Reduced Complexity APP for Turbo Equalization

Dan Raphaeli and Ami Saguy
Electrical Engineering-Systems Dept.
Tel Aviv University
Tel Aviv 69978 Isreal
Email: danr@eng.tau.ac.il, ami@mns-israel.com

***Abstract*— This paper investigates the subject of reducing the complexity of turbo equalization in which a receiver combines the equalization and decoding process in an iterative fashion. We show that it is possible to approximate the *a posteriori* probability (APP) module by two time varying linear transversal filters and a simple non-linear memoryless processor, and thus replace the exponential complexity of the APP module with quadratic complexity in the filters length. Further complexity reduction leads to linear complexity, for which the transversal filters are constant during the block. For this structure, which is similar to a previously suggested scheme, we calculate the optimized parameters and increase performance. The derivations cover the cases of BPSK and QAM modulations. Simulation results are presented for parallel-concatenated turbo code with BPSK and QAM modulation over channels that introduce severe amplitude distortion.**

I. Introduction

Application of the turbo principle [2] to the equalization process referred to as Turbo equalization was first proposed in [5]. In [6], the convolutional code used in [5] was replaced by a turbo code [1]. The combined turbo code and equalization yields performance that is superior to independent decoding and equalization. The best performance in turbo equalization is achieved when the equalizer is implemented as an APP module in its basic form [3] and [4]. However, the complexity of the APP module grows exponentially with the channel length and number of bits/symbol. In order to save in complexity, the APP equalizer module was replaced in [7] by two linear transversal filters. The first filter receives the channel output and attempts to maximize the signal to noise (SNR) at its output while the second filter receives an estimate of the symbols calculated from the extrinsic information supplied by the channel decoder and attempts to cancel the symbol interference at the output of the first filter. In this paper, we reach the same two filter structure but we derive it by analytic considerations rather than ad-hoc. Starting from the APP equalizer module, we show that if the intersymbol interference (ISI) is approximated by a random variable having a Gaussian distribution, then the APP module is reduced to two linear transversal filters, one for the input and one for the estimate of the symbols calculated from the extrinsic information. The sum of the two filters output is followed by a simple non-linear memoryless processor that computes the APP of each bit composing the symbol. The derived equations further imply that the tap gains of each filter change for every symbol entering the filters as a function of the extrinsic information. To further reduce complexity, for each iteration, we can replace the sequence of symbol variances that are calculated from the extrinsic information with their average value and thus obtain a solution of fixed tap gains.

The paper is organized as follows. After the introduction the system model is presented in Section II. In Section III we approximate the ISI by a Gaussian variable and derive the equalizer structure. Simulation results are provided in section IV and conclusions are drawn in section V.

II. System Model

The system model is illustrated in Fig. 1. The transmitter consists of a source that generates binary i.i.d. information bits $\{u_k\}$ taking on the values 0 or 1 with equal probability. The bits are supplied to a turbo channel encoder formed by two parallel concatenated recursive systematic convolutional (RSC) encoders separated by a random interleaver. The resulting coded bits $\{c_n\}$ are interleaved and every m-bits are associated with a complex constellation symbol $\{s_n\}$ by the modulator.

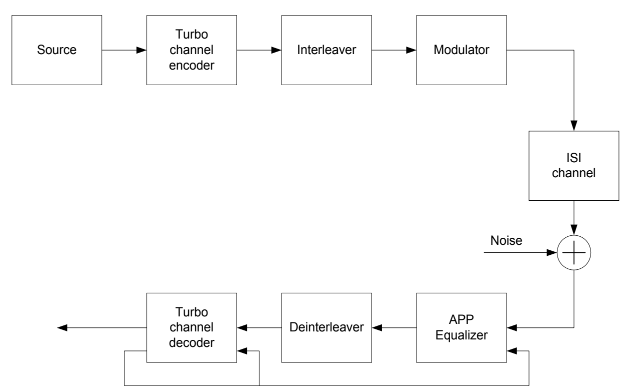

Fig. 1. System Model.

