Near-minimum bit-error rate equalizer adaptation for PRML systems

Citation for published version (APA):
Riani, J., Beneden, van, S. J. L., Bergmans, J. W. M., & Immink, A. H. J. (2007). Near-minimum bit-error rate equalizer adaptation for PRML systems. IEEE Transactions on Communications, 55(12), 2316-2327. DOI: 10.1109/TCOMM.2007.910693

DOI:
10.1109/TCOMM.2007.910693

Document status and date:
Published: 01/01/2007

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Near-Minimum Bit-Error Rate Equalizer Adaptation for PRML Systems

Jamal Riani, Steven van Beneden, Jan W. M. Bergmans, *Senior Member, IEEE*, and Andre H. J. Immink

***Abstract*—Receivers for partial response maximum-likelihood systems typically use a linear equalizer followed by a Viterbi detector. The equalizer tries to confine the channel intersymbol interferenceto a short span in order to limit the implementation complexity of the Viterbi detector. Equalization is usually made adaptive in order to compensate for channel variations. Conventional adaptation techniques, e.g., LMS, are, in general, suboptimal in terms of bit-error rate (BER). In this paper, we present a new equalizer adaptation algorithm that seeks to minimize the BER at the Viterbi detector output. The algorithm extracts information from the sequenced amplitude margin (SAM) histogram and incorporates a selection mechanism that focuses adaptation on particular data and noise realizations. The selection mechanism is based on the reliability of the add compare select (ACS) operations in the Viterbi detector. From a complexity standpoint, the algorithm is essentially as simple as the conventional LMS algorithm. Moreover, we present a further simplified version of the algorithm that does not require any hardware multiplications. Simulation results, for an idealized optical storage channel, confirm a substantial performance improvement relative to existing adaptation algorithms.**

***Index Terms*—Adaptive equalizers, intersymbol interference, partial response signaling, sequenced amplitude margin (SAM), Viterbi detection.**

I. Introduction

THE OPTIMAL receiver for estimating a data sequence in the presence of intersymbol interference (ISI) and additive Gaussian noise [1] can, generally, not be realized because of its excessive complexity. This fact has led to the development of a variety of suboptimal and lower complexity receivers.

In many practical systems, a linear equalizer is first used to shape the channel symbol response to an acceptably shorter target response. A Viterbi detector (VD), suitable for the target response [2], subsequently estimates the transmitted data sequence. Such systems are known as partial response maximum-likelihood (PRML) systems. PRML systems are widely used in digital recording [3] to combat the extensive ISI, caused by the channel, especially at high recording densities.

Equalization in PRML systems is, usually, made adaptive in order to compensate for channel variations. One of the most popular adaptation methods is based on the MMSE criterion [4]. This method minimizes the power of the error signal, with the error signal being the difference between the actual and the ideal (noiseless) VD input. This minimization is achieved regardless of correlation or data-dependency of the error signal, as caused, for example, by residual ISI (RISI) due to misequalization. However, it is known that RISI or correlated noise can cause considerable bit-error rate (BER) degradation when compared to a system operating with a comparable amount of additive white Gaussian noise (AWGN) and no RISI. Therefore, MMSE equalization does not guarantee, in general, optimum BER performance. To minimize BER, the equalizer must minimize RISI for data patterns that are critical for bit detection and might tolerate more RISI for less critical data patterns. In other words, the effort of equalization must be focused primarily on critical data patterns, by improving their corresponding detection SNR. As far as noise correlation is concerned, the equalizer must seek an appropriate tradeoff between noise correlation and RISI in order to achieve the best BER. These requirements cannot, in general, be fulfilled with MMSE equalization.

