The Soft-Feedback Equalizer for Turbo Equalization of Highly Dispersive Channels

Renato R. Lopes and John R. Barry, *Senior Member, IEEE*

***Abstract*—The complexity of a turbo equalizer based on the Bahl–Cocke–Jelinek–Raviv (BCJR) algorithm is manageable only for mildly dispersive channels having a small amount of memory. To enable turbo equalization of highly dispersive channels, we propose the *soft-feedback equalizer (SFE)*. The SFE combines linear equalization and soft intersymbol-interference cancellation. Its coefficients are chosen to minimize the mean-squared error (MSE) between the equalizer output and the transmitted sequence, under a Gaussian approximation to the *a priori* information and the SFE output. The resulting complexity grows only linearly with the number of coefficients, as opposed to the quadratic complexity of previously reported minimum-MSE structures. We will see that an SFE-based turbo equalizer consistently outperforms another structure of similar complexity, and can outperform a BCJR-based scheme when complexity is taken into account.**

***Index Terms*—Decision-feedback equalization, interference cancellation, minimum mean-squared error (MMSE) equalization.**

I. Introduction

HIGHLY dispersive channels arise in a wide variety of communications systems, including microwave links, powerline communications, magnetic recording, twisted pairs, and coaxial cables. In such systems, the channel impulse response can span tens or even hundreds of symbol periods. How can these systems profit from a turbo equalizer [1], [2]? The conventional approach to turbo equalization uses a soft-input soft-output (SISO) equalizer based on the forward–backward algorithm of Bahl, Cocke, Jelinek, and Raviv (BCJR) [3], but the computational complexity of this algorithm increases exponentially with the channel memory. This has motivated the development of reduced-complexity alternatives to the BCJR equalizer, such as the *soft interference cancellers* proposed in [4]–[9]. These structures use a linear filter to equalize the received sequence. The output of this filter contains residual intersymbol interference (ISI), which is estimated based on the *a priori* information, and then cancelled.

Paper approved by G. M. Vitetta, the Editor for Equalization and Fading Channels of the IEEE Communications Society. Manuscript received August 20, 2004; revised May 4, 2005 and August 4, 2005. This work was supported in part by the National Science Foundation under Grants CCR-0082329 and CCR-0121565, in part by the Information Storage Industry Consortium, and in part by the Funda ão de Amparo Pesquisa do Estado de São Paulo (FAPESP) under Grants 03/07814-1 and 02/12216-3. This paper was presented at Globecom, San Francisco, CA, December 2003.

R. R. Lopes was with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA. He is now with the School of Electrical and Computer Engineering, University of Campinas, CP 6101, 13083-970 Campinas, SP, Brazil (e-mail: rlopes@decom.fee.unicamp.br).

J. R. Barry is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA (e-mail: barry@ece.gatech.edu).

Digital Object Identifier 10.1109/TCOMM.2006.874009

In this letter, we propose the *soft-feedback equalizer* (SFE), a low-complexity structure that enables turbo equalization for highly dispersive channels. The SFE is also based on filtering and cancellation of residual ISI. However, unlike the interference cancellers of [4]–[9], the SFE uses a structure similar to a decision-feedback equalizer (DFE), combining the equalizer outputs and *a priori* information to form more reliable estimates of the residual postcursor ISI. As in [6]–[9], the SFE coefficients are computed so as to minimize the mean-squared error (MSE) between the equalizer output and the transmitted symbol. However, the coefficients of the minimum-MSE (MMSE) structures in [6]–[9] have to be computed anew for every symbol, even when the channel is static, resulting in a per-symbol complexity that is *quadratic* in the length of the equalizer. In contrast, by adopting a simple statistical model for the equalizer outputs and *a priori* information, we obtain a time-invariant, linear-complexity MMSE equalizer. As in [6]–[9], the resulting equalizer coefficients depend on the quality of the equalizer output and the *a priori* information.

We will see that in special cases, the SFE reduces to the MMSE linear equalizer, the MMSE DFE, and the interference canceller. We will show that, when computational complexity is taken into account, an SFE-based turbo equalizer can outperform a BCJR-based turbo equalizer. Finally, we will see that the SFE consistently outperforms the linear-complexity structure proposed in [9].

The remainder of the letter is organized as follows. In Section II, we describe the problem addressed in this letter. In Section III, we propose the SFE. In Section IV, we show some simulation results. Finally, in Section V, we draw some concluding remarks.