The symbols are transmitted over an equivalent discrete time white noise filter channel [8] having L+1 coefficients $\{h_l\}$ and further corrupted by additive white Gaussian noise (AWGN) $\{w_n\}$ having zero mean and variance $N_0/2$ (where N_0 is the power spectral density (PSD) of the AWGN at the output of the

0-7803-7400-2/02/$17.00 (C) 2002 IEEE

continuous channel). In this case, $\{h_l\}$ represents the combined responses of the transmitter pulse shape filter, channel response, receiver matched filter, sampler and discrete-time noise whitening filter. The received signal $\{r_n\}$ is expressed by

$$r_n = \sum_{l=0}^{L} h_l s_{n-l} + w_n \tag{1}$$

The receiver consists of an APP equalizer module followed by a block deinterleaver and a parallel concatenated turbo channel decoder. The extrinsic information $\{\lambda_n\}$ supplied by the turbo decoder is fed-back to the APP equalizer module and the process is iterated in both the equalizer and channel code to increase performance.

III. Gaussian Approximation

The received signal 1 can be reformulated as follows

$$r_n = h_0 s_n + \sum_{l=1}^{L} h_l s_{n-l} + w_n \tag{2}$$

and the second term on the right can be identified as the ISI. We approximate the ISI, which is a weighted sum of several independent symbols by a random variable having a Gaussian distribution (although the central limit theorem cannot be argued to support this approximation even for infinite ISI coefficients assuming $\sum_l |h_l|^2 < \infty$).

We now proceed to calculate the APP or the log likelihood ratio (LLR) of each code bit $c_{n,k}$ (k=0,1,..., m-1) composing the symbol s_n given a segment observation interval of $N = N_1 + N_2 + 1$ channel symbols $\mathbf{r}_n$

$$\mathbf{r}_n = [r_{n-N_1}, r_{n-N_1+1}, \ldots, r_{n-1}, r_n, r_{n+1}, \ldots, r_{n+N_2}] \tag{3}$$

and

$$\mathbf{r}_n = \mathbf{s}_n \mathbf{H}_c + \mathbf{w}_n \tag{4}$$

Here $\mathbf{H}_c$ is the $(N+L) \times N$ convolution matrix of a channel having a $1 \times N$ $\mathbf{h} = [0, \ldots, 0, h_0, h_1, \ldots, h_L, 0, \ldots, 0]$ coefficient vector, $\mathbf{s}_n$ is the $1 \times (N+L)$ transmitted symbols, and $\mathbf{w}_n$ is the $1 \times N$ additive Gaussian noise vector

$$E(\mathbf{w}_n) = 0, \quad \frac{1}{2} E(\mathbf{w}_n^H \mathbf{w}_n) = \frac{N_0}{2}\mathbf{I} \tag{5}$$

where $\mathbf{I}$ is the identity matrix and H denotes conjugate transpose. The LLR is given by

$$LLR(c_{n,k}) = \frac{Pr(c_{n,k} = 1|\mathbf{r}_n)}{Pr(c_{n,k} = 0|\mathbf{r}_n)} \tag{6}$$

which can be expressed as

$$\begin{aligned} LLR(c_{n,k}) &= \ln \frac{\sum\limits_{\forall S_i : c_{n,k}=1} Pr(\mathbf{r}_n|s_n = S_i) Pr(s_n = S_i)}{\sum\limits_{\forall S_i : c_{n,k}=0} Pr(\mathbf{r}_n|s_n = S_i) Pr(s_n = S_i)} \\ &= \ln \frac{\sum\limits_{\forall S_i : c_{n,k}=1} \frac{Pr(\mathbf{r}_n|s_n=S_i)}{Pr(\mathbf{r}_n|s_n=S_a)} Pr(s_n = S_i)}{\sum\limits_{\forall S_i : c_{n,k}=0} \frac{Pr(\mathbf{r}_n|s_n=S_i)}{Pr(\mathbf{r}_n|s_n=S_a)} Pr(s_n = S_i)} \end{aligned} \tag{7}$$

