A Log-MOP Algorithm for Decoding Turbo Codes

Yi-Nan Lin[*1], Wei-Wen Hung[*2]

[*]*Department of Electronic Engineering, Mingchi University of Tech., Taipei Hsien, Taiwan*

[1]jnlin@ccsun.mit.edu.tw
[2]wwhung@ns1.mit.edu.tw

Abstract— **In this paper, a new decoding algorithm is proposed for turbo decoding. This decoding algorithm based on maximum log ratio of *a-observation* probability (Log-MOP) to substitute maximum log ratio of *a-posteriori* probability (Log-MAP). Theoretically and experimentally, the proposed a new decoding scheme is shown no coding gain performance degradation compare to Log-MAP algorithm, *i.e.*, it can be achieved as well as coding gain performance of the Log-MAP algorithm.**

Keywords— **Turbo codes, maximum a-observation probability (MOP) decoding, iterative decoding.**

1. INTRODUCTION

For many applications, the component code to be used with iterative decoding is a recursive systematic convolutional code (RSC code). Turbo codes is a kind of iterative decoding first proposed by Berrou, Glavieux and Thitimajashima, who reported excellent coding gain results [1, 2], approaching the theoretical limit predicted by Shannon. Two suitable decoders of turbo coding are Soft-Output Viterbi algorithm (SOVA) proposed by Hagenauer and Hoeher [3] and Maximum *a-posteriori* (MAP) algorithm proposed by Bahl et al [4]. MAP algorithm has also become known as the BCJR algorithm, named after its inventors. Turbo codes have some applications mainly in modern communication systems, such as: CDMA (Code Division Multiple Access) and PCS (Personal Communication Service) [5], the third generation mobile system UMTS (Universal Mobile Telephone System) [6], and the Third Generation Partnership Project, etc.

The effectiveness of the turbo decoding scheme is based on iterating the maximum *a-posteriori* probability (MAP) algorithm, applied to each constituent code. In general, the MAP algorithm is implemented by means of a soft-in soft-out (SISO) decoder. This SISO decoder computers the *a-posteriori* probability (APP), i.e., a reliability value, for each received information symbol. However, this computation is extremely complex owing to the multiplications and exponential operations required for the forward and backward recursions in the trellis diagram. In order to reduce the decoding complexity of the MAP algorithm, researchers have developed other SISO decoders, which are less complex and can be used instead of the MAP algorithm. Two of such algorithms are the Max-Log-MAP [9] and the Log-MAP algorithms [8].

In this paper, we try to explore the feasibility of maximum the log of the ratio of the *a-observation* probabilities of the bit taking its two possible values for turbo decoding. Theoretically, the proposed a new decoding scheme can achieve similar performance, i.e., it can be achieved as well as BER performance of Log-MAP algorithm. The organization of the paper is as follow: We first define some notations and review the modified algorithm, Max-Log-MAP algorithm in Section II. The subsequent sections then describe proposed a new SISO Algorithm for decoding turbo codes in Section III. Comparisons the maximum *a-posteriori* probability and a maximum *a-observation* probability describe in Section IV. Finally, conclusions are made in Section V.

2. TURBO CODES

In this section, we review the encoding and decoding schemes of the convolutional turbo codes [7-11].

2.1. Turbo Encoding

From now on, we will study the behaviour of decoding of convolutional codes (and in particular recursive systematic convolutional (RSC) codes), we will choose a notation that complies with such an encoder, for example one with two memory elements as shown in Fig. 1 [1].

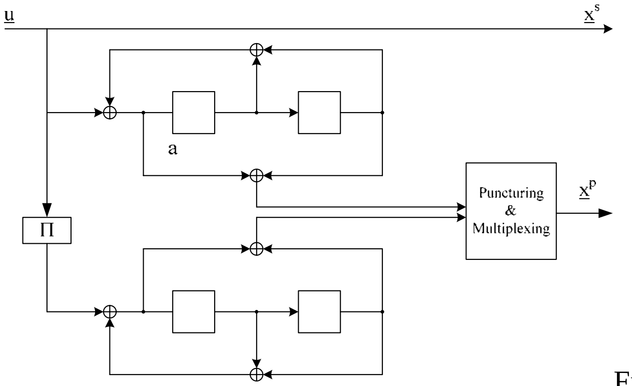

Fig. 1. Two identical recursive systematic convolutional encoders employed in a turbo coding scheme.