II. Problem Statement

We consider the transmission of a sequence of coded and interleaved symbols $\{a_0, \dots, a_{N-1}\}$ through a channel whose output at time k is given by

$$r_k = \sum_{i=0}^{\mu} h_i a_{k-i} + n_k \tag{1}$$

where h_k is the channel impulse response with memory μ, and where n_k is additive white Gaussian noise (AWGN) with variance σ^2. We assume that h_k and σ^2 are known to the receiver. For notational ease, we restrict our presentation to the binary phase-shift keying (BPSK) alphabet, $a_k \in \{\pm 1\}$. The results can be extended to other alphabets using the techniques described in [7].

We consider a receiver using turbo equalization, whereby a SISO equalizer interacts with a SISO error-control decoder. The decoder provides *a priori* information to the equalizer, in the

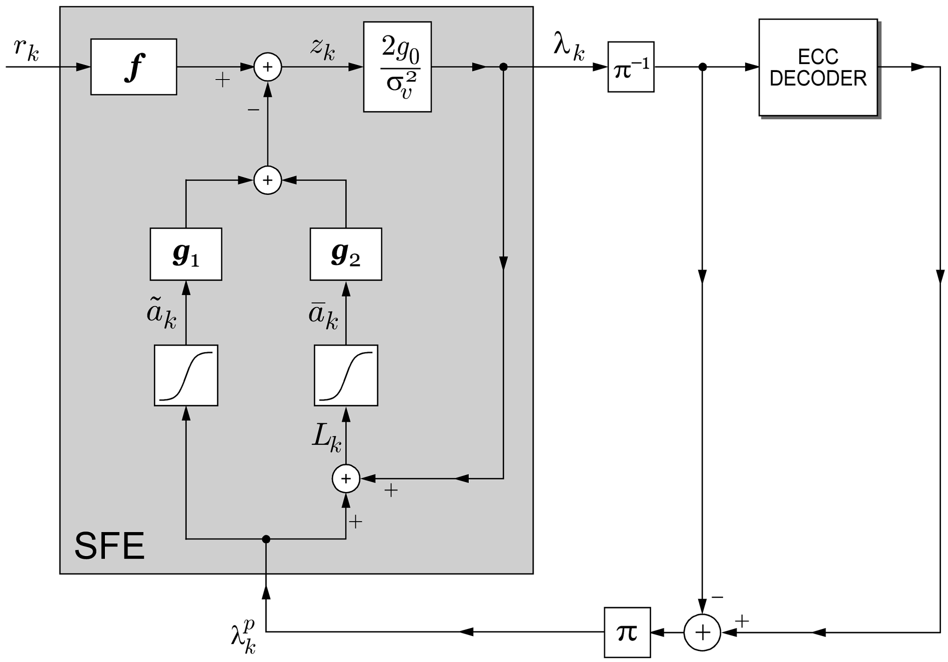

Fig. 1. An SFE-based turbo equalizer.

form of the logarithm of the ratio of the *a priori* probabilities, loosely known as the *a priori log-likelihood ratio (LLR)*

$$\lambda_k^p = \log \frac{\Pr(a_k = +1)}{\Pr(a_k = -1)}. \tag{2}$$

Our goal is to design an equalizer that, given $\{\lambda_0^p, \dots, \lambda_{N-1}^p\}$ and $\boldsymbol{r} = [r_0, \dots, r_{N+\mu-1}]$, computes the logarithm of the ratio of the *a posteriori* probabilities, loosely known as the *full LLR*

$$L_k = \log \frac{\Pr(a_k = +1|\boldsymbol{r})}{\Pr(a_k = -1|\boldsymbol{r})}. \tag{3}$$

The equalizer then passes the so-called *extrinsic LLR* $\lambda_k = L_k - \lambda_k^p$ to the error-control decoder. Note that λ_k is not a function of λ_k^p [9].

III. The Soft-Feedback Equalizer

Fig. 1 shows a block diagram of the proposed SFE in the context of a turbo equalizer. In this figure, the received signal r_k is first filtered by a linear filter $\boldsymbol{f}$, whose output contains residual ISI. At time k, when the equalizer seeks to estimate a_k, the *a priori* information $\{\lambda_{l\neq k}^p\}$ for the *interfering* symbols, as well as the previous equalizer outputs, are used to estimate and cancel the residual ISI. Specifically, let

$$\tilde{a}_l = \mathrm{E}\left[a_l|\lambda_l^p\right] = \tanh(\lambda_l^p/2) \tag{4}$$

be a soft estimate of the lth transmitted symbol. The SFE feeds the estimates $\{\tilde{a}_{l>k}\}$ of the future interfering symbols $\{a_{l>k}\}$ through the filter $\boldsymbol{g}_1$, whose output is an estimate of the anticausal (precursor) part of the residual ISI at the output of $\boldsymbol{f}$. This estimate is then subtracted from the output of $\boldsymbol{f}$, effectively cancelling the precursor ISI.

