A Modified Turbo-Detector for long delay spread channels

B. Penther*, D. Castelain * and H. Kubo**

*Mitsubishi Electric ITE, 80 avenue des Buttes de Coësmes, 35700 RENNES, FRANCE.
** Mitsubishi Electric ITC, 5-1-1 Ofuna, Kamakura-City, Kanagawa, 247-8501, JAPAN.
Tel : (+33) 2 99 84 21 16, Fax : (+33) 2 99 84 21 15
E-mail: penther@tcl.ite.mee.com

Abstract: *This paper presents a turbo-detector scheme. The detection part of the receiver is based on a variant of the APP equalizer. This variant employs reduced-state and per survivor techniques to enable a moderate complexity even when the frequency-selective channel memory length gets large. Thanks to the iterative process, this sub-optimal APP equalizer concatenated with a decoder can achieve good performance even for severe propagation conditions. Therefore this scheme is a possible candidate for future mobile radio communications with high bit rate transmission.*

Keywords: modified turbo-detector, modified *APP* algorithm, reduced state, Per Survivor Processing, List type equalizer.

1. INTRODUCTION

Since the first introduction of the turbo-equalization concept in [2], two trends have emerged: the turbo-detection [2] based on *APP* equalizer and the turbo-equalization [3] based on Interference Canceller. We propose here to employ an alternate solution using a novel variant of reduced-state *APP*, called List-type *APP* equalizer with soft output. This novel "turbo-equalizer" (or modified turbo-detector) proves to perform well for frequency selective channels with long delay profile. As a special case it can be considered as a "Turbo-DFSE" and in more general as a "Turbo-List Viterbi Equalizer".

2. TRANSMISSION SYSTEM

Lets us consider the transmission system described in Figure 1, where the information data d_k are first convolutionally encoded, interleaved, modulated then transmitted over a frequency-selective channel. The channel output (received symbols affected by ISI) can be expressed as:

$$R_k = \sum_{i=0}^{L-1} c_i D_{k-i} + \eta_k \qquad (1)$$

where c_i are the channel coefficients, L is the constraint length of the channel, D_{k-i} is a M-ary modulated symbol and η_n is the sampled AWGN. The receiver employs a iterative equalization/decoding scheme known as turbo-equalizer/detector [2,3].

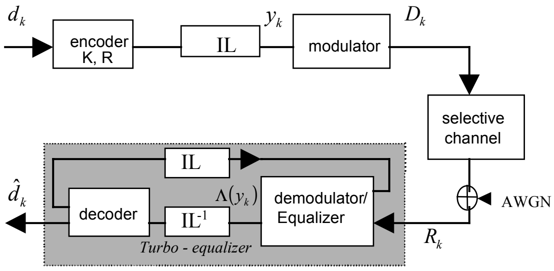

Figure 1: Transmission system

As explained in [4], it is important to employ a powerful equalization scheme at the first iteration of the turbo-equalizer. However, the complexity of the optimum *APP* detector grows exponentially with the channel memory. Therefore we propose to employ a novel variant of the *APP* algorithm for the equalizer, called *List-type APP* equalizer which realizes a good trade-off between performance and complexity. Indeed, this scheme provides *soft-output* while employing a *reduced trellis* with M^{J-1} *states* with J being the reduced constraint length of the channel $(J<L)$, whereas the remaining part of the ISI is cancelled by an internal per-survivor processing with a *list* of N survivors, which is an arbitrary integer. This scheme is used in all iterations of a turbo-equalizer to realize a "*modified turbo-detector"*. Part 3 of this paper will introduce this new equalization algorithm and how it is incorporated in the turbo-detection scheme. Part 4 will provide simulations results and Part 5 conclusions.

