

Combined Turbo Equalization and Turbo Decoding

Dan Raphaeli, *Member, IEEE*, and Yoram Zarai

Abstract—In this letter, the subject of turbo coding in the presence of intersymbol interference channel will be investigated. An iterative decoder structure will be presented, which combines the channel equalization and the turbo decoding. At each iteration extrinsic information from the channel detector is fed into the turbo decoders, and then their extrinsic information is fed back to the channel detector. Simulation results are presented for rate 1/2 turbo code with binary phase-shift keying (BPSK) modulation, transmitted over intersymbol interference (ISI) channel having severe frequency distortion. The performance is about 0.8 dB from the ISI channel capacity at bit-error rate of 10^{-5} .

Index Terms—Equalization, ISI, turbo codes.

I. INTRODUCTION

IN THE AREA of coding theory and digital communication, turbo codes made the most exciting development in the last years. They were first introduced by [1] and since have been the object of great interest, and consequently of wide investigation in the coding community.

This letter investigates the performance of a turbo code in the presence of an intersymbol interference (ISI) channel. The general concept of turbo equalization, introduced in [2], used a convolutional code as the channel encoder. Here, we have used a turbo code to benefit from its high coding gain, and to achieve near capacity performance.

In this letter, we present a decoder structure which combines turbo coding gain with the ISI mitigation. We assume that the receiver input is preceded by a matched filter followed by a noise whitening filter (also called a whitened matched filter) [3]. The cascade of the pulse shape filter, linear channel distortion, whitened matched filter and symbol rate sampling can be represented as an equivalent discrete-time transversal filter (DTTF) $F(z)$, having the set $\{f_n\}$ as its tap coefficients [3]

$$r_k = \sum_{n=0}^L f_n c_{k-n} + \eta_k \quad (1)$$

where $\{r_k\}$ is the DTTF output sequence, $\{\eta_k\}$ is a white Gaussian noise sequence (having zero mean and variance N_0) and $\{c_k\}$ is the input symbol sequence. The channel ISI length is assumed to be $L + 1$.

To overcome the discrete-time channel selectivity, it is possible to use equalization independent to the decoding, with a certain loss in performance. Another approach, used in this

Manuscript received January 1997; revised November 18, 1997. The associate editor coordinating the review of this letter and approving it for publication was Prof. Y. Bar-Ness.

The authors are with the Electrical Engineering-Systems Department, Tel Aviv University, Tel Aviv 69978, Israel (e-mail: danr; yoramz@eng.tau.ac.il).
Publisher Item Identifier S 1089-7798(98)03177-9.

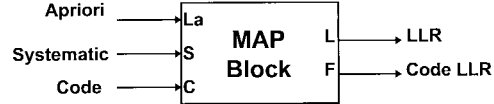


Fig. 1. A MAP block.

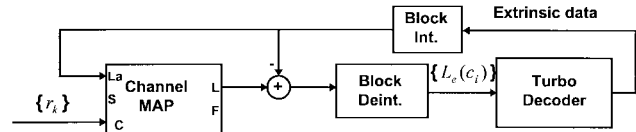


Fig. 2. General combined decoder structure.

paper as in [2], takes the discrete channel memory effect into account in the iterations of the turbo decoder. The DTTF can be modeled as a Markov chain and its behavior can be represented by a trellis diagram [3]. The channel detector and the turbo component codes decoder are implemented by the symbol by symbol maximum *a posteriori* (MAP) algorithm, that yields the log-likelihood ratio (LLR) of the decoded and encoded symbols [4]. In any iteration of the combined decoder extrinsic information from the channel detector is fed into the turbo decoders, and then their extrinsic information is fed back to the channel detector.

II. SYSTEM MODEL

Consider that binary source bits $\{d_i\}$ are encoded by a turbo encoder, similar to the one used in [1]. The turbo encoder is made of two identical recursive encoders, separated by a random interleaver (π). The coded bits, $\{c_i\}$, are block interleaved, BPSK modulated and transmitted over a band-limited channel. The BPSK's output symbols will be referred as $\{c_k\}$.