We could also use $\tilde{a}_l$ to cancel the postcursor ISI, as is done in [4]–[9]. However, at time k, the extrinsic information of previous symbols, $\{\lambda_{k-j} : j > 0\}$, has already been computed by the equalizer. With these values, we may compute the full LLR $L_{k-j} = \lambda_{k-j}^p + \lambda_{k-j}$, which provides a better estimate of a_{k-j} than the *a priori* LLR λ_{k-j}^p alone. Thus, instead of using $\tilde{a}_{k-j}$ to cancel postcursor interference, the SFE uses $\bar{a}_{k-j} = E[a_{k-j}|L_{k-j}]$ for $j > 0$. These estimates are fed through the filter $\boldsymbol{g}_2$, whose output is then an estimate of the causal (postcursor) part of the residual ISI at the output of $\boldsymbol{f}$. This idea is similar in spirit to the principle behind a DFE. A DFE-based structure was also proposed in [9], but it feeds back hard decisions on the equalizer output, without combining them with the *a priori* information. A DFE system was also proposed in [10], but it does not use the *a priori* information to cancel precursor ISI.

We now follow the strategy of [4]–[9] to compute the extrinsic information that the equalizer must pass on to the decoder. To that end, let the equalizer output (after cancellation) be written as

$$z_k = g_0 a_k + v_k \tag{5}$$

where $g_0 = \sum_i h_i f_{-i}$, and where $v_k = z_k - g_0 a_k$ is the channel noise and residual ISI still present in z_k. Although v_k depends on $\{a_{i\neq k}\}$, it is clear from (5) and the definition of g_0 that v_k is independent of a_k. Following [4]–[9], let us further assume that v_k is a sequence of independent and identically distributed

Gaussian random variables with variance σ_v^2. Then, the LLR of a_k given z_k is given by

$$\lambda_k = 2g_0 z_k/\sigma_v^2. \tag{6}$$

Recall that the filters $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are strictly anticausal and causal, respectively. This implies that λ_k is not a function of λ_k^p. Thus, λ_k will be considered as extrinsic LLR.

A. The SFE Coefficients

We now compute the time-invariant filters $\boldsymbol{f}$, $\boldsymbol{g}_1$, and $\boldsymbol{g}_2$ that minimize the MSE $E[|z_k - a_k|^2]$. To that end, we write the SFE output z_k as

$$z_k = \boldsymbol{f}^T\boldsymbol{r}_k - \boldsymbol{g}_1^T\tilde{\boldsymbol{a}}_k - \boldsymbol{g}_2^T\overline{\boldsymbol{a}}_k \tag{7}$$

where $\boldsymbol{f} = [f_{-M_1}, \ldots, f_{M_2}]^T$, $\boldsymbol{r}_k = [r_{k+M_1}, \ldots, r_{k-M_2}]^T$, $\boldsymbol{g}_1 = [g_{-M_1}, \ldots, g_{-1}]^T$, $\boldsymbol{g}_2 = [g_1, \ldots, g_{M_2+\mu}]^T$, $\tilde{\boldsymbol{a}}_k = [\tilde{a}_{k+M_1}, \ldots, \tilde{a}_{k+1}]^T$, $\overline{\boldsymbol{a}}_k = [\overline{a}_{k-1}, \ldots, \overline{a}_{k-(M_2+\mu)}]^T$, and M_1 and M_2 determine the lengths of the filters. Now, assume that $E[\tilde{a}_k a_j] = E[\overline{a}_k a_j] = E[\tilde{a}_k \overline{a}_j] = 0$ when $k \neq j$, and that $E[|\tilde{a}_k|^2] = E[\tilde{a}_k a_k]$ and $E[|\overline{a}_k|^2] = E[\overline{a}_k a_k]$ [14]. Then, following steps similar to the derivation of a traditional MMSE-DFE, it is easy to show that the minimum MSE values of $\boldsymbol{f}$, $\boldsymbol{g}_1$, and $\boldsymbol{g}_2$ are given by [14]

