# MODIFIED MAP ALGORITHM FOR EXTENDED TURBO BCH CODES AND TURBO EQUALISERS

**T.H. Liew, B.L. Yeap, J. Woodard, L. Hanzo**
Dept. of ECS., Univ. of Southampton, SO17 1BJ, UK.
Tel: +44-703-593 125, Fax: +44-703-593 045
Email: thl97r@ecs.soton.ac.uk;
http://www-mobile.ecs.soton.ac.uk *†

## Abstract

**The Maximum A-Posteriori (MAP) algorithm is modified in order to incorporate the parity check bit of extended BCH codes into calculating the Log Likelihood Ratio (LLR) of the coded bits. The algorithm is then used jointly with channel equalisation – in a scheme referred to as turbo equalisation – in order to mitigate the effects of Inter-Symbol Interference (ISI) introduced by dispersive channels. The performance of the extended BCH codes is evaluated and compared to that of conventional BCH codes. For example, the extended BCH(32,26) code was found to outperform the conventional BCH(31,26) code by approximately $0.5 - 1.0$ dB at a bit error rate (BER) of $10^{-5}$ over dispersive Rayleigh-fading channels.**

## 1 Introduction

Conventional primary BCH codes are denoted by BCH-($n$,$k$,$d_{min}$), where $n, k, d_{min}$ are the codeword length, the number of information data bits and the minimum free distance, respectively. They are referred to as extended BCH$(n+1, k, d_{min}+1)$ codes [1], if a parity check bit is appended to the primary BCH codeword. The Maximum A-Posteriori (MAP) algorithm [2] by Bahl *et al.* has been modified for calculating the Log Likelihood Ratio (LLR) of the encoded bits. The soft outputs of the coded bits are fed back to the equaliser – in a scheme referred to as turbo equalisation – in order to mitigate the effects of Inter-Symbol Interference (ISI) introduced by dispersive channels.

Turbo equalisation [3] was first introduced by C. Douillard *et al* in 1995 for a serially concatenated rate $R = 0.5$ convolutional coded BPSK system. Specifically, Douillard et al demonstrated that the turbo equaliser was capable of mitigating the effects of Inter-Symbol Interference (ISI), provided that the channel's impulse response (CIR) is known. Instead of performing the equalisation and error correction decoding independently, better performance can be achieved by considering the channel's memory, when performing joint equalisation and decoding iteratively. Gertsman and Lodge [4] then showed that the iterative process of turbo equalisers can compensate for the performance degradations due to imperfect channel estimation. Research into combined turbo coding using convolutional constituent codes and turbo equalisation have also been investigated by Raphaeli and Zarai [5].

*The financial support of the European Union and that of the EPSRC, Swindon, UK is greatefully acknowledged
†International Conference on 3G Mobile Communications Technologies

## 2 Modified MAP Algorithm

The novel modified MAP algorithm generates, for each decoded bit $u_k$, the so-called Log Likelihood Ratio (LLR) that this bit was $+1$ or $-1$, given the received channel symbol sequence $\underline{y}$ as [2]:

$$L(u_k|\underline{y}) = \ln \frac{\sum_{\substack{(\dot{s},s)\Rightarrow \\ u_k=+1}} \alpha_{k-1}(\dot{s}).\gamma_k(\dot{s},s).\beta_k(s)}{\sum_{\substack{(\dot{s},s)\Rightarrow \\ u_k=-1}} \alpha_{k-1}(\dot{s}).\gamma_k(\dot{s},s).\beta_k(s)}, \quad (1)$$

where $(\dot{s}, s) \Rightarrow u_k = +1$ describes the set of legitimate decoding trellis transitions from the previous state $\dot{s}$ to the present state $s$ caused by the input bit $u_k = +1$. Similarly, $(\dot{s}, s) \Rightarrow u_k = -1$ represents the transitions caused by bit $u_k = -1$, while $\alpha_{k-1}(\dot{s})$, $\gamma_k(\dot{s}, s)$ and $\beta_k(s)$ are three independent probabilities [2]. The novel MAP algorithm finds $\alpha_k(s)$ and $\beta_k(s)$ for all states $s$ throughout the decoding trellis and $\gamma_k(\dot{s}, s)$ for all possible transitions from the previous states $\dot{s}$ to the current states $s$. The values of $\alpha_k(s)$ and $\beta_k(s)$ can be calculated using the forward recursion and the backward recursion [2], respectively.