We represent the binary information sequence by $\underline{u}=(u_1,\ ...,\ u_N)$. Assume that the encoder has M memory elements. In the example, it has two outputs, one is the sequence of systematic information bits $\underline{x}^s=(x_1^s,\ ...,\ x_N^s)=\underline{u}$. The other is the parity information sequence $\underline{x}^p=(x_1^p,\ ...,\ x_N^p)$, with $x_k^p=\sum_{i=0}^{M} g_i^f a_{k-i}$, where $\underline{g}^f=(g_0^f,\ ...,\ g_M^f)$ is the feed-forward generator and $a_k=u_k\oplus\sum_{i=1}^{M} g_i^b a_{k-i}$; Similarly, $\underline{g}^b=(g_0^b,\ ...,\ g_M^b)$ is the feed-back generator of the encoder. The RSC encoder of this conventional convolutional encoder is represented as $\underline{G}=(g^f,\ g^b)$, where $g^f,\ g^b$ being the decimal representation of $\underline{g}^f$ and $\underline{g}^b$.

2.2. MAP Algorithm for Turbo Decoding

Let us consider Fig. 2 showing the MAP decoder trellis for the generator of the code is designed (7, 5) in Fig. 1. We assume that the transmitted bit $x_k=\pm1$ is sent over AWGN channel with one-sided noise power spectral density N_0 using BPSK modulation. The received sequence $\underline{y}=(\underline{y}^s,\ \underline{y}^p)$ will refer to the sequence of pairs of received systematic and parity symbols. The received sequence $\underline{y}$ can be split up into three sections: the received codeword associated with the present transition $\underline{y}_k$, the received the received sequence priori to the present transition $\underline{y}_{j<k}$, and the received sequence after the present transition $\underline{y}_{j>k}$.

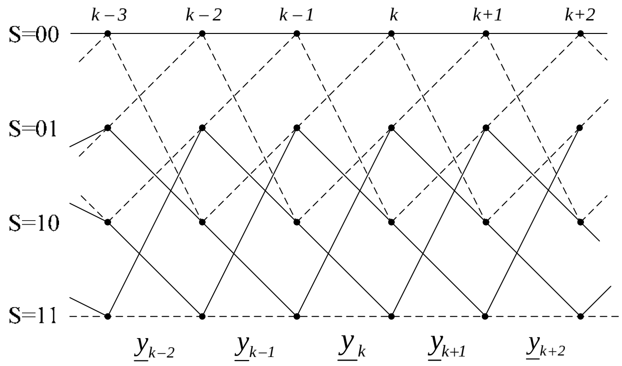

Fig. 2. MAP decoder trellis for designed (7, 5) RSC code.

The Max-Log-MAP algorithm simplifies the MAP algorithm using the approximation:

$$\ln\sum_i e^{X_i}\simeq\max_i\{X_i\}\qquad(1)$$

where $\max_i\{x_i\}$ means the maximum value of x_i. Let the state of the encoder at time $k-1$ be $S_{k-1}=s'$ and at time k be $S_k=s$. Goal of the Max-Log-MAP algorithm is to find *a-posteriori* Log Likelihood Ratio LLR $L(u_k\mid\underline{y})$ as following:

$$L(u_k\mid\underline{y})\equiv\ln\left(\frac{P(u_k=+1\mid\underline{y})}{P(u_k=-1\mid\underline{y})}\right)$$

$$=\ln\left(\frac{\sum_{\substack{(s',\,s)\to\\ u_k=+1}}\exp[A_{k-1}(s')+\Gamma_k(s',\ s)+B_k(s)]}{\sum_{\substack{(s',\,s)\to\\ u_k=-1}}\exp[A_{k-1}(s')+\Gamma_k(s',\ s)+B_k(s)]}\right),\qquad(2)$$

where $(s',\ s)\to u_k=\pm1$ is the set of transitions from the previous state $S_{k-1}=s'$ to the present state $S_k=s$ that can be occur if the input bit $u_k=\pm1$. And the forward and backward recursion term $A_k(s)$, $B_{k-1}(s')$ can be expressed use the Equation (1), respectively:

$$A_k(s)=\ln\left(\sum_{all\ s'}\exp[A_{k-1}(s')+\Gamma_k(s',\ s)]\right)$$

$$\approx\max_{s'}(A_{k-1}(s')+\Gamma_k(s',\ s)),\qquad(3)$$

$$B_{k-1}(s')=\ln\left(\sum_{all\ s}\exp[B_k(s')+\Gamma_k(s',\ s)]\right)\qquad(4)$$