Adaptive minimum-BER equalization has already been studied for the case of full response equalization and sample-by-sample detection [5] and decision-feedback equalization [6]. However, in the context of PRML systems, no such studies have been reported. A step toward minimum-BER adaptive equalization was reported in [7] where a new equalizer adaptation criterion was derived from the sequenced amplitude margin (SAM) [8], [9]. The novel idea in [7], known as the LMS SAM (LMSAM) error, is to base equalizer adaptation on minimizing the “variance” of the SAM for particular bit patterns and error events. The error events considered by the LMSAM technique are single-bit errors at transitions in the data. This restriction to single-bit errors makes the LMSAM technique suboptimal for channels where other error events are important. Moreover, basing the equalizer adaptation on minimizing the SAM variance only is, in general, not optimal in terms of the BER, as will be shown in this paper.

This paper presents a new equalizer adaptation algorithm that seeks to minimize the BER. The algorithm incorporates a selection mechanism that focuses equalizer adaptation only on a particular region of the SAM histogram. The selection mechanism is based on the reliability of the Add Compare Select (ACS) operation in the VD. From an implementation standpoint, our algorithm is essentially as simple as the LMS algorithm.

Paper approved by X. Dong, the Editor for Modulation and Signal Design of the IEEE Communications Society. Manuscript received April 28, 2005; revised November 1, 2006 and May 14, 2007. This work was supported by the EU under Project IST-2001-34168 (TwoDOS).

J. Riani and J. W. M. Bergmans are with the Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands (e-mail: j.riani@tue.nl; j.w.m.bergmans@tue.nl.).

S. V. Beneden was with the Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands. He is now with Fortis Bank N.V., 3584 BV Utrecht, The Netherlands (e-mail: s.v.beneden@tue.nl).

A. H. J. Immink was with the Philips Research Laboratories, 5656 AA Eindhoven, The Netherlands. He is now with a Philips Healthcare Incubator, NY (e-mail: andre.immink@philips.com).

Digital Object Identifier 10.1109/TCOMM.2007.910693

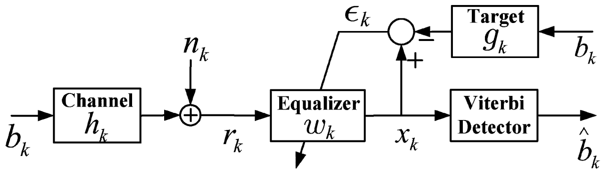

Fig. 1. Discrete-time model of a PRML system.

Moreover, a further simplified version of the algorithm that does not require any multiplications is proposed.

The remainder of this paper is organized as follows. Section II describes the system model and nomenclature. Section III provides analytical steps needed to understand the behavior of the VD as a function of the error signal at its input. This allows us to propose a cost function for equalizer adaptation. Section IV explains the new equalizer adaptation schemes. Simulation results, presented in Section V, show the merits of our algorithm compared to existing ones.

II. System Model and Nomenclature

A discrete-time model of a PRML system is shown in Fig. 1 . A binary sequence $b_k \in \{\pm 1\}$ is transmitted, at a rate $1/T$, over a linear dispersive channel with finite impulse response h_k. The channel output is corrupted by additive zero-mean noise n_k. The reasoning in this paper, concerning equalization, is quite general and does not assume any prior knowledge of the nature of the noise n_k, e.g., the noise n_k is not necessarily Gaussian and can be data-dependent. The received or replay signal r_k is the noisy channel output and is given by

$$r_k = (h * b)_k + n_k$$

where '$*$' denotes convolution. The channel impulse response is, in general, quite long and may be time-varying. For this reason, adaptive partial response (PR) equalization [4] is used in order to transform the channel response to a shorter and well-defined impulse response. The equalizer impulse response w_k is optimized so that the overall impulse response, at its output, is as close as possible to a prescribed short impulse response that we refer to as the target response g_k. The equalizer output x_k serves as input to a VD that is matched to the target response g_k and that produces bit decisions $\widehat{b}_k$. The detector input x_k is ideally equal to the reference signal $(g * b)_k$. However, because of channel noise and RISI, x_k can be written as

$$x_k = (g * b)_k + \epsilon_k$$

where ϵ_k denotes the error signal at the detector input and contains contribution of channel noise and RISI caused by misequalization.

