PARTIAL ITERATIVE EQUALIZATION AND CHANNEL DECODING

Jinhong Wu and Branimir Vojcic

The George Washington University
Department of Electrical and Computer Engineering
Email: johnnywu@gwu.edu, vojcic@gwu.edu

ABSTRACT

This paper proposes a bit-selective iterative processing scheme for turbo equalization. We consider convolutionally coded and turbo coded transmissions over inter-symbol interference (ISI) channels. The receiver starts with conventional iterative channel equalization and decoding. After each iteration, bit convergence status is measured via cross-entropy. Short windows are then applied around unconverged bits and the subsequent detection/decoding iterates only within the windows. The rapid decrease in selected bits after each such iteration reduces computational complexity significantly, but performance is well maintained by utilizing the Markov property in the trellis detection/decoding.

Index Terms— partial iteration, turbo equalization, turbo decoding, cross-entropy criterion.

1. INTRODUCTION

The "turbo equalization" approach [3] [7] has shown strong capability in mitigating ISI incurred by frequency selective channels. In the original setting, the information data were encoded with a convolutional code. To save more in transmission power, stronger error control codes such as turbo codes are applied and the receiver performs turbo equalization with turbo decoding [8] [6] [2] [10]. While these approaches have excellent performance, the required computational complexities are often prohibitive when many iterations are needed.

To reduce equalization complexity, trellis states reduction method such as delayed decision feedback sequence estimation (DDFSE) [4] has been utilized. It feeds back hard decisions on detected inputs to the last channel memory units, thus reduces the trellis states to represent only the unknown channel memory units. The complexity reduction is roughly proportionate to the states reduction. However, DDFSE generally suffers from performance degradation due to error propagation. Furthermore, it does not reduce the complexity in channel decoding. An effective joint complexity reduction method is introduced in [1]. It is an extension from the well known turbo decoding cross-entropy stop criterion [5] where iteration stop is based on frame detection convergence.

In this paper, we consider another approach based on bit convergence status. We extend the approach in [11] for turbo decoding to turbo equalization. Complexity reduction is realized by limiting the bits involved in each iteration. For convolutional codes, we measure cross-entropy between equalizer and decoder's a posteriori distribution for each coded bit, while for turbo codes we measure the cross-entropy for systematic bits at the output of component decoders, in order to avoid slow convergence by the equalizer.

The rest of this paper is organized as follows. In section II we specify the system models. We consider binary convolutional codes and turbo codes at the transmitter, and specify the corresponding receiver structures. Section III elaborates on the bit-selection based processing algorithm. Section IV shows the performance and corresponding complexity reductions by simulations. Finally, we conclude in section V.

2. SYSTEM MODELS

For simplicity, we consider binary phase shift keying (BPSK) modulation for coded data frames. We restrict discussions to ISI channels that can be modeled by an equivalent discrete-time finite impulse response (FIR) filter so that MAP detection is feasible. Denote the information bit sequence as $\{u(k), k = 1, 2, \cdots, N\}$, the rate $1/v_c$ coded bits as $\{c_j(k), k = 1, 2, \cdots, N\, j = 1, 2, \cdots, v_c\}$, the BPSK modulated symbols $\{x(n), n = 1, 2, \cdots, Nv_c\}$, and the ISI channel impulse response as $\{h(m), m = 0, 1, \cdots, M\}$. The received signal after demodulation is represented by

$$y(n) = \sum_{m=0}^{M} h(m)x(n-m) + z(n) \tag{1}$$

where $z(n)$ is the additive white Guassian noise (AWGN) with distribution $N(0, \sigma^2)$.

