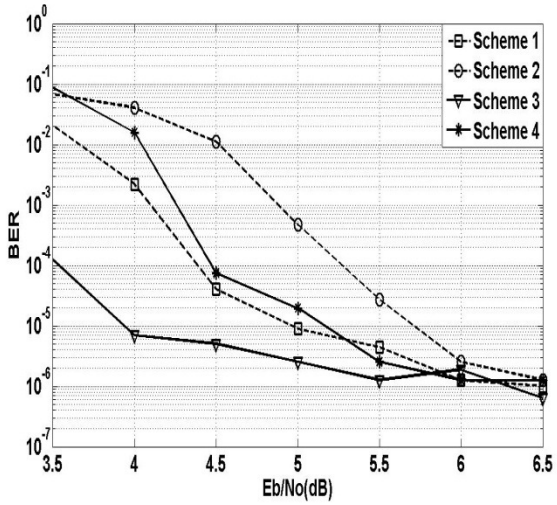

Fig. 6. Iterative high Eb/No performance with N = 4056.

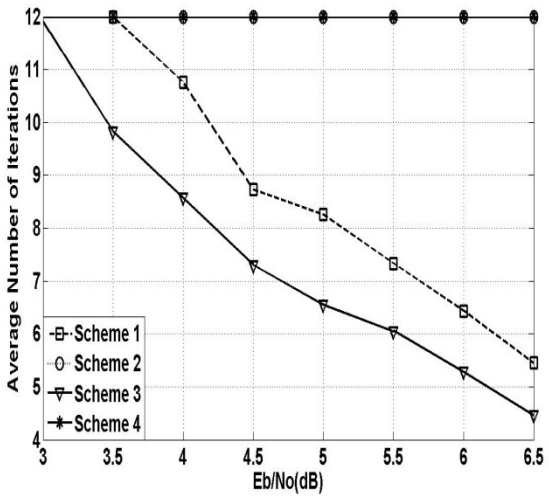

Fig. 7. Average number of iterations vs Eb/No for N = 4056 .

Fig. 8 shows the graph of BER against Eb/No for non-iterative decoding over the low Eb/No range. It is observed that Scheme 1 outperforms all the other schemes with an average gain of 2.5 dB over Scheme 4 and 2 dB over Scheme 3. 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 Fig. 9 whereby Scheme 1 outperforms Scheme 3 by 0.5 dB on average and Scheme 4 by almost 1.5 dB. It is to be noted that in [15], whereby only bit-reordering was used, it was also observed that with non-iterative decoding a performance gain is obtained throughout the Eb/No range whereas with iterative decoding convergence takes place at a certain point. Hence when prioritized constellation mapping is used the iterative scheme does not present a similar relation as the non-iterative.

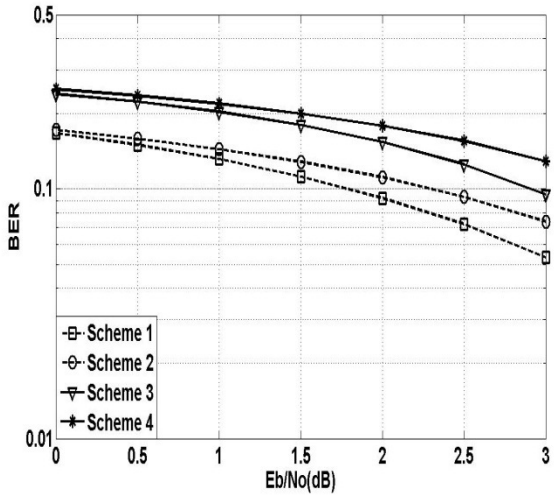

Fig. 8. Non-Iterative low Eb/No performance with N = 4056.

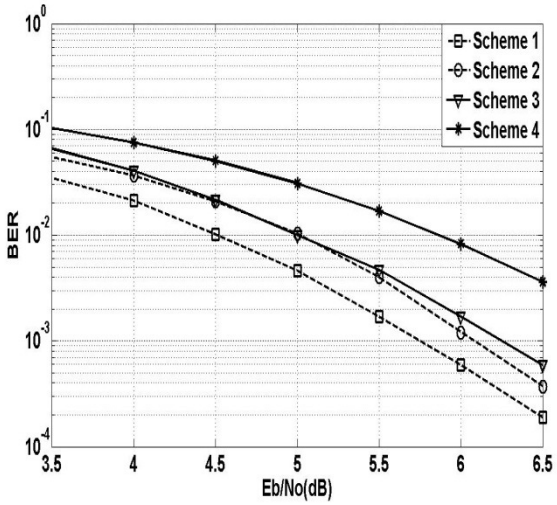

Fig. 9. Non-Iterative high Eb/No performance with N = 4056.

IV. CONCLUSION AND FUTURE WORK

This paper proposed an efficient Turbo coded 64-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 64-QAM constellation. To enhance the decoding process, JSCD is used to incorporate a-priori source statistics and adaptive SDR based scaling is also performed. At BERs above 10^{-1}, the proposed scheme provides a significant performance gain of 2.5 dB with iterative decoding over a conventional Turbo coded scheme. For BERs below 10^{-1}, the use of prioritised constellation mapping degrades performance as a result of convergence. Hence, for BERs below 10^{-1}, it is preferable to use only JSCD and SDR scaling, which achieves a gain of 1 dB on average over a conventional Turbo coded scheme. However, with non-iterative decoding, the proposed scheme, outperforms a conventional Turbo coded scheme at all BERs because there is no performance degradation due to prioritised constellation mapping. Overall, the combination of prioritised constellation mapping with JSCD and SDR based scaling appears promising for Turbo coded 64-QAM systems.

Several interesting future works can be envisaged from the scheme proposed in this work. A straightforward extension would be to assess its suitability for communication systems such as LTE. A more challenging future work would be to use JSCD schemes, which are less complex and hence do not incur significant delays while still allowing the exploitation of source statistics. The prioritised constellation mapping scheme could also be optimised so that performance gains could be obtained in the high Eb/No range also. Finally, investigations could be made on how to extend the scheme to block Turbo codes and also on the possibility of using bit interleaved coded modulation.

 ISBN: 978-1-61208-263-9

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] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes", IEEE International Conference on Communications, ICC 93. Geneva, vol. 2, 23-26 May 1993, pp.1064-1070.

[2] 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.

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

[4] 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].

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

[6] 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(2), 2006, pp.270-273.

[7] 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.

[8] 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(1), 2005, pp.10-15.

[9] W. Xiang and P. Lu, "Bit-Based Joint Source-Channel Decoding of Huffman Encoded Markov Multiple Sources", *Journal of Networks*, vol. 5(4), 2010, pp. 443-450.

[10] 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.

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

[12] 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 (7) ,2008, pp. 585-590.

[13] Y. Lin, W. Hung, W. Lin, T. Chen, 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.

[14] 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.

[15] 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.

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

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

 ISBN: 978-1-61208-263-9