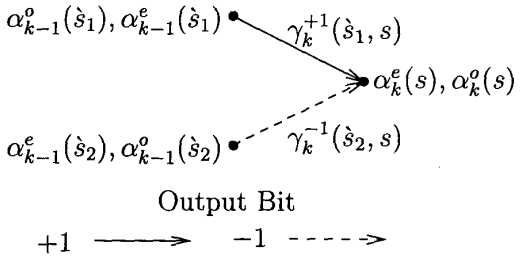


Figure 1: Forward Recursion of $\alpha^e_k(s)$ and $\alpha^o_k(s)$.

In order to incorporate the parity check bit of extended BCH codes in the MAP algorithm, we first make modifications to both the forward and backward recursion and then adjust the transition probability, which is given in Equation 1. Let us define $\alpha^e_k(s)$ as the probability that the current trellis state is $s$ at time instant $k$, and that the path arriving at this state gave an even (e) number of transmitted bits, which were $+1$, given that the received channel sequence up to this point was $\underline{y}_{j \le k}$. Similarly, $\alpha^o_k(s)$ is the corresponding probability

for the path which gave an odd ($o$) number of +1 transmitted bits. For each state in the trellis, we have $\alpha_k^e(s)$ and $\alpha_k^o(s)$, which are shown in Figure 1. Subsequently, using forward recursion and Figure 1, we can derive for a binary trellis – for example in BCH codes – the total probability of an even number of transmitted binary +1 bits as the sum of having an even number of +1s at instant $k-1$ and encountering a $-1$ during the current transition, plus the probability of an odd number of +1s at the previous stage and encounting a +1 during the current transition, yielding:

$$\alpha_k^e(s) = \alpha_{k-1}^o(\dot{s}_1) \cdot \gamma_k^{+1}(\dot{s}_1, s) + \alpha_{k-1}^e(\dot{s}_2) \cdot \gamma_k^{-1}(\dot{s}_2, s) \ . \tag{2}$$

It is seen explicitly in Figure 1 that since transition $\gamma_k^{-1}(\dot{s}_2, s)$ from the previous state $\dot{s}_2$ to the present state $s$ results in the transmitted bit being $-1$, the product $\alpha_{k-1}^e(\dot{s}_2) \cdot \gamma_k^{-1}(\dot{s}_2, s)$ is the probability of the path, which reaches state $s$ and gave an even number of transmitted bits, that were +1. Conversely, $\alpha_{k-1}^o(\dot{s}_1)$ is the probability of the path, which reaches state $\dot{s}_1$ and gave an odd number of transmitted +1 bits. Since transition $\gamma_k^{+1}(\dot{s}_1, s)$ from the previous state $\dot{s}_1$ to the present state $s$ results in the transmitted bit being +1, $\alpha_{k-1}^o(\dot{s}_1) \cdot \gamma_k^{+1}(\dot{s}_1, s)$ is the probability of the path arriving at state $s$, giving an even number of transmitted bits, that were +1. Again, using forward recursion and Figure 1, $\alpha_k^o(s)$ is derived for a binary trellis as:

$$\alpha_k^o(s) = \alpha_{k-1}^e(\dot{s}_1) \cdot \gamma_k^{+1}(\dot{s}_1, s) + \alpha_{k-1}^o(\dot{s}_2) \cdot \gamma_k^{-1}(\dot{s}_2, s) \ . \tag{3}$$

The same modification is made to the backward recursion.