$$\boldsymbol{f} = \left(\boldsymbol{H}\boldsymbol{H}^T - \alpha_1\boldsymbol{H}_1\boldsymbol{H}_1^T - \alpha_2\boldsymbol{H}_2\boldsymbol{H}_2^T + \sigma^2\boldsymbol{I}\right)^{-1}\boldsymbol{h}_0 \tag{8}$$

$$\boldsymbol{g}_1 = \boldsymbol{H}_1^T\boldsymbol{f} \tag{9}$$

$$\boldsymbol{g}_2 = \boldsymbol{H}_2^T\boldsymbol{f} \tag{10}$$

where $\boldsymbol{H}$ is the $M \times (M+\mu)$ channel convolution matrix

$$\boldsymbol{H} = \begin{bmatrix} h_0 & h_1 & & h_\mu & 0 & 0 & \ldots & 0 \\ 0 & h_0 & h_1 & \ldots & h_\mu & 0 & \ldots & 0 \\ \vdots & & \ddots & \ddots & & \ddots & & \vdots \\ 0 & \ldots & \ldots & 0 & h_0 & h_1 & \ldots & h_\mu \end{bmatrix} \tag{11}$$

$M = M_1 + M_2 + 1$, $\alpha_1 = E[\tilde{a}_k a_k]$, $\alpha_2 = E[\overline{a}_k a_k]$. The vector $\boldsymbol{h}_0$ is the zeroth column of $\boldsymbol{H}$, where the columns of $\boldsymbol{H}$ are numbered as $\boldsymbol{H} = [\boldsymbol{h}_{-M_1}, \ldots, \boldsymbol{h}_{M_2+\mu}]$. Also, $\boldsymbol{H}_1 = [\boldsymbol{h}_{-M_1}, \ldots, \boldsymbol{h}_{-1}]$ and $\boldsymbol{H}_2 = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{M_2+\mu}]$.

The constant $g_0 = \sum_i h_i f_{-1}$ can be written as $g_0 = \boldsymbol{f}^T\boldsymbol{h}_0$. The constant σ_v^2, needed in (6) to compute the extrinsic LLR λ_k from the equalizer output, is shown in [14] to satisfy $\sigma_v^2 = g_0(1-g_0)$. Thus, (6) reduces to

$$\lambda_k = 2z_k/(1-g_0). \tag{12}$$

This equation implies that $|\lambda_k| = \infty$ whenever $g_0 = 1$, which makes sense, since in this case, the effective noise variance is zero, $\sigma_v^2 = 0$.

B. Computing the Expected Values

Exploiting symmetries, it is not hard to see that α_1 and α_2 may be computed by conditioning on $a_k = 1$. In other words, $\alpha_1 = E[\tilde{a}_k | a_k = 1]$, and $\alpha_2 = E[\overline{a}_k | a_k = 1]$.

Now, assume as in [13] that λ_k^p can be modeled as coming from an equivalent AWGN channel with signal-to-noise ratio (SNR) $\gamma_p/2$, i.e., $\lambda_k^p = \gamma_p a_k + w_k$, where $w_k \sim \mathcal{N}(0, 2\gamma_p)$. Conditioning on $a_k = 1$, it follows that $\lambda_k^p \sim \mathcal{N}(\gamma_p, 2\gamma_p)$, so we get

$$\alpha_1 = \psi_1(\gamma_p) \tag{13}$$

where we have introduced the function

$$\psi_1(\gamma) = \mathrm{E}\left[\tanh(u/2)\right], u \sim \mathcal{N}(\gamma, 2\gamma). \tag{14}$$

Although there is no closed-form formula for $\psi_1(\gamma)$, it is a well-behaved function that may be tabulated or computed by simple numerical algorithms.

Note that the value of γ_p is needed in (13). Fortunately, γ_p can be estimated easily and accurately from the *a priori* information. Indeed, the maximum-likelihood (ML) estimate of γ_p is given by

$$\hat{\gamma}_p = \sqrt{1 + \frac{1}{N}\sum_{k=0}^{N-1} |\lambda_k^p|^2} - 1. \tag{15}$$