Before proceeding with equalizer adaptation that minimizes BER, let us first understand, in the next section, the dependency of the VD performance on the error signal ϵ_k. This is, then, used in order to derive a practical equalizer adaptation criterion that is directly linked to the BER.

For mathematical convenience, we omit the delays of the different modules and the latency of the bit detector and assume that $\widehat{b}_k = b_k$.

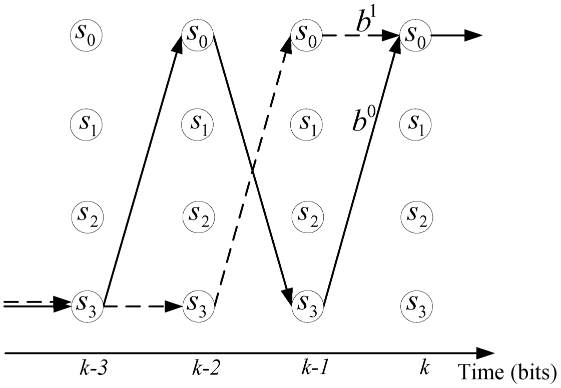

Fig. 2. Example of a 4-state trellis.

III. Derivation of the Adaptation Criterion

The VD in Fig. 1 operates on a trellis that is matched to the target response g_k. Every path in this trellis corresponds to an admissible bit sequence. The detector selects the sequence that leads to the smallest path metric in the trellis [1]. The metric of a bit-sequence a_k is given by the Euclidian metric

$$\mathcal{M}(a) = \sum_i (x_i - (g * a)_i)^2 \tag{1}$$

where this summation is taken over all received symbol indices. Obviously, the metric $\mathcal{M}(a)$ is optimal, in the maximum-likelihood sense, when the error signal ϵ_k is white and Gaussian. Because this is not always the case in practice, the VD employing the Euclidian metric (1) can be suboptimal. However, because the metric $\mathcal{M}(a)$ is widely used in practice, due to its simplicity, we focus in the sequel on this metric. The results of this paper can be extended to other metrics. An example is shown in Section IV-B.

An example of a 4-state trellis is shown in Fig. 2. At time kT, the VD employs, for every state, an ACS operation to select the best path arriving at each state; the other path is discarded. Let us assume for the sake of the argument that the path corresponding to the transmitted bit-sequence b_k arrives at state S_0 at time kT. We denote by b_k^0 and b_k^1 the selected and discarded paths by the ACS operation at state S_0 and time kT. An erroneous ACS decision will occur at time kT when the correct path, corresponding to b_k, is discarded, i.e., when $b^1 = b$. The selected path, in this case, is $b^0 = b + 2e$, where $e = \frac{b^0 - b}{2}$ ($e_k \in \{0, \pm 1\}$) is referred to as the bit-error sequence. This erroneous ACS decision occurs with a probability

$$\Pr(\text{ACS error}|b, e) = \Pr(\mathcal{M}(b + 2e) - \mathcal{M}(b) < 0). \tag{2}$$

The left part of (2) represents the probability that the ACS operation induces a decision error, by discarding the correct path, given the transmitted bit-sequence b_k and an admissible bit-error sequence e_k, i.e., a sequence in $\{0, \pm 1\}$ for which $b_k + 2e_k$ is an admissible bit sequence.

With the assumption of an infinitely-long backtracking depth in the VD, the overall BER is directly related to the probability of ACS errors over all possible data patterns and admissible bit-error sequences. Minimization of the probability of ACS error for a given bit-error sequence leads to minimization of the BER for that specific bit-error sequence, i.e., of the contribution of this sequence to the overall BER.