3. THE MODIFIED TURBO-DETECTOR

3.1. Reduced Complexity APP Equalizer

The List-type approach originally proposed for *hard-output* Viterbi algorithms in [5,6,7] has been investigated here, with respect to the MAP algorithm with *soft-output* (also called *APP*). Indeed, as explained in [7], in the case of a List-type variant of the MLSE, the trellis shall have M^{J-1} states where J is the reduced constraint length of the channel ($J<L$), while the number N of survivors for each state can be set to an arbitrary value. Therefore the overall complexity of the scheme is proportional to $N.M^{J-1}$. In the packet-oriented MAP described in [8], the soft-outputs are obtained after the calculation of *forward* $\alpha_k(m)$ and *backward* $\beta_k(m)$ state metrics, computed recursively for each state m and each time k. Similarly here, we will compute recursively *forward* and *backward* state metrics, but *forward* state metrics $\alpha^{Sn}{}_k(m)$ will now also depend on the Survivors S_n ($n = 1, 2,\ldots N$).

With $J<L$, each state m of the trellis is composed of the $J-1$ most recent symbols. Besides "Survivors" depending on each state will be used to compute the branch metric for the reduced-state trellis as shown in (2) (in the case of squared Euclidian metric):

$$\gamma(R_k, m, m') \approx -\left| R_k - \sum_{i=0}^{J-1} \tilde{y}_{k-i} c_i - \sum_{i=J}^{L-1} \hat{y}_{k-i} c_i \right|^2 / 2\sigma^2 \quad (2)$$

where $\tilde{y}_{k-i}$ are the elements of the trellis transition from state m to state m', $\hat{y}_{k-i}$ are the past-estimates belonging to the "Survivors" (per-survivor processing) and σ^2 is the noise variance.

In the case of *List-type* algorithm, several survivors (a "list" of N) can be associated to each state and the computation is done as follows: Let us assume that at time $k-1$, a state m is associated to a forward state metric list $\alpha^{Sn}{}_{k-1}(m)$ and a Survivor list $S_{n,k-1}$. At next time k, $N.M$ paths (the modulation is *M-ary)* converge to the following state m' and their path metrics are defined as: $\alpha^{Sn}{}_{k-1}(m).\gamma^{Sn}(R_{k-1},m,m')$, where $\gamma^{Sn}(R_{k-1},m,m')$ is the branch metric between states m and m', for the Survivor $S_{n,k-1}$ (see(2)). These $N.M$ path metrics are sorted by order. The N best paths will then define, for the state m' at time k, the Survivors $S_{n',k}$ list and their associated forward state metrics list $\alpha^{Sn'}{}_k(m')$ $(0\leq n'<N)$. Each state metric $\alpha^{Sn'}{}_k(m')$ is a combination (*e.g.* sum for the MAP version or max for the Max Log MAP version of this scheme) of all the path metrics leading to m' which are inferior or equal to the path metric associated to $S_{n',k}$.

As regards to the *backward* state metrics $\beta_k(m)$, their calculation shall be based on the Survivors S_n obtained during the forward recursion and for a given state m at time k, the backward state metric $\beta_k(m)$ is equal to the combination (*e.g.* sum for the MAP) of all the $N.M$ backward path metrics leading to m.

Finally the LLR of coded bits y_k will be obtained (in the case $M=2$) by the relation (3).

$$\Lambda(y_k) = Log \frac{\sum_{m=1}^{2^{J-1}} \sum_{j=0}^{N-1} \alpha_k^{Sj}(m) \gamma^{Sj}(R_k, m, S_f^1(m)) \beta_{k+1}(S_f^1(m))}{\sum_{m=1}^{2^{J-1}} \sum_{j=0}^{N-1} \alpha_k^{Sj}(m) \gamma^{Sj}(R_k, m, S_f^0(m)) \beta_{k+1}(S_f^0(m))} \quad (3)$$

where $S^i{}_f(m)$ is the state following the state m if the input symbol is i. We can notice that the LLR in (3) combines the state metrics related to all Survivors ("Survivor Combining").

A simplification of this algorithm is possible with the omission of the *backward* state metrics in (3) to save storage complexity and which results in small performance degradation. In this case, a decision delay corresponding to the reduced constraint length J is introduced in (4) (case $M=2$):

$$\Lambda(y_{k-J+1}) = Log \frac{\sum_{m=1}^{2^{J-1}} \sum_{j=0}^{N-1} \alpha_k^{Sj}(S_b^1(m)) \gamma^{Sj}(R_k, S_b^1(m), m)}{\sum_{m=1}^{2^{J-1}} \sum_{j=0}^{N-1} \alpha_k^{Sj}(S_b^0(m)) \gamma^{Sj}(R_k, S_b^0(m), m)} \quad (4)$$

where $S^i{}_b(m)$ is the state which leads to the state m if the output symbol is i.