where S_i are the constellation symbols and S_a is an arbitrarily selected constellation symbol such that $Pr(\mathbf{r}_n|s_n = S_a) \neq 0$. Due to the Gaussian approximation, the term in the summation is given by

$$\begin{aligned} &\frac{Pr(\mathbf{r}_n|s_n = S_i)}{Pr(\mathbf{r}_n|s_n = S_a)} = \\ &= \frac{K \exp\left(-\frac{1}{2}\left[\mathbf{r}_n - E(r_n|S_i)\right] \mathbf{\Lambda}^{-1} \left[\mathbf{r}_n - E(r_n|S_i)\right]^H\right)}{K \exp\left(-\frac{1}{2}\left[\mathbf{r}_n - E(r_n|S_a)\right] \mathbf{\Lambda}^{-1} \left[\mathbf{r}_n - E(r_n|S_a)\right]^H\right)} \\ &= \exp(Re\left[(S_i - S_a)^* (\mathbf{r}_n \mathbf{d}_n^H - E(\mathbf{s}_n)\mathbf{f}_n^H)\right] \\ &\qquad - G\left[|S_i|^2 - |S_a|^2\right]) \end{aligned} \tag{8}$$

where $\mathbf{d}_n$ is equal to

$$\mathbf{d}_n = \mathbf{h}\mathbf{\Lambda}_n^{-1} \tag{9}$$

$\mathbf{f}_n$ is given by

$$\mathbf{f}_n = \mathbf{h}\mathbf{\Lambda}_n^{-1}\mathbf{H}_c^H = \mathbf{d}_n \mathbf{H}_c^H, \quad f_0 = 0 \tag{10}$$

f_0 is the center tap of filter $\mathbf{f}_n$ and G_n is a constant

$$G_n = \frac{1}{2}\mathbf{h}\mathbf{\Lambda}_n^{-1}\mathbf{h}^H \tag{11}$$

Here $\mathbf{\Lambda}_n$ is the cross-correlation matrix

$$\mathbf{\Lambda} = \mathbf{H}_c^H \mathbf{\Gamma} \mathbf{H}_c + \frac{N_0}{2}\mathbf{I} \tag{12}$$

where $\mathbf{\Gamma} = \text{diag}(\xi_{n-N_1-L}^2, \cdots, \xi_{n-1}^2, 0, \xi_{n+1}^2, \cdots, \xi_{n+N_2}^2)$ and $\{\xi_n^2\}$ are the symbol variances, which along with the mean $E(s_n)$ are calculated from the extrinsic information $\{\lambda_n\}$ supplied by the turbo channel decoder. Both, are calculated under the assumption that the bits forming the symbol are independent. For example, for BPSK modulation the mean is equal to

$$\begin{aligned} E\{s_n\} &= 1 \cdot Pr(s_n = 1) - 1 \cdot Pr(s_n = -1) \\ &= \frac{e^{\lambda_n} - 1}{e^{\lambda_n} + 1} \end{aligned} \tag{13}$$

and the variance ξ_n^2 is given by

$$\begin{aligned} \xi_n^2 &= 1/2E\left([s_n - E(s_n)][s_n - E(s_n)]^*\right) \\ &= 1^2 \cdot Pr(s_n = 1) + (-1)^2 \cdot Pr(s_n = -1) - E(s_n)^2 \\ &= 1 - \left(\frac{e^{\lambda_n} - 1}{e^{\lambda_n} + 1}\right)^2 \end{aligned} \tag{14}$$

The equalizer structure can now be identified from (7) and (8) as having the structure in Fig. 2. The structure has the following parts: 1) filter $\mathbf{d}_n$ accepts the corrupted channel signals, 2) filter $\mathbf{f}_n$ accepts an estimate of the symbols calculated from the extrinsic information provided by the turbo decoder, 3) an LLR processor, which is a non-linear memoryless processor for calculating the LLRs of each code bit according to (7), 4) an extrinsic processor for calculating the symbol probabilities and accordingly the symbols' mean and variances and 5) a tap gain update module for evaluating the inverse of the cross-correlation matrix and the tap gains of the filters.