The variable $S(e) = \mathcal{M}(b+2e) - \mathcal{M}(b)$ is known in literature as the SAM and was first introduced in [8]. Upon invoking (1), $S(e)$ can be written as

$$\begin{aligned} S(e) &= 4\sum_i \left((g*e)_i^2 - (g*e)_i\epsilon_i\right) \\ &= 4(\underline{\delta}_e^T\,\underline{\delta}_e - X_e) \end{aligned} \tag{3}$$

where $\underline{\delta}_e$ is a column vector given by $\underline{\delta}_{e,i} = (g*e)_i$, $\underline{\delta}_e^T\,\underline{\delta}_e$ is the Euclidian weight of the bit-error sequence e_k, and $X_e = \underline{\delta}_e^T\,\underline{\epsilon}$ denotes the correlation between $\delta_{e,k}$ and the error signal ϵ_k. Using (3), (2) can be rewritten as

$$\Pr(\text{ACS error}|b,e) = \Pr(\underline{\delta}_e^T\,\underline{\delta}_e < X_e). \tag{4}$$

In order to minimize (4) for a particular bit-error sequence e_k, optimal equalization must shape ϵ_k, or, equivalently, the variable X_e, such that $\Pr(\underline{\delta}_e^T\,\underline{\delta}_e < X_e)$ is minimized. A first attempt toward this goal is to minimize $E[X_e^2]$ according to the LMSAM algorithm, as suggested, for single-bit errors, in [7]. However, this is not optimal because minimization of $E[X_e^2]$ yields no control on the sign of $E[X_e]$ whereas this sign is of capital importance for $\Pr(\underline{\delta}_e^T\,\underline{\delta}_e < X_e)$.

By way of illustration, we consider in Appendix A the case when the channel noise n_k is additive and Gaussian and study the impact of residual ISI on the SAM. We show mainly two points. First, $E[X_e]$ and $E[X_e^2]$ are both functions of the equalizer response w_k (20), (21). Second, $E[X_e]$ affects (4) differently than the variance $\sigma_{X_e}^2 = E[X_e^2] - E[X_e]^2$ of X_e. The average of X_e, when positive, causes a degradation in effective Euclidian weight of the bit-error sequence e_k. The variance of X_e can be seen as an increase in channel noise power. Thus, minimizing $E[X_e^2]$ is suboptimal because, on the one hand, this does not provide the optimal tradeoff between $E[X_e]$ and σ_{X_e} and, on the other hand, this does not constrain the sign of $E[X_e]$ whereas the latter is of capital importance for (4). This sign tells whether the residual ISI is constructive or destructive in terms of (4).

Because Appendix A assumes the prior knowledge of the channel response and noise characteristics, its results cannot be directly used in the context of adaptive equalization. In order to come up with a simple criterion on X_e, which is directly linked to minimization of (4), we make the following observations:

1) First, an ACS error occurs only when $\underline{\delta}_e^T\,\underline{\delta}_e < X_e$. Therefore, it is natural to consider the values of X_e only in a certain interval of interest, namely, when X_e is higher than a certain threshold around $\underline{\delta}_e^T\,\underline{\delta}_e$.
2) Second, although the distribution of X_e is, in general, not Gaussian, its tail above $\underline{\delta}_e^T\,\underline{\delta}_e$, or, equivalently, the tail of $S(e)$ below zero, can be approximated with a Gaussian tail. This argument has been first used and validated in [9] in order to extract the BER estimates from the SAM distribution. The validation of this argument in [9] was based on both simulated data and experimental replay signals taken from different optical disk systems.

Example 1: In order to provide a simple explanation of the Gaussian tail approximation, let us consider the case where the channel noise n_k has a Gaussian distribution. The error signal ϵ_k can be written as $\epsilon_k = (m*b)_k + u_k$ where $m_k = (w*h)_k - g_k$ and $u_k = (w*n)_k$ is Gaussian as it is a filtered version of a Gaussian noise. The variable X_e, which is written as $X_e = \sum_k (g*e)_k(m*b)_k + \sum_k (g*e)_k u_k$, can, then, be interpreted as a superposition of different Gaussian distributions; one distribution per bit sequence. For a given bit-sequence b_k, the mean of the corresponding Gaussian distribution is given by $\sum_k (g*e)_k(m*b)_k$ and its variance by $\underline{\delta}_e^T R_{uu}\underline{\delta}_e$ where R_{uu} denotes the autocorrelation matrix of u_k. Because the variance of these Gaussian distributions is independent of b_k, the tail of X_e, above $\underline{\delta}_e^T\,\underline{\delta}_e$, is mainly determined by the bit-sequence $\bar{b}$ for which $\sum_k (g*e)_k(m*b)_k$ is the biggest, i.e., $\bar{b} = \arg\max_b \sum_k (g*e)_k(m*b)_k$. This justifies the Gaussian tail approximation on the distribution of X_e. Note that the bit-sequence $\bar{b}$ corresponds to the sequence with the most destructive ISI for the bit-error sequence e_k. ◇