We consider the Log-MAP algorithm [9] at both the equalizer and the decoder(s). The equalizer computes log-likelihood ratios (LLR) for each transmitted bit by [6]

$$L_{Eq}(n) = L(x(n) \mid \mathbf{y})$$
$$= \ln \frac{\sum_{(S',S)\Rightarrow +1} \exp(\mathbf{A}_{n-1}(S') + \Gamma_n(S',S) + \mathbf{B}_n(S))}{\sum_{(S',S)\Rightarrow -1} \exp(\mathbf{A}_{n-1}(S') + \Gamma_n(S',S) + \mathbf{B}_n(S))}. \tag{2}$$

The recursive computations for $\mathbf{A}_n(S)$, and $\mathbf{B}_n(S)$ are

$$\mathbf{A}_n(S) = \ln(\sum_{allS'} \exp(\mathbf{A}_{n-1}(S') + \Gamma_n(S',S))) \tag{3}$$

$$\mathbf{B}_n(S') = \ln(\sum_{allS} \exp(\mathbf{B}_{n+1}(S) + \Gamma_n(S',S))). \tag{4}$$

The branch transition metric $\Gamma_n(S', S)$ is defined as

$$\Gamma_n(S',S) = -\frac{1}{2\sigma^2}(y(n) - \sum_{m=0}^{M} h(m)\hat{x}(n-m))^2 + L^a_{Eq}(n) \tag{5}$$

with $\hat{x}(n-m)$'s being the input to the ISI channel that can cause the state transition from S' to S, and $L^a_{Eq}(n)$ the a priori LLR about $x(n)$. Furthermore, $L_{Eq}(n)$ is decomposed into

$$L_{Eq}(n) = L^a_{Eq}(n) + L^e_{Eq}(n) \tag{6}$$

where $L^e_{Eq}(n)$ is referred to as the extrinsic information.

At the decoder, computations are largely in parallel with (2) - (5). A major difference is that the transition branch metric for the d-th systematic bit is [6],

$$\Gamma_d(S',S) = \sum_{j=1}^{v_c} \frac{1}{2} L^e_j(d) c_j(d) + L^a_1(d) \tag{7}$$

where $L^e_j(d)$ is extrinsic information for coded bit $c_j(d)$ produced by the equalizer, and $L^a_1(d)$ being the a priori LLR on the d-th systematic bit from the other component decoder.

3. PARTIAL ITERATIVE PROCESSING FOR TURBO EQUALIZATION AND DECODING

The Markov property in (3) - (4), and the memoryless characteristic in (5) and (7) for branch metric calculation imply that if converged bits can be identified, then the Log-MAP detection/decoding within windows bounded by converged symbols can produce largely the same results as those by complete iterations. The boundary soft information for each window can be borrowed from previous iterations, to start the shortened trellis detection. On the other hand, as in general most bits in a frame converge fast after a few iterations, selective computation focusing on the uncoverged bits can be very cost-effective. Extensive experiments have shown that a sufficiently low cross-entropy value well indicates the bit convergence while erroneous bits are mostly among those with high

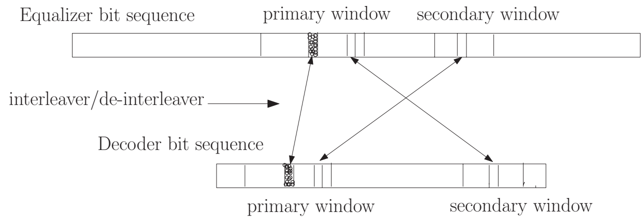

Fig. 1. Bit sequences and windows deployments for receivers with a single decoder and an equalzier.

cross-entropy values. This suggests that we may select bits with high cross-entropy and iterate detection/decoding among a few converged bits around them.

Consider the first case where an equalizer is followed by a single convolutional decoder, the cross-entropy on coded bits can be measured by [1]

$$T^i_{Eq}(n) \approx \frac{|L^{a,i}_{Eq}(n) - L^{a,i-1}_{Eq}(n)|^2}{\exp(|L^i_{Eq}(n)|)} \tag{8}$$

where $L^i_{Eq}(n)$ is the a posteriori LLR.