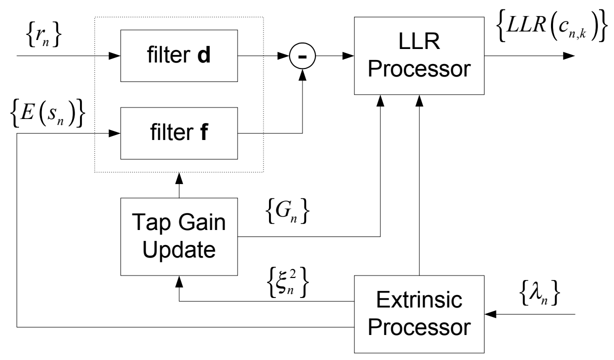

Fig. 2. Equalizer Structure.

We explore the equations of (9) and (10) for three cases: 1) the first iteration where no *a posteriori* information is available, 2) the "last" iteration where the mean symbol estimates equal their transmitted value and 3) "in between" iterations where *a posteriori* information is available but the symbol estimation do not equal their transmitted value.

<u>Case 1</u>: For the first iteration, no *a posteriori* is available and therefore filter $\mathbf{f}$ is redundant. The tap gains of filter $\mathbf{d}$ are constant throughout the iteration and are given by

$$\mathbf{d} = \frac{E_s}{mse}\mathbf{h}_0 \left(\frac{E_s}{2}\mathbf{H}_c^H \mathbf{H}_c + \frac{N_0}{2}\mathbf{I} \right)^{-1} \tag{15}$$

where E_s is the average constellation energy and mse is the mean square error at the output of a linear minimum mean square error (MMSE) equalizer. It can be recognized that the tap gains of $\mathbf{d}$ are equal to the MMSE solution [9] scaled by the SNR at the output of the equalizer.

<u>Case 2</u>: For the "last" iteration where the estimated symbol values equal their transmitted value i.e. $E(s_n) = s_n$

$$\mathbf{d} = \frac{1}{N_0/2}\mathbf{h}, \qquad \mathbf{f} = \frac{1}{N_0/2}\mathbf{h}\mathbf{H}_c^H, \quad f_0 = 0 \tag{16}$$

$\mathbf{d}^H$ is the channel's matched filter and the component of $\mathbf{f}$ are the aperiodic autocorrelation function of the channel coefficients. Filter $\mathbf{d}$ maximizes the SNR at its output while filter $\mathbf{f}$ is able to completely eliminate the ISI.

<u>Case 3</u>: For "in between" iterations, it is necessary to calculate the inverse of $\mathbf{\Lambda}_n$ given in (12) for every symbol entering the filters. Consequently, filters $\mathbf{d}$ and $\mathbf{f}$ change every symbol. This yields a complexity measured in the number of complex multiplications of $O(N^3)$ per symbol. Due to the structure of $\mathbf{\Lambda}_n$, a recursive algorithm for calculating the inverse can be derived which yields a complexity of $O(N^2)$ per symbol [11]. Further complexity reduction can be achieved if the tap gains of filter $\mathbf{d}$ and $\mathbf{f}$ are calculated using the average symbol variances in the block. In this case, the tap gains are constant for each iteration but change from one iteration to the next.

For BPSK modulation, the LLR processor is unnecessary and the equalizer structure is simplified to

$$LLR(c_n) = 2\left(\mathbf{r}_n\mathbf{d}^T - E(\mathbf{s}_n)\mathbf{f}^T\right) + \ln\frac{Pr(s_n = +1)}{Pr(s_n = -1)} \tag{17}$$

For this case an alternative tap optimization criterion was considered in [10]. We note that even though the equalizer structure is the same as in [7] there are several differences in the governing equations: 1) the values of the tap gains are different for "in between" iteration and 2) the LLR processor is more efficient in the use of the extrinsic information provided by the turbo decoder.