The branch metric term are given by:

$$\Gamma_k(s',\ s)=C+\frac{1}{2}u_kL(u_k)+\frac{L_c}{2}\sum_{l=1}^{n}y_{kl}x_{kl}$$

$$=C+\frac{1}{2}u_kL(u_k)+\frac{L_c}{2}y_{k1}x_{k1}+\frac{L_c}{2}\sum_{l=2}^{n}y_{kl}x_{kl},\ (5)$$

where $L_c=2E_b/\sigma^2$ with E_b is the transmitted energy per bit and σ^2 is the noise variance. Bit x_{kl} and y_{kl} are the individual bits within the transmitted and received codeword $\underline{x}_k$ and $\underline{y}_k$. Bit y_{k1} is the received version of the transmitted systematic bit $x_{k1} = u_k$.

Finally, from the Equation (1), we can write for the *a-posteriori* LLR $L(u_k \mid \underline{y})$ which the Max-Log-MAP algorithm calculates:

$$L(u_k \mid \underline{y}) \equiv \ln\left(\frac{P(u_k = +1 \mid \underline{y})}{P(u_k = -1 \mid \underline{y})}\right)$$

$$= \ln\left(\frac{\sum_{\substack{(s',s)\to\\ u_k=+1}} \exp[A_{k-1}(s') + \Gamma_k(s', s) + B_k(s)]}{\sum_{\substack{(s',s)\to\\ u_k=-1}} \exp[A_{k-1}(s') + \Gamma_k(s', s) + B_k(s)]}\right)$$

$$\approx \max_{\substack{(s',s)\to\\ u_k=+1}} (A_{k-1}(s') + \Gamma_k(s', s) + B_k(s)) - \max_{\substack{(s',s)\to\\ u_k=-1}} (A_{k-1}(s') + \Gamma_k(s', s) + B_k(s)), \quad (6)$$

it means that in the Max-Log-MAP algorithm for each bit u_k the *a-posteriori* LLR $L(u_k \mid \underline{y})$ is calculated by considering every transition from the trellis stage S_{k-1} to the stage S_k. These transitions are grouped into those that might have occurred if $u_k = +1$, and those that might have occurred if $u_k = -1$.

3. A New Siso Algorithm For Decoding Turbo Codes

In this section, we concentrate on our proposed algorithm that is derived by *a-observation* LLR $L(\underline{y} \mid u_k)$ as following:

$$L(\underline{y} \mid u_k) = \ln\left(\frac{p(\underline{y} \mid u_k = +1)}{p(\underline{y} \mid u_k = -1)}\right)$$

$$= \ln\left(\frac{\sum_{x \in C^{+k}} p(\underline{y} \mid \underline{x})}{\sum_{x \in C^{-k}} p(\underline{y} \mid \underline{x})}\right), \quad (7)$$

where $C^{\pm k} \triangleq \{\underline{x} \mid \underline{x} \in C,\ x_{k1} = \pm 1\}$, means $C^{\pm k}$ is a set which sequence $\underline{x}$ has systematic bit $x_{k1} = \pm 1$.

Consider sending information over AWGN channel using BPSK modulation, we can obtain:

$$p(y_{li} \mid x_{li}) = (2\pi\sigma^2)^{\frac{-1}{2}} \exp\left(-\frac{E_b (y_{li} - x_{li})^2}{2\sigma^2}\right) \quad (8)$$

and hence:

$$p(\underline{y}_l \mid \underline{x}_l) = \prod_{i=1}^{n} p(y_{li} \mid x_{li})$$

$$= (2\pi\sigma^2)^{\frac{-n}{2}} \exp\left(-\frac{E_b}{2\sigma^2}\sum_{i=1}^{n}({y_{li}}^2 - 2y_{li}x_{li} + {x_{li}}^2)\right)$$

$$= (2\pi\sigma^2)^{\frac{-n}{2}} \exp\left(-\frac{E_b}{2\sigma^2}\sum_{i=1}^{n}({y_{li}}^2 + 1)\right) \cdot \exp\left(\frac{E_b}{\sigma^2}\sum_{i=1}^{n}(y_{li}x_{li})\right)$$

$$= \Omega \cdot \exp\left(\frac{E_b}{\sigma^2}\sum_{i=1}^{n}(y_{li}x_{li})\right), \quad (9)$$

where $\Omega = (2\pi\sigma^2)^{\frac{-n}{2}} \exp\left(-\frac{E_b}{2\sigma^2}\sum_{i=1}^{n}({y_{li}}^2 + 1)\right)$