Even if this work was done independently, it is worth noting that in case of omission of the backward state metric and if N=1, this equalizer is equivalent to the RS-SDVE (Reduced State–Soft Decision Viterbi Equalizer) described in [9]. Besides, we shall also mention the LOVA (List Output Viterbi Algorithm) with Likelihood Post-Processing which generate soft-values presented in [10]. However, our scheme is not a "List Output" since the likelihood processing (i.e. generation of soft values) is internal to the algorithm and therefore no post-processing is required which makes it simple to implement.

In the case of N=1, this soft-output scheme is hence equivalent to a "soft-DFSE" (Decision Feedback Sequence Estimation [6]) and if N>1 it acts like a "soft-LVE" (List Viterbi Equalizer [7]). As explained in [7], reduced-state algorithm performance highly depends on the channel profile. For instance, the DFSE will perform better on Line Of Sight channel (LOS) whereas a LVE with similar complexity will perform better on Non Line Of Sight channels (NLOS). In the same way the soft-output scheme presented here will have similar behavior and we will show in the section 4 that for a severe NLOS channels, the choice of N>1 is profitable.

3.2. List-type reduced-state APP based Turbo-Detector

The modified *APP* equalizer described above is used in all iterations of a turbo-detector [11] scheme to realize a "*modified turbo-detector"*. Figure 2 describes the turbo-detector principle beyond the second iteration. We can notice now that there are in fact two "loops" in this receiver: one is a "large" loop due to the feedback of the extrinsic information from the decoder to the equalizer (and vice versa) and is the basis of the "turbo-effect". The second "loop" is internal to the equalizer and is related to the per survivor processing (PSP) which introduces a decision-directed behavior.

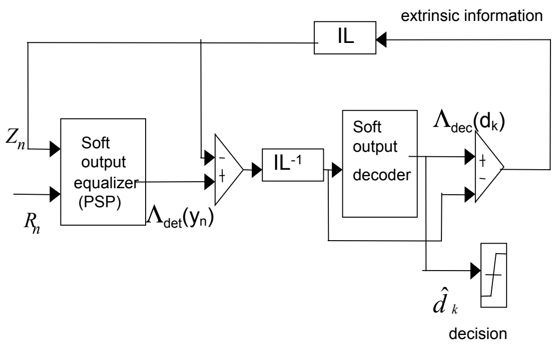

Figure 2: Modified turbo-detector

This point is illustrated by the formula of the branch metric (5) in the equalizer for iterations > 1. The branch metric now depends on 3 different components, two of them being decision-directed ($\hat{y}_{k-i}$ and Z_k) :

$$\left| R_k - \sum_{i=0}^{J-1} \tilde{y}_{k-i} c_i - \sum_{i=J}^{L-1} \hat{y}_{k-i} c_i \right|^2 + \gamma \cdot \left| Z_k - \tilde{y}_k \right|^2 \qquad (5)$$

where Z_k is the extrinsic value at time k and γ is a weighting coefficient which depends on the iteration number and can be typically $\gamma = \sigma^2/\sigma^2_z$, σ^2_z being the variance of the extrinsic information.

One can wonder whereas these two "loops" can operate jointly without risk of error-propagation. As it will be shown in the simulation results, it seems that the turbo-effect tends to compensate the sub-optimality of the reduced-state detector. That is to say that Z_k will tend to compensate erroneous tentative decision of $\hat{y}_{k-i}$ and vice versa. However, to guarantee the trigger of this effect, the List-type detector shall achieve a satisfactory performance even at the first iteration, which can be done thanks to an adequate choice of the parameters, J and N, taking into account carefully the channel profile and its severity. Consequently, in the case of N=1, this scheme realizes a kind of "Turbo-DFSE" (suitable for LOS channels) and if N>1, it realizes a "Turbo-LVE" (List Viterbi Equalizer, suitable for NLOS channels).