IV. Performance and Results

We present simulation results for the channel $\{h_l\} = [0.1275, 0.450, 0.750, 0.450, 0.1275]$. The transmitter consists of two identical RSC encoders (code rate of 1/3) with generating polynomials $(37, 21)_8$ separated by a random interleaver of length 10000 followed by a block interleaver with 300 rows and 100 columns. The lengths of the filters $\mathbf{d}$ and $\mathbf{f}$ were fixed at 31. The performance of four equalizers types were compared: 1) APP module 2) Gaussian Approximation (GA) in which the tap gains change every symbol 3) Gaussian approximation with constant tap gains during the block (GAC) and 4) equalizer according to [7] named after the authors (GLL). To increase performance we modified the original GLL such that the *a priori* information was used more efficiently. For BPSK modulation six iterations were performed. At the 6^{th} iteration, the GA gains 0.2dB vs. GAC in Fig. 3 and is inferior by 0.5dB to the APP module in Fig. 4. The GAC gains 0.2dB over GLL in Fig. 5. For 16QAM modulation the first six iterations along with the 12^{th} iteration are presented. At the 12^{th} iteration, the GA gains 0.5dB vs. GAC in Fig. 6 and the GAC gains 0.1dB over GLL in Fig. 7.

V. Conclusions

In this paper, a reduced complexity turbo equalizer was investigated. We approximated the ISI by a random variable having a Gaussian distribution and evaluated the LLR of each code bit. The equalizer structure that stems from this approximation consists of two linear transversal filters followed by a simple non-linear memoryless processor. Theoretical equations were

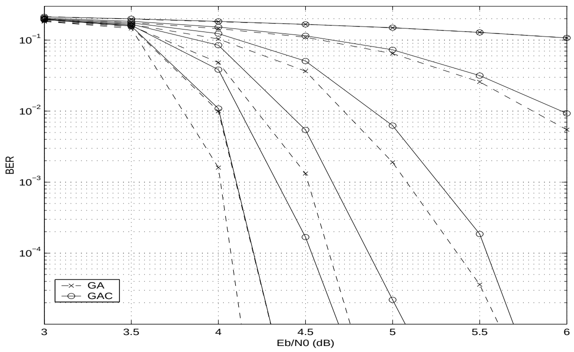

Fig. 3. GA vs. GAC BPSK.

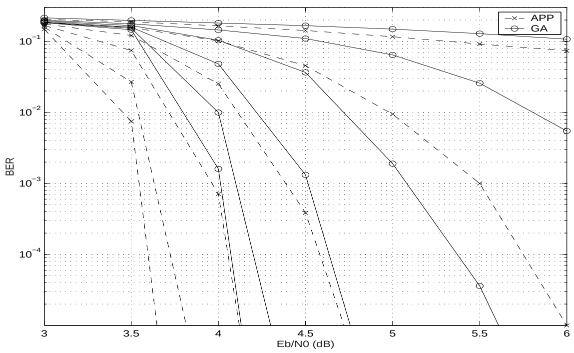

Fig. 4. GA vs. APP BPSK.

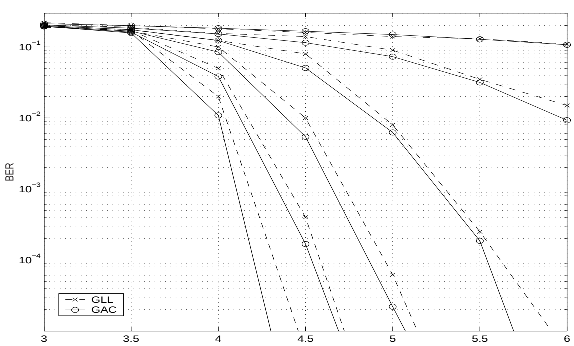

Fig. 5. GAC vs. GLL BPSK.