Following the aforementioned observations, we introduce the truncated version of X_e over the interval $]T_e, +\infty[$, where the positive threshold T_e is smaller than $\underline{\delta}_e^T\,\underline{\delta}_e$, i.e., $0 < T_e \leq \delta_e^T\,\underline{\delta}_e$. The truncated version of X_e is denoted by X_e' and is defined as

$$X_e' \doteq X_e \mathbb{1}_{\{X_e > T_e\}} = \begin{cases} X_e & \text{if } X_e > T_e \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where the function $\mathbb{1}_{\{Y\}}$ takes the value 1 if the Boolean variable Y is true and 0 otherwise.

Under the assumption that the tail of the distribution of X_e over $]T_e, +\infty[$ can still be approximated as a tail of a Gaussian, we will show that, for a judicious choice of T_e, $\Pr(\underline{\delta}_e^T\,\underline{\delta}_e < X_e)$ is an increasing function of $E[X_e']$. In other words, increasing $E[X_e']$ leads necessarily to an increase in $\Pr(\text{ACSerror}|b,e)$ and vice versa. In fact, if we denote by μ_e and σ_e^2, respectively, the average and the variance of the Gaussian distribution that fits best the tail of the distribution of X_e over $]T_e, +\infty[$ (see Fig. 3), then one can write

$$\Pr(\underline{\delta}_e^T\,\underline{\delta}_e < X_e) \simeq \mathrm{Q}\left(\frac{\underline{\delta}_e^T\,\underline{\delta}_e - \mu_e}{\sigma_e}\right) \tag{6}$$

where the Q-function is defined as $\mathrm{Q}(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{\frac{-t^2}{2}}dt$. Besides, it can be shown that

$$\begin{aligned} E[X_e'] = \mu_e \mathrm{Q}\left(\frac{T_e - \mu_e}{\sigma_e}\right) \\ + (2\pi)^{-1/2}\sigma_e \exp\left\{-\frac{(T_e-\mu_e)^2}{2\sigma_e^2}\right\}. \end{aligned}$$

This expression can be further simplified, over the SNR range of practical interest, by using the approximation $\mathrm{Q}(x) \simeq (2\pi x^2)^{-1/2}\exp\{-x^2/2\}$ for $x > 2$. This leads to

$$E[X_e'] \simeq T_e \mathrm{Q}\left(\frac{T_e - \mu_e}{\sigma_e}\right). \tag{7}$$

In order to make the argument of the Q-function in (7) proportional to that in (6), an obvious choice of T_e is $T_e = \underline{\delta}_e^T\,\underline{\delta}_e$. However, this choice of T_e implies that X_e' is nonzero only when

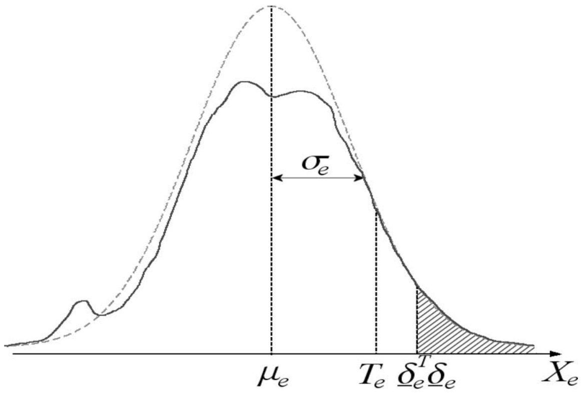

Fig. 3. Conceptual plot of the distribution of X_e (solid). The dashed curve corresponds to the Gaussian fitting of the tail of X_e on $]T_e, +\infty[$. The hashed area corresponds to $\Pr(\underline{\delta}_e^T\,\underline{\delta}_e < X_e)$.