At last, to summarize the principle of this scheme, we can say that, first we have decreased *exponentially* the complexity of the *APP* equalizer by using a reduced channel memory length of J<L. Then, via the use of iterative process (turbo-effect) we have increased "*linearly"* the complexity of the receiver (including the decoder part) in order to compensate the loss of performance due to state-reduction. Fortunately, as in the case of turbo-codes, the performance improvement due to the iterative process is more than linear. And the modified turbo-detector which is a *disjoint iterative receiver* with simple (sub-optimal) components shall be compared first to *the optimum non-iterative disjoint receiver* [4] (namely the first iteration of a conventional turbo-detector). In some cases our scheme can even be compared with a conventional turbo-detector (which bound is the *optimum non-iterative joint receiver* [4]) as illustrated in the next section in Figure 3.

4. SIMULATIONS RESULTS

BER performance has been obtained by computer simulation, for a system with a rate ½ convolutional code (CC) with constraint length *5* and generators (*23,35*). The interleaver is 64x64 pseudo-random matrix. The modulation is *QPSK* (*M=4).* The receiver employs a *List-type APP* equalizer with *J=3* (*16* states trellis) and *N=2* and with *no backward* recursion, associated in the iterative process with a CC Log MAP decoder. Two type of frequency selective channels have been considered:

- *Channel 1*: *Proakis C, L*=5 taps
$\vec{c}_i$ = (0.227; 0.460; 0.688; 0.460; 0.227)

- *Channel 2*: *L*=10 taps, with equal power
$\vec{c}_i = (1/\sqrt{10}\,;1/\sqrt{10}\,;1/\sqrt{10}\,;1/\sqrt{10}\,;1/\sqrt{10}\,;$
$1/\sqrt{10}\,;1/\sqrt{10}\,;1/\sqrt{10}\,;1/\sqrt{10}\,;1/\sqrt{10}\,)$

The channel impulse response (CIR) is supposed to be perfectly known at the receiver side. For each of these two channels, the first 3 taps are processed by the trellis transitions (J=3) of the modified detector whereas the remaining taps are processed by per survivor processing with N=2 survivors at each state.

Figure 3 shows the performance on *channel 1* (*Proakis C* channel) and compare it to the *optimum non-iterative disjoint* receiver (concatenation of an *APP* equalizer with 256 state trellis and same decoder). We can see that after 3 iterations, the simplified turbo detector (including *16* states trellis equalizer) outperforms this optimum disjoint receiver.

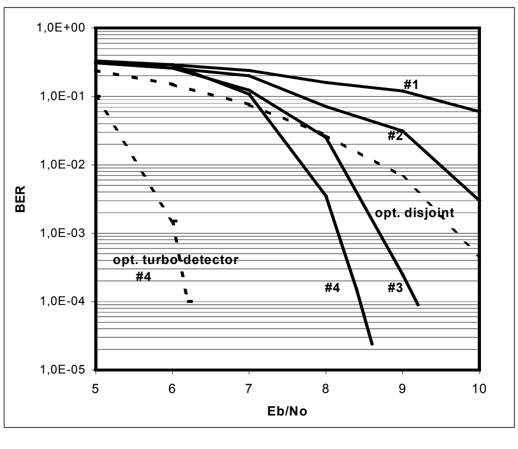

Figure 3: Performance on *channel 1* (*Proakis C*)

Of course the performance of our scheme remains inferior to that of the optimal turbo-detector with 256 state trellis (iteration #4) also shown in Figure 3.

Figure 4 shows the performance on *channel 2* (*L=10,* each path with equal power). It was not possible to simulate on this channel the optimum *disjoint* receiver because the *APP* equalizer part (*J=L=10*) would require a trellis with $4^9 = 262{,}144$ states. On the contrary, the modified turbo-detector with (*J*=3, 16 states) offers a reasonable complexity. Furthermore, it is worth noting that this channel is quite severe because each path has the same power ($1/\sqrt{10}$), thus it is a good test for evaluating the performance of the modified turbo-detector.