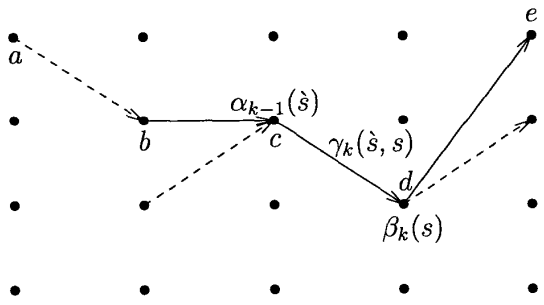


Figure 2: Probability of A Transition in the Trellis.

Now we have to adjust the transition probability expression in Equation 1. Figure 2 shows a simple trellis diagram, which commences at state $a$ and ends at state $e$. The probability of the transition from state $c$ to state $d$ is given by:

$$P(c \to d) = \alpha_{k-1}(\dot{s}) \cdot \gamma_k(\dot{s}, s) \cdot \beta_k(s) \ . \tag{4}$$

For extended BCH codes, the probability of a transition in the trellis no longer depends solely on $\alpha_{k-1}(\dot{s})$, $\gamma_k(\dot{s}, s)$ and $\beta_k(s)$. It also depends on whether the whole path gave an even or odd number of transmitted bits, which were +1. In Figure 2, path $a \to b \to c \to d \to e$ gives an odd number of transmitted bits, which were +1. Hence, the probability of the transition $c \to d$ is:

$$P(c \to d) = \alpha_{k-1}(\dot{s}) \cdot \gamma_k(\dot{s}, s) \cdot \beta_k(s) \cdot P(y_{n+1}|x_{n+1} = +1) \ , \tag{5}$$

where $y_{n+1}$ and $x_{n+1}$ are the received and transmitted parity check bit and $P(y_{n+1}|x_{n+1} = +1)$ is the probability of the path which gives an odd number of +1s. Conversely, $P(y_{n+1}|x_{n+1} = -1)$ is the probability of the path which gives an even number of +1s. For each state, we now have $\alpha_k^e(s)$, $\alpha_k^o(s)$, $\beta_k^e(s)$ and $\beta_k^o(s)$ as compared to $\alpha_k(s)$ and $\beta_k(s)$ for the conventional MAP algorithm. In other words, we have separated $\alpha_k(s)$ and $\beta_k(s)$ into two groups, respectively, depending on the nature of the paths that reached state $s$. Additionally, we also have the probability $P(y_{n+1}|x_{n+1})$ for the transmitted parity check bit. Hence, upon using Equation 5 we can rewrite the a-posteriori LLR $L(u_k|\underline{y})$ of Equation 1, while observing that the notations $\alpha^e$, $\alpha^o$, $\beta^e$ and $\beta^o$ have been simplified from $\alpha_{k-1}^e(\dot{s})$, $\alpha_{k-1}^o(\dot{s})$, $\beta_k^e(s)$ and $\beta_k^o(s)$, respectively, as in Equation 6.