III. THE DECODER

Before we present the decoder structure, we first define the following. A *MAP block*, illustrated in Fig. 1, is a soft in soft out MAP decoder based on [4]. The underlying trellis code can be either of the convolutional code or the ISI channel. For a systematic encoder having a_i as the systematic data and b_i as the coded data, the MAP block has the following inputs and outputs:

A priori (L_a) *a priori* information of the source data in LLR format:

$$L_a(a_i) = \log \frac{\text{Prob}(a_i = +1)}{\text{Prob}(a_i = -1)}. \quad (2)$$

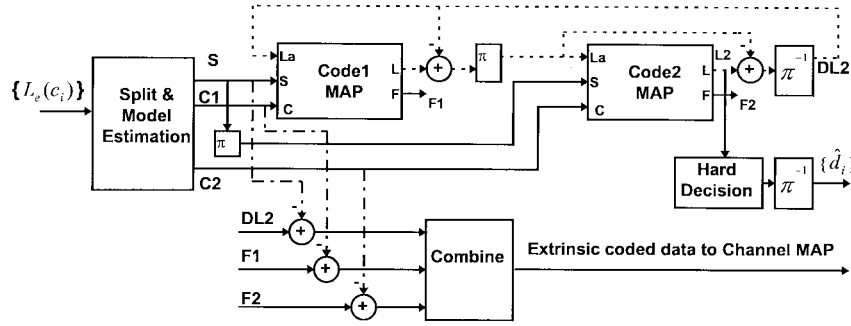


Fig. 3. Turbo decoder structure. S : Estimated turbo systematic data; $C1$: Estimated turbo Code1 data; $C2$: estimated turbo Code2 data; $DL2$: deinterleaved LLR output of Code2 MAP; $F1$: code LLR output of Code1 MAP; $F2$: code LLR output of Code2 MAP; - - -: iterative turbo decoding of source data; in LLR format (see Section V).

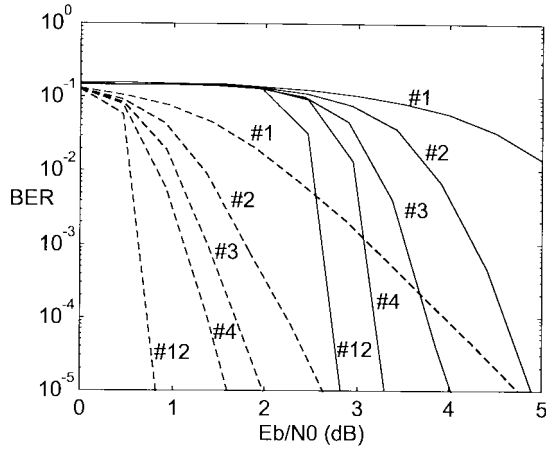


Fig. 4. BER versus E_b/N_0 in white noise channel and ISI channel.

Systematic (S) received systematic symbols. In the ISI case there is no systematic data;
Code (C) received coded symbols;
LLR (L) log-likelihood ratio of the source data conditioned by the MAP block inputs:

$$L(a_i) = \log \frac{\text{Prob}(a_i = +1/L_a, S, C)}{\text{Prob}(a_i = -1/L_a, S, C)}. \quad (3)$$

Code LLR (F) LLR of the coded data (sometimes referred as “MAP filter”) condition by the MAP block inputs:

$$F(b_i) = \log \frac{\text{Prob}(b_i = +1/L_a, S, C)}{\text{Prob}(b_i = -1/L_a, S, C)}. \quad (4)$$

The general decoder structure is presented in Fig. 2. The channel detector is a MAP block that receives the corrupted symbols from the channel and evaluates the LLR of the turbo-coded symbols c_k . It has no systematic data.