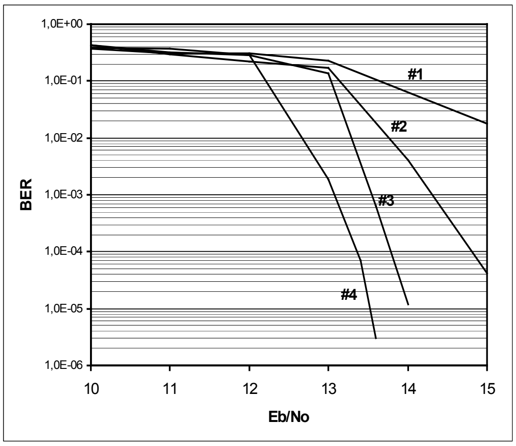

Figure 4: *channel 2* (*L*=10, equally-distributed)

We can notice that at iteration #1 the BER performance remains relatively high. However, after few iterations (#4), thanks to a large turbo-effect, low BER can be achieved. Hence on this long delay spread channel, the modified turbo-detector is a good solution: its complexity is moderate and the performance improvement due to the iterative process is large.

5. CONCLUSION AND FUTURE WORK

A *modified turbo-detector* based on a List-type variant of the *APP* detector, employing reduced-state trellis and per survivor technique has been presented. This scheme can be employed on frequency selective channels with long delay profile (where the optimum *APP* complexity is prohibitive). Adequate choice of the parameters *J* and *N* enables to trigger the turbo-effect even for severe frequency selective channels, where it achieves very good performance after few iterations. Therefore this scheme is a possible candidate for future mobile radio communications with high bit rate transmission.

Further work will consider the influence of imperfect CIR estimation impact on the performance of this modified turbo-detector and possible enhancement with turbo channel-estimation.

ACKNOWLEDGEMENT

The authors would like to thank D. Mottier, A. Guéguen and P. Didier from Mitsubishi Electric for their helpful advices and comments.

REFERENCES

[1] C. Berrou, A. Glavieux, P. Thitimajshima, “Near Shannon limit error correcting coding and decoding: Turbo codes”, *IEEE ICC’93,* vol. 2/3,pp.1064-1070*,* Geneva, May 1993.

[2] C. Douillard, M. Jézéquel, C. Berrou, A. Picard, P. Didier and A. Glavieux, “Iterative correction of intersymbol-interference: Turbo-equalization”, *Europ. Trans. on Telecom.,* vol. 6, pp. 507-511, Sept-Oct 1995.

[3] A. Glavieux, C. Laot, J. Labat, “Turbo-equalization over a frequency selective channel”, *International Symposium on Turbo Codes,* Brest, France 1997.

[4] A. Roumy, I. Fijalkow, D. Pirez, “Joint Equalization and Decoding: why choose the iterative solution?”, *International VTC fall Conference,* Amsterdam, pp 2989-2993, Sept. 1999.

[5] T. Hashimoto, “A list-type reduced-constraint generalization of the Viterbi algorithm*”, IEEE Trans. Inform. Theory*, vol. IT-33, 6, pp. 866-876, Nov 1987.

[6] A. Duel-Hallen and C. Heegard : “Delayed decision-feedback sequence estimation”, *IEEE Trans. Commun*., vol. COM-37, 5, pp. 428-436, May 1989.

[7] H. Kubo *et al.*,” A List-output Viterbi Equalizer with two kinds of metric criteria” *in Proc., IEEE International Conference on Universal Personal Comm. 1998 (ICUPC)*, pp 1209-1213.

[8] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate", *IEEE Trans. Inform. Theory,* vol. IT-20, pp. 284-287, March 1974.

[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.

[10] P. Hoeher, "Advances in soft-output decoding," in *Proc. IEEE GLOBECOM* '93, Houston, Texas, pp.793-797, Nov./Dec.1993.

[11] P. Didier “la Turbo-Egalisation et son application aux communications radio-mobiles”, *PhD* report, December 1996.

[12] W. Feng and B. Vucetic “a List Bidirectional Soft Output Decoder of Turbo-Codes”, *International Symposium on Turbo Codes,* Brest, France 1997.