After the i-th iteration, bits that satisfy $T^i(n) < T^1(n)10^{-3}$ are classified as converged, and otherwise as unconverged. A short window is then applied around each unconverged bit, both at the equalizer and at the decoder. These windows are defined as primary windows. We denote the primary window lengths at equalizer and decoder as $2N^1_{Eq} + 1$ and $2N^1_{De} + 1$, respectively. Primary windows ensure that each unconverged bit receives updated extrinsic information in future iterations. For each neighboring, converged bit in the primary windows at the equalizer, its interleaved position at the decoder is located and a secondary window of length $2N^2_{De} + 1$ is applied. Vice versa, a secondary window of length $2N^2_{Eq} + 1$ is applied at the equalizer for each bit that after de-interleaving resides in a primary window at the decoder. The employment of secondary windows ensures that all bits in the primary windows can receive updated extrinsic information, thus help detect the unreliable bit at the center of each primary window. After this process, the next equalization and decoding iteration is restricted within the windowed pieces. The boundary $\mathbf{A}_n(S)$ and $\mathbf{B}_n(S)$ values are inherited from the last iteration to start the trellis computation. The updated $L^a_{Eq}(n)$'s are only exchanged among the windowed bits. This procedure is depicted in Fig. 1. Note that in this process windows often overlap which reduces the overall amount of selected bits. Furthermore, the original long trellis is replaced by some short ones, which can help reduce detection/decoding delays by parallel implementations.

For receivers with turbo decoding, we measure cross-entropy at the two decoders' output to utilize the faster decoding convergence. In place of (8), we adopt [5]

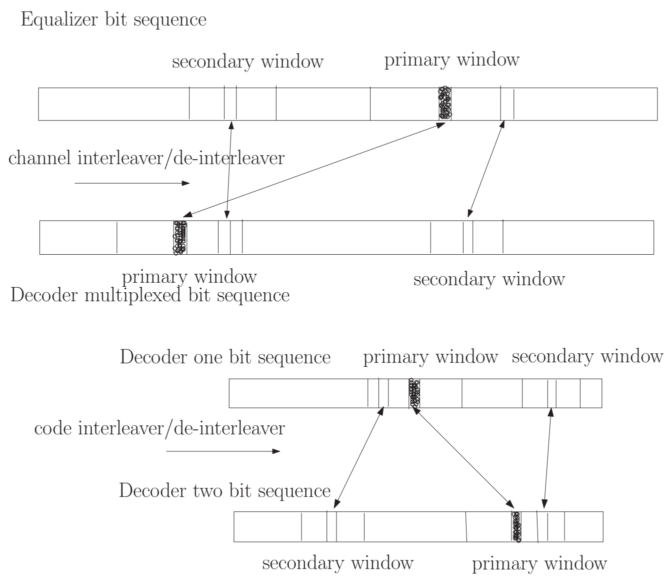

Fig. 2. Bit sequences and windows deployments for receivers with two component decoders and an equalzier.

$$T^i_{De}(d) \approx \frac{|L^{a,i}_{De,1}(d) - L^{a,i-1}_{De,1}(d)|^2}{\exp\left(|L^i_{De,1}(d)|\right)} \tag{9}$$

where $L^i_{De,1}(d)|)$ is the a posteriori LLR for the d-th systematic bit at the first decoder, and $L^{a,i}_{De,1}(d)$ the a priori LLR.

The bit convergence criterion is $T^i_{De,1}(d) < T^1_{De,1}(d)10^{-3}$. For unconverged bits, we apply primary and secondary windows with lengths $2N^1_{De}+1$ and $2N^2_{De}+1$ at both decoders. After each decoding iteration, the code bits from the two decoders are multiplexed and interleaved to form the input to the equalizer, with windowing process specifying which bits are to be further equalized. This procedure is illustrated in Fig. 2.

Considering the additional complexity incurred by bit selection, we notice that (8) or (9) requires only 1 addition, 2 multiplications, and 1 $\exp()$ function. Note that here the $\exp()$ function can be implemented using either a piece-wise linear approximation or a look-up table. Overall, the extra complexity is negligible compared with trellis detection.