The probability $p(\underline{y} \mid \underline{x})$ must be:

$$p(\underline{y} \mid \underline{x}) = \prod_{l=1}^{N} p(\underline{y}_l \mid \underline{x}_l) \quad (10)$$

Define the branch term $\Gamma_k(s', s)$ as:

$$\Gamma_k(s', s) \equiv \ln(p(\underline{y}_k \mid \underline{x}_k))$$

$$= \ln\left(\Omega \cdot \exp\left(\frac{E_b}{\sigma^2}\sum_{i=1}^{n}(y_{li}x_{li})\right)\right)$$

$$= \Omega^{-1} + \frac{E_b}{\sigma^2}\sum_{i=1}^{n}(y_{li}x_{li})$$

$$= \Omega^{-1} + \frac{L_c}{2}\sum_{i=1}^{n}(y_{li}x_{li}) \quad (11)$$

Define the forward recursion term $A_k(s)$ as:

$$A_k(s) \equiv \ln\left(\sum_{x_{j\le k}(s)} p(\underline{y}_{j\le k} \mid \underline{x}_{j\le k}(s))\right)$$

$$= \ln\left(\sum_{s'}\left(p(\underline{y}_{j\le k-1} \mid \underline{x}_{j\le k-1}(s')) \cdot p(\underline{y}_k \mid \underline{x}_k(s', s))\right)\right)$$

$$= \underset{s'}{Jaco.}\{A_{k-1}(s') + \Gamma_k(s', s)\}, \quad (12)$$

with $A_0(S = 00) = 0$ and $A_0(S \ne 00) = -\infty$. And define the backward recursive term $B_{k-1}(s')$ as:

$$B_{k-1}(s') \equiv \ln\left(\sum_{x_{j\ge k}(s')} p(\underline{y}_{j\ge k} \mid \underline{x}_{j\ge k}(s'))\right)$$

$$= \underset{s}{Jaco.}\{B_k(s) + \Gamma_k(s', s)\}, \quad (13)$$

with $B_N(S = 00) = 0$ and $B_N(S \ne 00) = -\infty$.;

where using the Jacobian logarithm [8, 9]:

$$\underset{i=1,\ 2}{Jaco.}\{x_i\} = \ln\sum_{i=1}^{2}\exp(x_i)$$

$$= \max(x_1,\ x_2) + \ln(1 + \exp(-\mid x_1 - x_2 \mid)) \quad (14)$$

Substituting Equations (11), (12), (13) into Equation (7), we obtain:

$$L(\underline{y}|u_k)=\ln\left(\frac{\sum_{x\in C^{+k}} p(\underline{y}|\underline{x})}{\sum_{x\in C^{-k}} p(\underline{y}|\underline{x})}\right)$$

$$=\ln\left(\frac{\sum_{(s',s)\to+1} p(\underline{y}_{j<k}|\underline{x}_{j<k}(s'))\cdot p(\underline{y}_k|\underline{x}_k(s',s))\cdot p(\underline{y}_{j>k}|\underline{x}_{j>k}(s))}{\sum_{(s',s)\to-1} p(\underline{y}_{j<k}|\underline{x}_{j<k}(s'))\cdot p(\underline{y}_k|\underline{x}_k(s',s))\cdot p(\underline{y}_{j>k}|\underline{x}_{j>k}(s))}\right)$$

$$=\ln\left(\frac{\sum_{(s',s)\to+1} \exp(A_{k-1}(s'))\cdot\exp(\Gamma_k(s',s))\cdot\exp(B_k(s))}{\sum_{(s',s)\to-1} \exp(A_{k-1}(s'))\cdot\exp(\Gamma_k(s',s))\cdot\exp(B_k(s))}\right)$$

$$=\underset{(s',s)\to+1}{Jaco.}\{A_{k-1}(s')+\Gamma_k(s',s)+B_k(s)\}$$

$$-\underset{(s',s)\to-1}{Jaco.}\{A_{k-1}(s')+\Gamma_k(s',s)+B_k(s)\} \qquad (15)$$

For iterative decoding:

$$L(\underline{y}|u_k)=\ln\left(\frac{\sum_{x\in C^{+k}} p(\underline{y}|\underline{x})}{\sum_{x\in C^{-k}} p(\underline{y}|\underline{x})}\right)$$

$$=\frac{2E_b}{\sigma^2}y_{k1}+\Delta_k, \qquad (16)$$

then, $\Delta_k = L(\underline{y}|u_k)-\frac{2E_b}{\sigma^2}y_{ks}$.