A similar approach may be used to compute α_2. Indeed, note that $L_k = \lambda_k^p + \lambda_k$. Now, consider the Gaussian approximation (GA) to λ_k in (5) and (6), and let $\gamma_e = 2g_0^2/\sigma_v^2$ be a parameter that is twice the SNR of the equivalent AWGN channel that generates λ_k. Then, using the GA to λ_k^p, we get that

$$L_k = (\gamma_p + \gamma_e)a_k + \gamma_p w_k + \gamma_e v_k. \tag{16}$$

Therefore, conditioning on $a_k = 1$, $L_k \sim \mathcal{N}(\gamma_p + \gamma_e, 2(\gamma_p + \gamma_e))$, and hence, α_2 is given by

$$\alpha_2 = \psi_1(\gamma_p + \gamma_e). \tag{17}$$

To determine $\gamma_e = 2g_0^2/\sigma_v^2$, we exploit the facts that that $g_0 = \boldsymbol{f}^T\boldsymbol{h}_0$ and $\sigma_v^2/g_0(1-g_0)$, yielding

$$\gamma_e = 2\boldsymbol{f}^T\boldsymbol{h}_0/(1 - \boldsymbol{f}^T\boldsymbol{h}_0). \tag{18}$$

But this is problematic, because we need $\boldsymbol{f}$ in order to compute γ_e, while we need γ_e in order to compute α_2, and thus $\boldsymbol{f}$. To find both simultaneously, we propose that given an initial estimate for γ_e, we compute

$$\boldsymbol{f} = \Big(\boldsymbol{H}\boldsymbol{H}^T - \alpha_1\boldsymbol{H}_1\boldsymbol{H}_1^T - \psi_1(\gamma_p + \gamma_e)\boldsymbol{H}_2\boldsymbol{H}_2^T + \sigma^2\boldsymbol{I}\Big)^{-1}\boldsymbol{h}_0 \tag{19}$$

$$\gamma_e = 2\boldsymbol{f}^T\boldsymbol{h}_0/(1 - \boldsymbol{f}^T\boldsymbol{h}_0) \tag{20}$$

iteratively, until a stop criterion is met. The convergence behavior of this procedure is studied graphically in [14] through plots of the value of γ_e at one iteration versus its previous value.

It is shown that for one particular scenario, the iterative procedure in (19) and (20) presents a single fixed point and fast convergence. We have observed the same fast convergence and unimodal behavior in all scenarios we have studied.

To summarize, the SFE coefficients are computed in four steps.

1) Estimate γ_p using (15).
2) Estimate γ_e using the iterative procedure in (19) and (20).
3) Compute α_1 and α_2 using (13) and (17).
4) Compute the SFE coefficients using (8), (9), and (10).

C. Special Cases

The values of γ_p and γ_e are proportional to the SNR of the equivalent AWGN channels that generate λ_k^p and λ_k, respectively, and hence reflect the quality of these channels. Furthermore, the SFE coefficients depend on γ_p and γ_e through $\psi_1(\cdot)$. It is not hard to show that $\psi_1, (\gamma) \to 0$ as $\gamma \to 0$, and $\psi_1(\gamma) \to 1$ as $\gamma \to \infty$ [14]. Based on these observations, some interesting and intuitively pleasing conclusions may be drawn from a careful inspection of (8)–(10).

- When both $\gamma_p \to 0$ (unreliable *a priori* information) and $\gamma_e \to 0$ (unreliable decisions), the SFE reduces to a conventional MMSE linear equalizer.
- When $\gamma_p \to 0$ (unreliable *a priori* information) and $\gamma_e \to \infty$ (reliable decisions), the SFE reduces to a conventional MMSE-DFE.
- When $\gamma_p \to \infty$ (reliable *a priori* information), the SFE reduces to a traditional ISI canceller. This is known to be optimal when the interfering symbols are known, achieving the matched-filter bound.

D. Turbo Equalization With the SFE

Note that the SFE coefficients depend on the quality of the *a priori* information, which changes with each iteration in a turbo equalizer. Thus, the SFE coefficients must be computed at the beginning of every turbo iteration. However, this computation may be simplified, and simple changes may be introduced to improve the performance of the turbo equalizer. We discuss these changes below.

In (19) and (20), we proposed an iterative procedure to compute γ_e. If we used this strategy in a turbo equalizer, we would have to repeat the iterative procedure for every turbo iteration. However, we observed that no performance loss is incurred if the iterations in (19) and (20) are used only in the first turbo iteration. In later turbo iterations, we may compute the equalizer coefficients using the value γ_e from the previous turbo iteration. An updated value of γ_e is then computed and passed on to the next turbo iteration.