4. SIMULATIONS

We test the algorithms with a static ISI channel. The FIR's coefficients are $[\sqrt{0.45}, \sqrt{0.25}, \sqrt{0.15}, \sqrt{0.1}, \sqrt{0.05}]$. In the first case, a single recursive systematic convolutional (RSC) encoder with the generator polynomials $(37, 21)$ is employed. The 2048-bit, half rate code is randomly interleaved before modulation. The frame error rates (FER) are compared. Here we denote C_{Eq} as the required complexity for a single full equalization iteration, and C^{cc}_{De} for a single full convolutional decoding iteration. The extra overhead by convergence measurements and symbol selection are omitted(8) or (9). We set $2N^1_{Eq}+1 = 2N^1_{De}+1 = 15$ and $2N^2_{Eq}+1 = 2N^2_{De}+1 = 3$. The maximum number of conventional complete iterations is 12, while for partial iterations it is limited to 18. Fig. 3 and Table 1 compare the results. The proposed partial iterations and the block stop method in [1] are similar in complexity and performance, but provide different flexibilities. However, the two methods can be combined such that if the block stop method results in certain performance loss, partial iterations can continue to detect the very few unconverged bits.

Table 1. Complexity by partial iterations. An equalzier with a convolutional decoder. Channel interleaver length 2048.

E_bN_o(dB)	4.5	5	5.5	6
Partial	$4.20C_{Eq}$	$3.66C_{Eq}$	$3.20C_{Eq}$	$3.03C_{Eq}$
iterations	$4.53C^{cc}_{De}$	$3.97C^{cc}_{De}$	$3.47C^{cc}_{De}$	$3.11C^{cc}_{De}$
Block Stop	$3.93C_{Eq}$	$3.32C_{Eq}$	$3.05C_{Eq}$	$3.00C_{Eq}$
iterations	$3.93C^{cc}_{De}$	$3.32C^{cc}_{De}$	$3.05C^{cc}_{De}$	$3.00C^{cc}_{De}$

In the second case, the turbo code employs two identical RSC $(7, 5)$ encoders to produce a puctured, half rate code. A random interleaver of length 2048 is also applied before modulation. For equalizer, only primary windows are applied. We denote the complexity by a single complete turbo decoding iteration as C^{tc}_{De}. The results are shown in Fig. 4 and Table 2.

5. CONCLUSIONS AND DISCUSSIONS

We propose a partial iterative equalization and channel decoding scheme. For receivers with an equalizer and a single convolutional decoder, bit convergence is measured for all coded bits. When the equalization is followed by turbo decoding, we measure convergence status for systematic bits at the two component decoders' output. Unconverged bits are applied with short windows and future iterations are restricted within the windows. The rapid decreases in selected bits after each iteration proportionately reduce the computational load. As the performance is well maintained, the approach can be categorized as a redundancy reduction method rather than a complexity-performance trade-off.

From the detection/decoding on graphs point of view, the proposed algorithm only passes selected extrinsic information. The information flows are completely stopped when no bits are selected. It can be easily verified that if part of the information flows become unchanged, then computations based on these input become redundant. From this perspective, the introduced windowing mechanism may be only one example of reducing the redundancy. Further simplified detection algorithms are expected in future work.

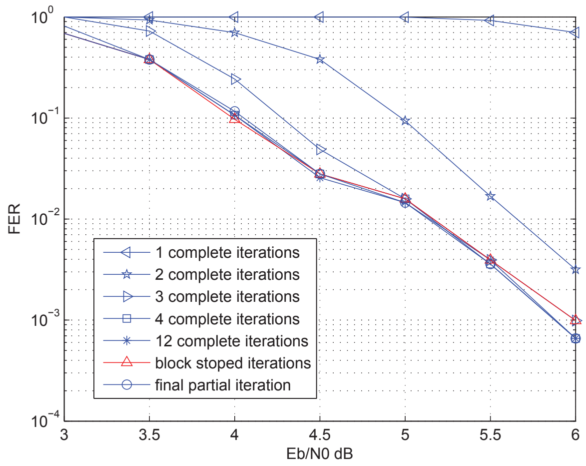

Fig. 3. FER comparison for case I: An equalizer with a convolutional decoder. Channel interleaver length 2048.