In the i th iteration, the first SISO decoder receives the channel sequence $(\frac{2E_b}{\sigma^2}\cdot y_{k1}, \frac{2E_b}{\sigma^2}\cdot y_{kp1})$ from the first encoder and the *a-priori* information $\Delta^i_{a,1}(u_k)$ provided by de-interleaving the extrinsic information $\Delta^{i-1}_{e,2}(u_k)$ of the second SISO decoder in the $(i-1)$ th iteration, and hence it can produce an improved *a-observation* information $L^i_1(\underline{y}|u_k)$. Next, the second SISO decoder comes into operation. It uses the interleaved channel sequence $(\frac{2E_b}{\sigma^2}\cdot \tilde{y}_{k1}, \frac{2E_b}{\sigma^2}\cdot y_{kp2})$ from the second encoder and the *a-priori* information $\Delta^i_{a,2}(u_k)$ derived by interleaving the extrinsic information $\Delta^i_{e,1}(u_k)$ of the first SISO decoder to calculate the *a-posteriori* information $L^i_2(\underline{y}|u_k)$. Above iteration process continues, and on average the BER of the decoded bits decreases as the number of decoding iteration increases. It is shown as follows:

$$\Delta^i_{e,1}(u_k)=L^i_1(\underline{y}|u_k)-\left(\frac{2E_b}{\sigma^2}y_{k1}+\Delta^{i-1}_{a,2}(u_k)\right) \qquad (17)$$

$$\Delta^i_{e,2}(u_k)=L^i_2(\underline{y}|u_k)-\left(\frac{2E_b}{\sigma^2}\tilde{y}_{k1}+\Delta^i_{a,2}(u_k)\right) \qquad (18)$$

The iterative process is implemented by setting

$$\Delta^i_{a,1}(u_k)=\pi^{-1}\left[\Delta^{i-1}_{e,2}(u_k)\right] \qquad (19)$$

$$\Delta^i_{a,2}(u_k)=\pi\left[\Delta^i_{e,1}(u_k)\right] \qquad (20)$$

where, π and π^{-1} denote the interleaving and de-interleaving operations, respectively.

The block diagram of our proposed scheme based on an iterative MOP decoder is shown in Figure 3.

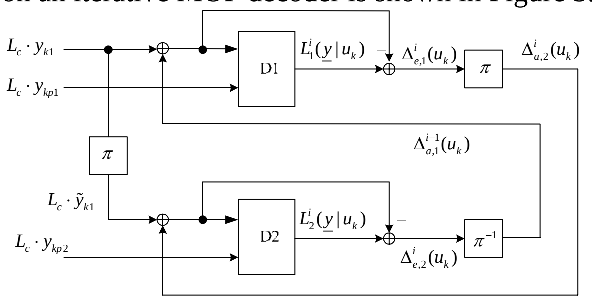

Fig. 3. Block diagram of a proposed iterative SISO turbo decoding.

4. SIMULATION RESULTS

In this section, we conduct a series of experiments to evaluate the effectiveness of the MOP algorithm compare to the MAP algorithm. In the simulation, a data block of 1000 bits ($N=1000$) are considered and 1000 data blocks are transmitted. Two kinds of encoding schemes are under investigation. One is "two 4-state (memory size $M=2$) RSC constituent encoder with generator polynomials (7, 5)" and another is "two 8-state (memory size $M=3$) RSC constituent encoders with generator polynomials (13, 15)." The encoders are linked together by a pseudo random interleaver. The overall code rate is $1/3$. The coded bits are modulated using binary phase shift keying (BPSK) and white Gaussian noise with a double-sided power spectral density of $N_0/2$ is added to the modulated signal. At the decoder, only eight iterations are carried out.

The results of the simulations are shown in Figs. 4-5. Figures 4 and 5 show the BER plotted versus E_b/N_0 and the encoding parameters are $M=2$, $M=3$, respectively.

Apparently, the proposed Log-MOP algorithm for turbo decoding compares with the Log-MAP, there is no any difference in coding gain performance.