Also, we have observed that the performance of an SFE-based turbo equalizer may be improved if, instead of using (15) and (18), we estimate γ_p and γ_e directly from the equivalent AWGN channels that we assume that generate λ_k^p and λ_k. This can be done with the scalar channel estimator analyzed in [10], repeated here for convenience. Let $z_k = g_0 a_k + v_k$ be the equalizer output. Then, given initial estimates $\hat{g}_{0,0}$ and $\hat{\sigma}_0^2$, γ_e can be estimated iteratively using

$$\begin{aligned} \hat{g}_{0,i} &= \frac{1}{L}\sum_{k=0}^{L-1} \tanh\left(\hat{g}_{0,i-1} z_k / \hat{\sigma}_{i-1}^2\right) z_k \\ \hat{\sigma}_i^2 &= \frac{1}{L}\sum_{k=0}^{L-1} \left(\hat{g}_{0,i}\mathrm{sign}(z_k) - z_k\right)^2 \\ \hat{\gamma}_e^i &= \frac{2\hat{g}_{0,i}^2}{\hat{\sigma}_i^2} \end{aligned} \tag{21}$$

where the index $i > 0$ refers to the turbo iteration. If we replace z_k by λ_k^p in the equations above, we obtain an estimate for γ_p. The initial values $\hat{g}_{0,0}$ and $\hat{\sigma}_0^2$ required to compute $\hat{\gamma}_e^i$ are obtained from the iterative procedure of (19) and (20). For computing $\hat{\gamma}_p^i$, we set $\hat{\sigma}_0^2 = 2\hat{g}_{0,0}$, which reflects our initial approximation that λ_k^p is a Gaussian random variable whose variance is equal to twice its mean.

IV. Simulation Results

In this section, we compare the performance of turbo equalizers based on several different soft-output equalizers. In all cases, the transmitted symbols are encoded with a recursive rate-1/2 convolutional encoder with parity generator $(1 + D^2)/(1 + D + D^2)$, followed by a random interleaver whose length is equal to the block length. We assume that the channel parameters are known to the receiver, and that the error-control decoder is implemented using the BCJR algorithm.

We begin by comparing the performance and complexity of an SFE-based turbo equalizer to a BCJR-based turbo equalizer. We first use the same simulation scenario as [9], in which $K = 2^{15}$ message bits are encoded and transmitted through the channel $h = [0.227, 0.46, 0.688, 0.46, 0.227]$. The performance-complexity tradeoff is illustrated in Fig. 2 by plotting complexity versus performance, where complexity is quantified by the total number of operations (additions and multiplications) required by each turbo equalizer (equalizer and decoder), and where performance is quantified by the value of $E_b/N_0 = \|h\|^2/\sigma^2$ required to achieve a bit-error rate (BER) of 10^{-3}. Each curve is parameterized by the number of turbo iterations. We see from Fig. 2 that, for the channel of [9], the BCJR-based turbo equalizer requires 1.4 times as many operations as the SFE-based turbo equalizer with $M_1 = 9$ and $M_2 = 5$ to achieve a BER of 10^{-3} at $E_b/N_0 = 8$ dB. On the other hand, if we are limited to 400 operations, the SFE-based system can operate at an E_b/N_0 that is 1.5 dB less than that made possible by the BCJR-based system.

In Fig. 2, we also show the performance-complexity tradeoff for a Lorentzian magnetic recording channel (MRC) with channel density $D = 3$ [15]. We use the main nine taps of the MRC before the matched filter, namely $h = [0.037, 0.162, 0.354, 0.071, -0.145, -0.111, -0.068 - 0.043, -0.029]$, which encompass 99.6% of the channel energy. For this channel, we use an SFE with $M_1 = 11$ and $M_2 = 4$. The gap between the SFE- and the BCJR-based systems is even more pronounced in this scenario. Not shown in the figure is the performance of the exact MMSE equalizer

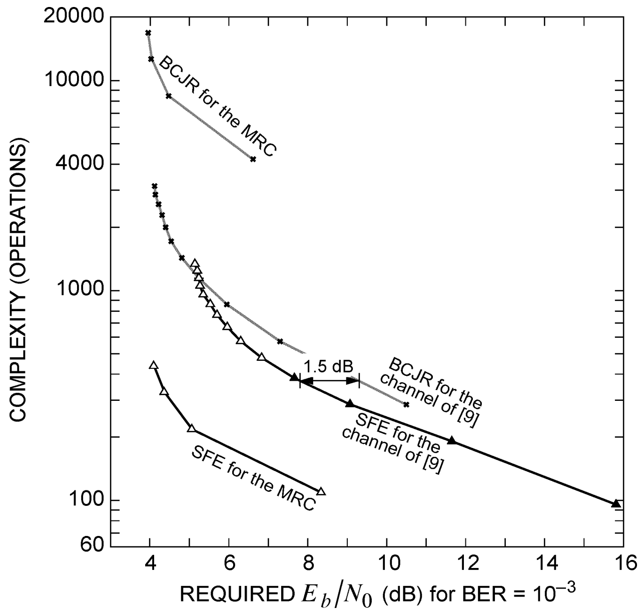

Fig. 2. Complexity-performance tradeoff.

of [9] (EMMSE), because, according to [9, Table I], it is even *more* complex than the BCJR equalizer for both of the scenarios considered.