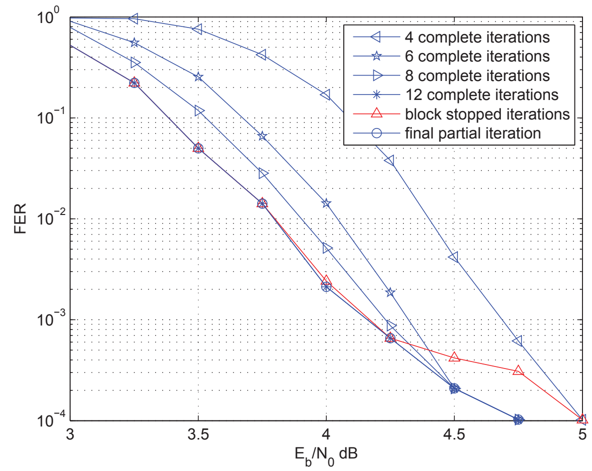

Fig. 4. FER comparison for case II: An equalizer with two component decoders. Code interleaver length 1024. Channel interleaver length 2048.

Table 2. Complexity by partial iterations. An equalzier with two component decoders. Code interleaver length 1024. Channel interleaver length 2048.

E_bN_o(dB)	3.5	4	4.5	5
Partial iterations	$7.26C_{Eq}$ $7.37C_{De}^{tc}$	$5.12C_{Eq}$ $5.22C_{De}^{tc}$	$4.12C_{Eq}$ $4.19C_{De}^{tc}$	$3.46C_{Eq}$ $3.58C_{De}^{tc}$
Block Stop iterations	$7.10C_{Eq}$ $7.10C_{De}^{tc}$	$4.97C_{Eq}$ $4.97C_{De}^{tc}$	$4.01C_{Eq}$ $4.01C_{De}^{tc}$	$3.31C_{Eq}$ $3.31C_{De}^{tc}$

6. REFERENCES

[1] G. Bauch, H. Khorram, and J. Hagenauer, "Iterative equalization and decoding in mobile communications systems," in *Proc. European Personal Mobile Commun. Conf.*, Bonn, Germany, Sept./Oct. 1997.

[2] A. O. Berthet, B. S. Unal, and R. Visoz, "Iterative decoding of convolutionally encoded signals overmultipath rayleigh fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 1729–1743, Sept. 2001.

[3] C. Douillard, A. Picart, P. Didier, M. Jzquel, C. Berrou, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo equalization," *European Transactions on Telecommunications*, vol. 6, pp. 507–511, Sept. 1995.

[4] A. Duel and C. Heegard, "Delayed decision-feedback sequence estimation," *IEEE Transactions on Communications*, vol. 37, pp. 428–436, May 1989.

[5] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429–445, Feb. 1996.

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

[7] J. G. Proakis, *Digital Communications*, McGraw-Hill, 4th edition, 2000.

[8] D. Raphaeli and Y. Zarai, "Combined turbo equalization and turbo decoding," *IEEE communications Letters*, vol. 2, pp. 107–109, Apr. 1998.

[9] P. Robertson, P. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain," in *Proc. of IEEE Intl. Conf. on Com.*, Seattle, Washington, pp. 1009–1013, June 1995.

[10] E. M. Sozer, J. G. Proakis, and F. Blackmon, "Iterative equalization and decoding techniques for shallow wateracoustic channels," in *Proc. of MTS/IEEE Oceans '01 Conference and Exhibition*, Honolulu, HI, USA, vol. 4, pp. 2201–2208, Nov. 2001.

[11] J. Wu, B. Vojcic, and Z. Wang, "Turbo decoding complexity reduction by symbol selection and partial iterations," in *Proc. of IEEE GLOBECOM*, Washington, DC, USA, pp. 3910–3914, Nov. 2007.