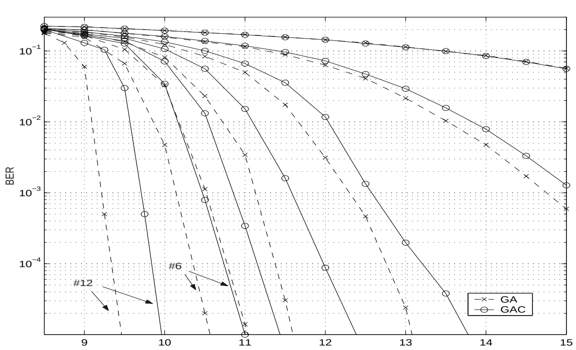

Fig. 6. GA vs. GAC 16QAM.

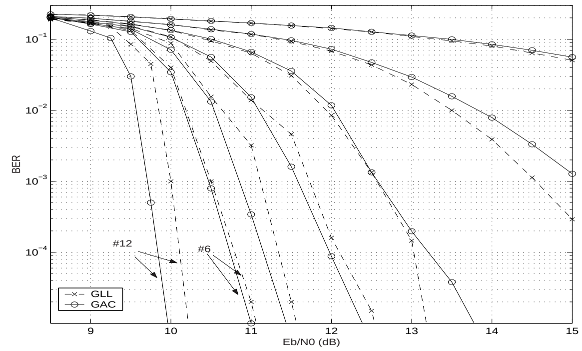

Fig. 7. GAC vs. GLL 16QAM.

presented for determining the tap gains of the filters for all iterations. The tap gains change for every symbol entering the filter as a function of the extrinsic information delivered by the turbo channel decoder. The complexity is dominated by the calculation of the filters' tap gains and yields a complexity of $O(N^3)$ (where N is the length of the filters) measured in the number of complex multiplication for every symbol. A recursive algorithm for calculating the tap gains exists, which yields a complexity of $O(N^2)$. Linear complexity $O(N)$ can be obtained if the tap gains are constant for each iteration (but changing from one iteration to the next) by using the average of the estimated symbols' variance in the block. The performance of the equalizer was compared to [7] (with improvements) where the optimization criterion was the MMSE. For the example shown, the gain compared to [7] is about 0.4dB at the 6^{th} iteration for BPSK and 0.6dB at the 12^{th} iteration for 16QAM.

References

[1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," *Proc. of ICC'93,* Geneva, Switzerland, May 1993, pp. 1064-1070.

[2] J. Hagenauer, "The turbo principle: Tutorial introduction and state of the art," International Symposium on Turbo Codes, Brest, France, 1997, pp. 1-11.

[3] 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, Mar. 1974.

[4] S. Bendetto, D. Divsalar, G. Montrosi and F. Pollara, "A soft-input soft-ouput APP module for iterative decoding of concatenated codes," *IEEE Comm. Letters,* vol. 1, No. 1, pp. 22-24, Jan. 1997.

[5] C. Douillard, M. Jezequel, C. Berrou, A. Picart and A. Glavieux, "Iterative correction of intersymbol interference: Turbo-equalization," European Trans. on Telecommunications, Vol. 6 No. 5, Sept.-Nov. 1995, pp. 507-511.

[6] D. Raphaeli and Y. Zarai, "Combined turbo equalization and turbo decoding," International Symposium on Turbo Codes, Brest, France, 1997, pp. 180-183.

[7] A. Glavieux, C. Laot and J. Labat, "Turbo equalization over a frequency selective channel," International Symposium on Turbo Codes, Brest, France, 1997, pp. 96-102.

[8] J. G. Proakis, "Digital Communication," NY, McGraw-Hill, 3rd Edition 1995.

[9] A. P. Clark, "Equalizers for Digital Modems," Pentech Press 1985.

[10] D. Raphaeli and A. Saguy, "Linear equalizer for turbo equalization - A new optimization criterion for determining the equalizer taps," International Symposium on Turbo Codes, Brest, France, 2000.

[11] A. Saguy, "Reduced complexity turbo equalization," M.Sc. Thesis, Tel-Aviv University, 2002.