The channels considered above were short enough that the BCJR equalizer was a feasible option. However, in channels with long impulse responses, the complexity of the BCJR equalizer is prohibitive. To obtain the gains of turbo equalization in these highly dispersive channels, low-complexity equalizers are a must. Consider, for instance, the microwave (MW) channel of [16]. For this example, we focus on the 44-tap section of this channel, corresponding to samples 98–141. Furthermore, since we are using a BPSK modulation, we use only the real part of the channel. For such a long channel, the complexity of BCJR is roughly 2^{47} additions and multiplications per symbol per iteration, and even quadratic-complexity equalizers such as the EMMSE are too complex. In cases like this, linear-complexity equalizers such as the SFE and the hybrid equalizer (HE) of [9] are the only feasible choices.

To determine the performance of the SFE- and HE-based turbo equalizers for highly dispersive channels, we simulate the transmission of $K = 2^{10}$ message bits through the MW channel and through a power line channel (PLC). For the latter, we use the first 58 taps of the channel in [17, Fig. 7], which correspond to 99.9% of the total channel energy. We used equalizers with $M_1 = 40$ and $M_2 = 20$ for the MW channel, and with $M_1 = 69$ and $M_2 = 11$ for the PLC. For each value of E_b/N_0 and every 30 codewords, we checked the total number of words detected in error. If this number was greater than 200, or if more than 20 000 codewords were transmitted, we would stop running the simulation for that particular E_b/N_0. In Figs. 3 and 4, we plot the performance of the turbo equalizers based on the SFE and the HE, in terms of BER versus E_b/N_0 for the MW channel and the PLC, respectively. In these figures, the gains of turbo equalization are clear, as evidenced by the performance gap between the first and tenth iterations of both turbo equalizers with both channels. Furthermore, these figures also highlight the performance

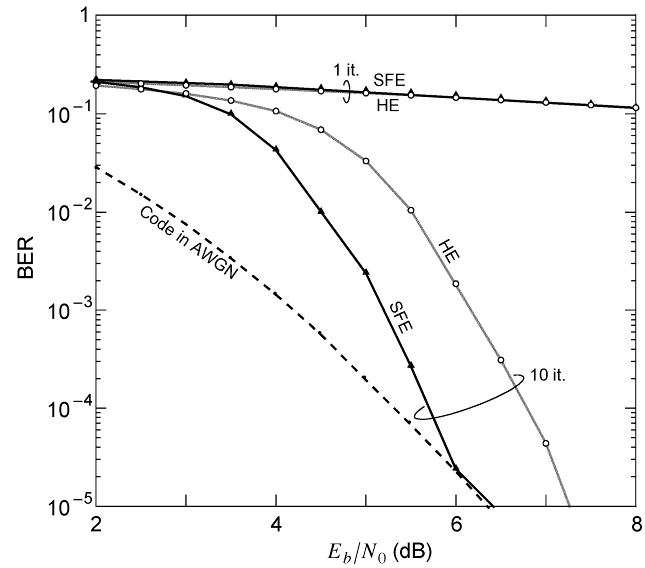

Fig. 3. BER performance of the SFE- and HE-based turbo equalizers for the MW channel.

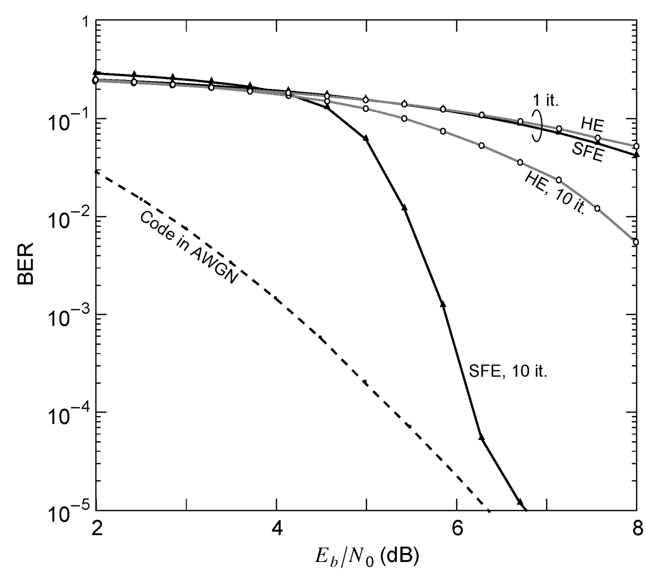

Fig. 4. BER performance of the SFE- and HE-based turbo equalizers for the PLC.