By definition, in the numerator of Equation 6, the transition from the previous state $\dot{s}$ to the present state $s$ always results in a transmitted bit of +1. As it can be visualised by the help of Figure 1, the path $\alpha_{k-1}^e(\dot{s}) \cdot \gamma_k^{+1}(\dot{s}, s) \cdot \beta_k^e(s)$ results in an odd number of transmitted +1 bits, since both the paths to state $\dot{s}$, and from state $s$, gave an even number of transmitted bits, which were +1, and the transition from state $\dot{s}$ to state $s$ results in a transmitted bit of +1. Similarly, as seen in Figure 1, the path $\alpha_{k-1}^o(\dot{s}) \cdot \gamma_k^{+1}(\dot{s}, s) \cdot \beta_k^o(s)$ also results in an odd number of transmitted +1 bits. Therefore, the probabilities of these transitions are $\alpha^e \cdot \gamma_k^{+1}(\dot{s}, s) \cdot \beta^e \cdot P(y_{n+1}|x_{n+1} = +1)$ and $\alpha^o \cdot \gamma_k^{+1}(\dot{s}, s) \cdot \beta^o \cdot P(y_{n+1}|x_{n+1} = +1)$, respectively. Conversely, paths $\alpha_{k-1}^e(\dot{s}) \cdot \gamma_k^{+1}(\dot{s}, s) \cdot \beta_k^o(s)$ and $\alpha_{k-1}^o(\dot{s}) \cdot \gamma_k^{+1}(\dot{s}, s) \cdot \beta_k^e(s)$ would result in an even number of +1 transmitted bits. Hence, the probabilities of these transitions are $\alpha^e \cdot \gamma_k^{+1}(\dot{s}, s) \cdot \beta^o \cdot P(y_{n+1}|x_{n+1} = -1)$ and $\alpha^o \cdot \gamma_k^{+1}(\dot{s}, s) \cdot \beta^e \cdot P(y_{n+1}|x_{n+1} = -1)$, respectively. The same argument is applied to the denominator in Equation 6, in order to derive the probabilities of each transition. Having described the modified MAP algorithm for extended BCH codes, we proceed to discuss the principles of turbo equalisation, which will be employed in addition to turbo decoding in order to evaluate the performance of the modified MAP algorithm.

## 3 Principle of Turbo Equalisation Using Turbo Codes

With reference to the turbo equalisation schematic in Figure 3, we describe the turbo equalisation principle for a baseband receiver consisting of an equaliser and $N_d$ component decoders. For conceptual simplicity, we have omitted the channel interleaver $\pi_c$ and turbo interleavers $\pi_t$. Instead, we have only marked their positions. The superscript '$-1$' is used to represent a deinterleaver. Typically, for turbo codes there are $N_d = 2$ component decoders, whereas for non-iterative BCH decoding we have $N_d = 1$.

Each of the blocks in Figure 3 employ a Soft-In/Soft-Out (SISO) algorithm, such as the optimal Maximum $A$

$$L(u_k|\underline{y}) = \ln \frac{\sum\limits_{\substack{(\dot{s},s)\Rightarrow \\ u_k=+1}} \gamma_k^{+1}(\dot{s},s)\cdot[(\alpha^e\cdot\beta^e+\alpha^o\cdot\beta^o)\cdot P(y_{n+1}|x_{n+1}=+1)+(\alpha^e\cdot\beta^o+\alpha^o\cdot\beta^e)\cdot P(y_{n+1}|x_{n+1}=-1)]}{\sum\limits_{\substack{(\dot{s},s)\Rightarrow \\ u_k=-1}} \gamma_k^{-1}(\dot{s},s)\cdot[(\alpha^e\cdot\beta^o+\alpha^o\cdot\beta^e)\cdot P(y_{n+1}|x_{n+1}=+1)+(\alpha^e\cdot\beta^e+\alpha^o\cdot\beta^o)\cdot P(y_{n+1}|x_{n+1}=-1)]} \quad (6)$$