The turbo decoder, illustrated in Fig. 3, is based on two MAP blocks. Each of them receives its information from the channel MAP and evaluates the LLR of both the source and coded symbols.

Extrinsic information is extracted, at each iteration, from both the channel MAP and the turbo MAP's. The extrinsic coded data, that will be used by the channel MAP as *a priori* input in the next iteration, is evaluated by subtracting the turbo MAP's code's LLR from the extrinsic channel MAP's output in the current iteration (after deinterleaving).

IV. EQUIVALENT CHANNEL MODEL

In order to use the channel MAP's LLR output by the turbo decoder (as systematic and coded data), a translation from LLR format to equivalent soft channel input is needed. This is done by the following equivalent channel model.

We assume that the channel MAP LLR's output represents the log-likelihood ratio of an equivalent additive white noise channel having $\{c_k\}$ as its input symbol sequence. This equivalent channel can be represented by the following equation:

$$\tilde{c}_k = \alpha c_k + n_k \quad (5)$$

where \tilde{c}_k is the received symbol, α is the equivalent channel attenuation and n_k/α is the equivalent normalized white noise. For this equivalent channel, the LLR can be easily calculated by

$$L(c_k) = \frac{2}{\sigma_q^2} \tilde{c}_k + L_a(c_k) \quad (6)$$

thus

$$\tilde{c}_k = \frac{\sigma_q^2}{2} L_e(c_k) \quad (7)$$

where $L(c_k)$ is the channel MAP's LLR output, $L_a(c_k)$ is the *a priori* value of c_k , $L_e(c_k) = L(c_k) - L_a(c_k)$ is the channel MAP extrinsic information and σ_q^2 is the equivalent normalized white noise variance.

By using (5) and (7), the estimation of σ_q^2 is done by

$$\sigma_q^2 = \frac{1}{\alpha^2} E\{n_k^2\} = \frac{E\{L_e(c_k)^2\}}{[E\{L_e(c_k)\hat{c}_k\}]^2} - 1 \quad (8)$$

where $\hat{c}_k = \text{sign}\{L(c_k)\}$.

The *split and model estimation* block in the turbo decoder structure has to estimate the equivalent variance using (8), translate the turbo code's LLR data to equivalent soft channel input \tilde{c}_k using (7), and split this data to the turbo MAP blocks. This is done in each iteration.

V. PERFORMANCE AND RESULTS

The performance of the decoder has been evaluated by simulations. A turbo code of rate 1/2 with two identical recursive systematic encoders (37, 21) separated by a random interleaver of length 10 000 was simulated. The coefficients of the band limited channel are $f_0 = \sqrt{0.45}$, $f_1 = \sqrt{0.25}$, $f_2 = \sqrt{0.15}$, $f_3 = \sqrt{0.1}$, $f_4 = \sqrt{0.05}$. The block interleaver rows and columns are 200 and 100, respectively.

The BER versus E_b/N_0 , for iteration 1–4 and 12, is plotted in Fig. 4. The dash line is for the turbo code in a AWGN channel without ISI.

The channel capacity limit of the simulated ISI channel is about 2 dB. The performance of our decoder after 12 iterations is about 0.8 dB from this limit.

REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error—Correcting coding and decoding: Turbo-codes," in *ICC'93*, Geneva, Switzerland, May 1993, pp. 1064–1070.
- [2] C. Douillard, M. Je'ze'quel, and C. Berrou, "Iterative correction of intersymbol interference: Turbo-Equalization," *European Trans. Telecommun.*, vol. 6, no. 5, pp. 507–511, Sept./Oct. 1995.
- [3] J. G. Proakis, *Digital Communication*, 2nd ed. New York: McGraw-Hill, 1989.
- [4] S. Pietrobon and S. Barbulescu, "A simplification of the modified Bahl decoding algorithm for systematic convolution code," in *ISITA'94*, Sydney, Australia, Nov. 1994, pp. 1073–1077.