gains of the SFE-based turbo equalizer when compared to the HE-based turbo equalizer: 1.1 dB for a BER of 10^{-4} with the MW channel, more than 4 dB with the PLC. Finally, it is noteworthy that turbo equalization works better for the MW channel, even though the performance at the first iteration is better for the PLC.

V. Conclusions

We have proposed the SFE, a low-complexity structure that enables turbo equalization for highly dispersive channels. In extreme cases, the SFE reduces to the MMSE linear equalizer, the MMSE DFE, and the interference canceller. In the general case, the SFE achieves a compromise between these canonical structures by choosing the equalizer coefficients according to the quality of both the *a priori* information and the equalizer output. When used as a building block in a turbo equalizer, the SFE will start out as a mixture of a linear equalizer or DFE, depending on the channel SNR. As turbo iterations progress and the *a priori* information becomes more reliable, the SFE will gradually and automatically morph into an interference canceller.

We have seen that SFE-based turbo equalizers consistently outperform other algorithms of similar complexity. We have also seen that, on shorter channels for which the BCJR is feasible, an SFE-based turbo equalizer can outperform a BCJR-based equalizer when complexity is taken into account.

References

[1] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo equalization," *Eur. Trans. Telecommun.*, vol. 6, no. 5, pp. 507–511, Sep.–Oct. 1995.

[2] D. Raphaeli and Y. Zarai, "Combined turbo equalization and turbo decoding," *IEEE Commun. Lett.*, vol. 2, no. 4, pp. 107–109, Apr. 1998.

[3] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 3, pp. 284–287, Mar. 1974.

[4] A. Glavieux, C. Laot, and J. Labat, "Turbo equalization over a frequency selective channel," in *Proc. Int. Symp. Turbo Codes, Related Topics*, Brest, France, Sep. 1997, pp. 96–102.

[5] Z. Wu and J. Cioffi, "Turbo decision aided equalization for magnetic recording channels," in *Proc. IEEE Global Commun. Conf.*, Dec. 1999, pp. 733–738.

[6] D. Raphaeli and A. Saguy, "Reduced complexity APP for turbo equalization," in *Proc. IEEE Int. Conf. Commun.*, 2002, vol. 3, pp. 1940–1943.

[7] A. Dejonghe and L. Vanderdorpe, "Turbo equalization for multilevel modulation: An efficient low-complexity scheme," in *Proc. IEEE Int. Conf. Commun.*, 2002, vol. 3, pp. 1863–1867.

[8] J. Rößler, W. Gerstacker, A. Lampe, and J. Huber, "Matched-filter- and MMSE-based iterative equalization with soft feedback for QPSK transmission," in *Proc. Int. Zurich Seminar Broadband Commun.*, Feb. 2002, pp. 19-1–19-6.

[9] M. Tüchler, R. Koetter, and A. C. Singer, "Turbo equalization: Principles and new results," *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 754–767, May 2002.

[10] R. R. Lopes and J. R. Barry, "Blind iterative channel identification and equalization," in *Proc. IEEE Int. Conf. Commun.*, Oulu, Finland, Jun. 2001, pp. 2256–2260.

[11] A. A. Chang and G. Wornell, "A class of block-iterative equalizers for intersymbol interference cancellation: Fixed channels results," *IEEE Trans. Commun.*, vol. 49, no. 11, pp. 1966–1976, Nov. 2001.

[12] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding," in *Proc. IEEE Int. Conf. Commun.*, May 1993, pp. 1064–1070.

[13] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 40, no. 10, pp. 1727–1737, Oct. 2001.

[14] R. R. Lopes, "Iterative estimation, equalization and decoding," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, GA, 2003.

[15] J. W. M. Bergmans, *Digital Baseband Transmission and Recording*. New York: Springer, 1996.

[16] [Online]. Available: http://spib.rice.edu/spib/data/signals/comm/microwave/chan14a

[17] M. Zimmermann and K. Dostert, "A multipath model for the powerline channel," *IEEE Trans. Commun.*, vol. 50, no. 4, pp. 553–559, Apr. 2002.