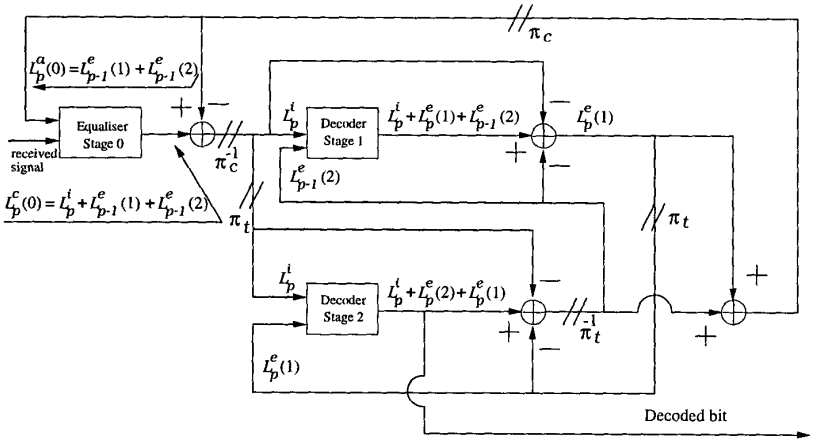


Figure 3: Structure of the turbo equaliser using $N_d = 2$ component decoders. For conceptual simplicity, we have omitted the interleavers and only marked the interleaver positions, where $\pi_c$ and $\pi_t$ represent the channel interleaver and turbo interleaver, respectively. The superscript '$-1$' is used to denote a deinterleaver.

*Posteriori* (MAP) algorithm [6] or the reduced-complexity Log-MAP algorithm [7], which yields the a *posteriori* information. The a *posteriori* information concerning a bit is the information that the SISO block generates taking into account all available sources of information about the bit. When using the MAP algorithm or the Log-MAP algorithm, we express the a *posteriori* information in terms of the Log Likelihood Ratio (LLR) [7]. For clarity, we have employed the approach used by Gertsman and Lodge [4] and expressed the LLR of the equaliser and decoder using vector notations. The superscript denotes the nature of the LLR. Namely superscript $c$ is used for the composite a *posteriori* information, $i$ denotes the combined channel and extrinsic information, $e$ indicates extrinsic information and $a$ stands for the a *priori* information. Furthermore, the subscript notation is used to represent the iteration index, while the argument within () is the index of the receiver stage. At the equaliser — which is denoted by Stage 0 in Figure 3 — the composite output a *posteriori* LLR $L_p^c(0)$ is generated by the sum of the LLR of the a *priori* information $L_p^a(0)$, and the LLR of the combined channel and extrinsic information $L_p^i$, giving:

$$L_p^c(0) = L_p^i + L_p^a(0). \quad (7)$$

Here, the Stage 0 a *priori* information $L_p^a(0)$ is obtained from the previously determined extrinsic information $L_{p-1}^e(1)$ and $L_{p-1}^e(2)$ of the other $N_d$ decoder stages — namely Stages 1 and 2 in Figure 3, — which can be expressed as:

$$L_p^a(0) = \sum_{j=1}^{N_d} L_{p-1}^e(j), \quad (8)$$

where $L_{p-1}^e(j)$ is the extrinsic LLR of the $j$th decoder stage from the previous iteration, namely from $p-1$. Upon substituting Equation 8 into Equation 7, the composite a *posteriori* LLR $L_p^c(0)$ at the output of the equaliser in Figure 3 becomes:

$$L_p^c(0) = L_p^i + \sum_{j=1}^{N_d} L_{p-1}^e(j). \quad (9)$$

The $d$th systematic decoder of the $p$th iteration — *i.e.* at the inputs of Stages 1 and 2 in Figure 3 — receives the sum of the augmented a *priori* information $L_p^a(d)$, as well as the combined channel and extrinsic information LLR $L_p^i$ from the equaliser. The augmented a *priori* information at the $p$th iteration can be expressed as:

$$L_p^a(d) = \sum_{j=1}^{d-1} L_p^e(j) + \sum_{j=d+1}^{N_d} L_{p-1}^e(j). \quad (10)$$

From Equation 10, we observe that the a *priori* information $L_p^a(d)$ consists of the extrinsic information from the decoders of the **previous stages** — namely for $j < d$ in the **current iteration** — and from the decoders of the **later stages** — namely for $j > d$ from the **previous iteration** — but does not include any extrinsic information from Stage $d$.