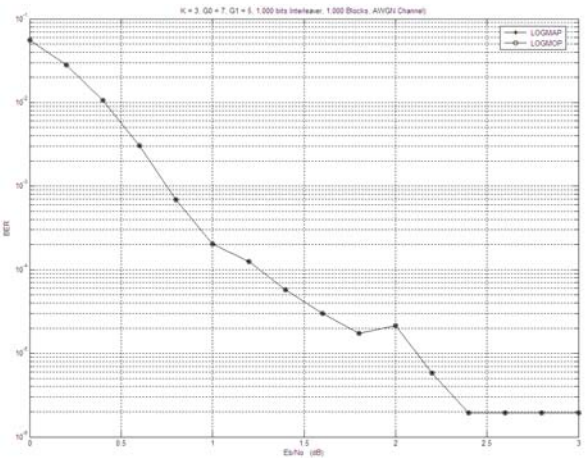

Fig. 4. BER versus E_b / N_0 for a LogMAP and a LogMOP decoder (encoding parameters $M = 2$, $N = 1000$).

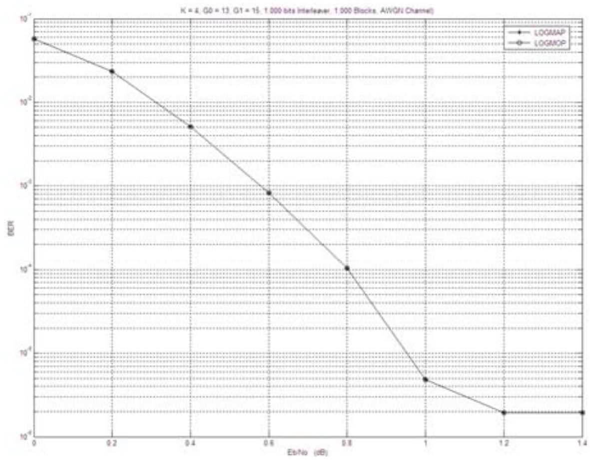

Fig. 5. BER versus E_b / N_0 for a LogMAP and a LogMOP decoder (encoding parameters $M = 3$, $N = 1000$).

5. CONCLUSIONS

Our proposed decoding scheme is based on maximum *a-observation* probability (MOP) information. It is a new computing method for an effective SISO iterative algorithm for decoding turbo codes based on MOP information. Theoretically, the proposed new decoding scheme is no coding gain sacrificing when compared to the Log-MAP decoding algorithm, *i.e.*, it can be achieved as well as BER performance of Log-MAP algorithm.

REFERENCES

[1] C. Berrou, A. Glavieux and P. Thitmajshima, "Near Shannon limit error-correction coding and decoding: Turbo codes," *Proceedings of 1993 International Conference on Communications*, pp. 1064-1090, Geneva, Switzerland, May 1993.

[2] C. Berrou and A. Glavieux, "Near Shannon limit error-correction coding and decoding: Turbo codes," *IEEE Transactions on Communications*, Vol. 44, pp. 1261-1271, 1996.

[3] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," *in Conf. Proceedings IEEE Globecom '89*, pp. 1680-1686, 1989.

[4] 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. IT-20, pp. 284-287, 1974.

[5] D. Divsalar and F. Pollara, "Turbo codes for PCS applications," in *Proc. ICC'95* (Seattle WA, June 1995).

[6] Y. Rosmansyah, "Channel and decoding effects on UMTS turbo code performance," *CCSR internal report*, University of Surrey, U.K., August 2001.

[7] L. Hanzo, T. H. Liew and B. L. Yeap, "Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Fading Channels," John Wiley & Sons, 2002.

[8] J. A. Erfanian, S. Pasupathy, and G. Gulak, "Reduced complexity symbol detectors with parallel structures for isi channel," *IEEE Trans. Commun.*, vol. 42, 99.1661-1671, February/March/April 1994.

[9] W. Koch and A. Baier, "Optimum and sub-optimum detection of coded data disturbed by time-varying intersymbol interference," in *Proc. GLOBECOM '90*, pp. 1679-1684, December 1990.

[10] P. Robertson, "Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes," *IEEE Globecom*, pp. 1298-1303, 1994.

[11] P. Robertson, E. Villebrun and P. Hoeher, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain," in *IEEE Int. Conf. on Communications* (Seattle, WA, June 1995), pp.1009-1013.