For the $d$th stage systematic decoder — *i.e.* at the inputs of the Stages 1 and 2 in Figure 3 — the composite a *posteriori* LLR $L_p^c(d)$ can be expressed as the sum of the augmented a *priori* information $L_p^a(d)$, the extrinsic information $L_p^e(d)$ produced by the decoder stage and the combined channel LLR and extrinsic LLR $L_p^i$ from the

equaliser:

$$L_p^c(d) = L_p^a(d) + L_p^e(d) + L_p^i. \quad (11)$$

By substituting Equation 10 into Equation 11 for our specific scheme, we obtain at the output of decoder Stages 1 and 2 in Figure 3:

$$\begin{aligned} L_p^c(d) &= L_p^a(d) + L_p^e(d) + L_p^i \\ &= \sum_{j=1}^{d-1} L_p^e(j) + \sum_{j=d+1}^{N_d} L_{p-1}^e(j) + L_p^e(d) + L_p^i \\ &= \sum_{j=1}^{d} L_p^e(j) + \sum_{j=d+1}^{N_d} L_{p-1}^e(j) + L_p^i. \end{aligned} \quad (12)$$

Explicitly, only two component encoders are used in practical turbo encoders. Therefore, by substituting $N_d = 2$ into Equations 9 and 12, we can determine the a *posteriori* LLR of the equaliser and decoders, whereas Equations 8 and 10 can be used to determine the corresponding a *priori* inputs of the equaliser and decoders.

## 4 Simulation Results

All simulation results were obtained using BPSK modulation over a five-path dispersive Rayleigh fading channel, using a normalised Doppler frequency of $f_d = 1.5 \times 10^{-4}$. In our investigations, the fading magnitude and phase was assumed to be constant for the duration of a transmission burst, a condition which we refer to as employing burst-invariant fading. The CIR coefficients of the band-limited and symbol-spaced channel were $f_0 = \sqrt{0.45}$, $f_1 = \sqrt{0.25}$, $f_2 = \sqrt{0.15}$, $f_3 = \sqrt{0.1}$ and $f_4 = \sqrt{0.05}$. Both the turbo equaliser and the channel decoders employed the Log-MAP algorithm [7], in order to calculate the soft-outputs of the bits. Note that the term **turbo decoding iteration** refers to a decoding iteration, where information is passed iteratively between the component decoders, but not between the decoders and the equaliser. Information is only passed iteratively between the equaliser and decoder(s), when **turbo equalisation** is employed.

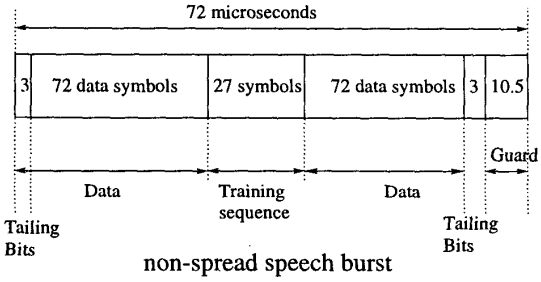


Figure 4: Transmission burst structure of the FMA1 non-spread speech burst of the FRAMES proposal [8]

The transmission burst structure used in this system is the FMA1 non-spread speech burst as specified in the Pan-European FRAMES proposal [8], which is shown in Figure 4. In our forthcoming discourse the effects of different number of iterations are shown. Specifically, Figure 5 portrayed our performance comparison between the

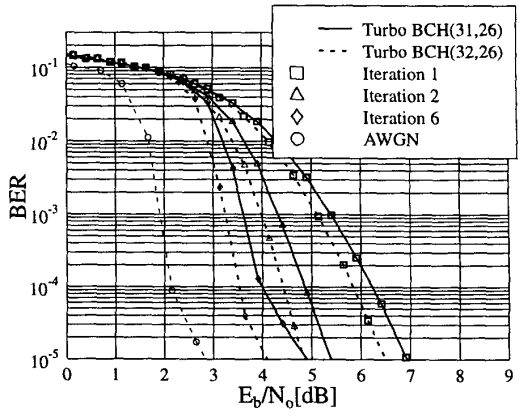


Figure 5: Turbo decoding performance comparison between the **turbo** BCH(31,26) and turbo BCH(32,26) codes for different number of **turbo decoding** iterations using BPSK modulation over five-path symbol-spaced Rayleigh fading channels using burst-invariant fading.

turbo decoding of the conventional BCH(31,26) and extended BCH(32,26) codes after one, two and six turbo decoding iterations. Both systems employed a random turbo interleaver depth of 14,976 bits. The random channel interleaver depth utilised was 20,736 and 21,888 bits for the turbo BCH(31,26) and turbo BCH(32,26) codes, respectively. As we can see from this figure, the performance of the extended turbo BCH(32,26) code was always better, than that of the turbo BCH(31,26) scheme, irrespective of the number of iterations. We achieved a higher performance gain, when using the extended turbo BCH(32,26) code as compared to the conventional turbo BCH(31,26) scheme employing six turbo decoding iterations. Explicitly, the performance of the extended turbo BCH(32,26) code was approximately 0.8 dB better, than that of the conventional turbo BCH(31,26) at a BER of $10^{-5}$. For comparison, the performance of the extended turbo BCH(32,26) code after 6 turbo decoding iterations over a non-dispersive Gaussian channel without inter-symbol interference was also shown in Figure 5. Beyond six turbo decoding and turbo equalisation iterations no significant performance gain was attained.

Figure 6 showed the turbo equalisation performance comparison between the extended turbo BCH(32,26) code and the conventional turbo BCH(31,26) scheme for different number of turbo equalisation iterations. Random turbo interleavers possessing depths of 14,976 bits were employed by both systems. Similarly, random channel interleavers having depths of 19,800 bits and 20,900 bits for the turbo BCH(31,26) and turbo BCH(32,26) codes were employed, respectively. The performance of the turbo BCH(32,26) codes was again observed to be superior to that of the turbo BCH(31,26) codes by approximately 1 dB after 6 turbo equalisation iterations at BER= $10^{-5}$. Figure 6 also showed that the maximum performance gain was achieved by employing turbo equalisation in conjunction with turbo BCH codes, as compared to turbo decoding in Figure 5. Furthermore, a

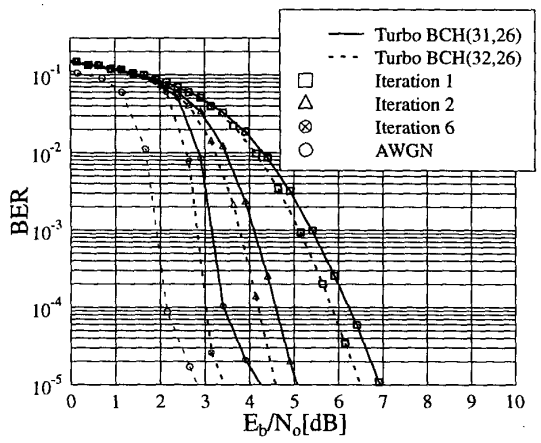


Figure 6: Performance comparison between the **turbo** BCH(31,26) and BCH(32,26) codes for different number of **turbo equalisation** iterations using BPSK modulation over five-path symbol-spaced Rayleigh fading channel using burst-invariant fading.

higher iteration gain – defined as the gain in $E_b/N_0$ performance terms with respect to the first iteration – can be achieved by this scheme as compared to the schemes characterised Figure 5 and 7. The performance of the turbo BCH(32,26) code over non-dispersive AWGN channels was also shown in Figure 6. We also observed that the turbo equaliser was capable of removing the channel-induced inter-symbol interference more effectively, than the turbo decoder of Figure 5.

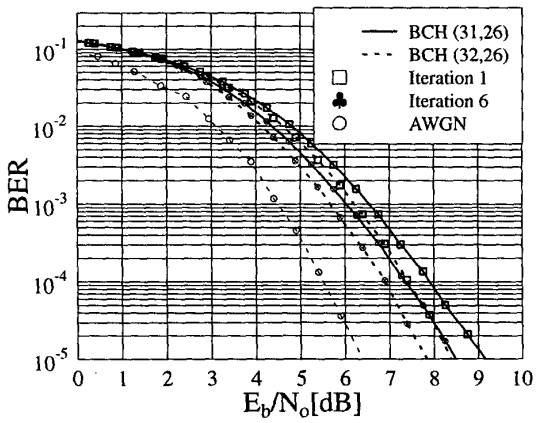


Figure 7: Turbo equalisation performance comparison between the conventional **non-turbo** BCH(31,26) and the extended BCH(32,26) codes for different number of turbo equalisation iterations using BPSK modulation over five-path symbol-spaced Rayleigh fading channel using burst-invariant fading.

Figure 7 portrays our performance comparisons between the conventional **non-turbo** BCH(31,26) code and the extended BCH(32,26) code for different number of turbo equalisation iterations. Both systems, namely the BCH(31,26)-coded BPSK system and the BCH(32,26)-coded BPSK scheme, employed random channel interleavers having interleaving depths of 17,856 bits and 18,432 bits, respectively. At a BER of $10^{-5}$, the performance of the BCH(32,26) code was about 0.5 dB better, than that of the BCH(31,26) scheme after one and six turbo equalisation iterations. However, the turbo equalisation iteration gain of these **non-turbo** coded schemes was lower in terms of $E_b/N_0$ as compared to the **turbo** coded schemes of Figure 6. This was due to the fact that the channel equaliser exploits the inherent channel diversity and hence achieves a good performance even after the first turbo equalisation iteration. Furthermore, the performance obtained approximated the BER achieved for the non-dispersive AWGN channel without inter-symbol interference, which was the curve marked with the circles in Figure 6. This curve was also equivalent to the performance of Viterbi decoding the BCH(32,26) codes in the non-dispersive AWGN channel.

## 5 Conclusion

In conclusion, we have proposed a modified MAP algorithm for extended BCH codes, in order to incorporate the parity check bit into calculating the soft-outputs of the coded bits. The modified MAP algorithm was then used to improve both turbo decoding and turbo equalisation. It was shown that the turbo decoding and turbo equalisation performance of the extended BCH(32,26) code was superior to that of the conventional BCH(31,26) code by approximately 0.5 – 1.0 dB at BER$= 10^{-5}$.

## References

[1] Shu Lin and Daniel J. Costello, Jr, "Error Control Coding: Fundamentals and Applications", *Prentice-Hall*, 1983

[2] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate", *IEEE Transactions on Information Theory*, vol. 20, pp. 284-287, March 1974

[3] C.Douillard, A.Picart, M.Jézéquel, et al, "Iterative correction of intersymbol interference: Turbo-equalization," *European Transactions on Communications*, vol. 6, pp. 507–511, September-October 1995.

[4] Michael J.Gertsman and John H.Lodge, "Symbol-by-symbol MAP demodulation of CPM and PSK signals on Rayleigh flat-fading channels," *IEEE Transactions on Communications*, vol. 45, pp. 788–799, July 1997.

[5] D.Raphaeli and Y.Zarai, "Combined turbo equalization and turbo decoding," *IEEE Communications Letter*, vol. 2, pp. 107–109, April 1998.

[6] L.R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal Decoding of Linear Codes for Minimising Symbol Error Rate," *IEEE Transactions on Information Theory*, pp. 284–287, March 1974.

[7] Patrick Robertson, Emmanuelle Villebrun and Peter Hoeher, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain," *Proceedings of the International Conference on Communications*, pp. 1009–1013, June 1995.

[8] A.Klein, R.Pirhonen, J.Sköld and R.Suoranta, "FRAMES Multiple Access Model - Wideband TDMA with and without spreading," *Proceedings of PIMRC'97*, pp. 37–41, 1997.