the VD makes a detection error. Accordingly any equalizer adaptation, in this case, can only operate in a data-aided (DA) mode where prior knowledge of the transmitted bits is available. In order to be able to also operate in the decision-directed (DD) mode, where the detected bits are used in the adaptation loop, the threshold T_e has to be taken strictly smaller than $\underline{\delta}_e^T\,\underline{\delta}_e$. To this aim, one can readily show that thresholds T_e of the form

$$T_e = (1-\alpha)\underline{\delta}_e^T\,\underline{\delta}_e + \alpha\mu_e \tag{8}$$

where $\alpha \in [0,1]$, make the argument of the Q-function in (7) proportional to that of (6). In fact, such a choice of T_e leads to

$$\frac{E[X'_e]}{T_e} \simeq \mathrm{Q}\left((1-\alpha)\frac{\underline{\delta}_e^T\,\underline{\delta}_e - \mu_e}{\sigma_e}\right). \tag{9}$$

It is apparent that minimizing (6) is equivalent to minimizing (9). Thus, in order to minimize the BER for a particular bit-error sequence e_k, equalizer adaptation can be based on minimizing the following cost function:

$$\Delta_e = \frac{E[X'_e]}{T_e} \tag{10}$$

where the threshold T_e is given by (8). The value of α is chosen such that the Gaussian tail approximation holds on $]T_e, +\infty[$. Typical values of α are in the interval $[0, 0.5]$. The dependence of T_e on μ_e (8) implies that, in practice, the variables μ_e for the different bit-error sequences must be estimated and adapted. However, because at reasonable SNRs, $\mu_e \simeq E[X_e] = E[\underline{\delta}_e^T\,\underline{\epsilon}] \ll \underline{\delta}_e^T\,\underline{\delta}_e$, one can simply neglect the dependency of T_e on μ_e. In fact, using the Cauchy–Schwartz inequality, one can prove that $E[\underline{\delta}_e^T\,\underline{\epsilon}] \le \sqrt{(\underline{\delta}_e^T\,\underline{\delta}_e)E[\underline{\epsilon}^T\,\underline{\epsilon}]}$, which implies, at reasonable SNRs, that $\mu_e \ll \underline{\delta}_e^T\,\underline{\delta}_e$ because $E[\underline{\epsilon}^T\,\underline{\epsilon}] \ll \underline{\delta}_e^T\,\underline{\delta}_e$. Unless specified otherwise, we fix a value of α and consider the threshold T_e to be equal to $(1-\alpha)\underline{\delta}_e^T\,\underline{\delta}_e$.

Example 2: For the sake of illustration, let us consider the error signal ϵ_k as a zero-mean Gaussian noise signal and denote its autocorrelation matrix by R_ϵ. This is especially true if the residual ISI at the detector input is negligible. For a given bit-error sequence e_k, the variable X_e is, then, Gaussian with a mean $\mu_e = 0$ and a variance $\sigma_e^2 = \underline{\delta}_e^T\,R_\epsilon\underline{\delta}_e$. The threshold T_e in (8) is, then, given by $T_e = (1-\alpha)\underline{\delta}_e^T\,\underline{\delta}_e$ and one can show, after a few mathematical steps, that (10) boils down to

$$\Delta_e = f\left(\frac{T_e}{\sigma_e}\right)$$

where the function f is given by $f(x) = \frac{1}{\sqrt{2\pi x^2}}\exp\left\{\frac{-x^2}{2}\right\}$.

Because f is a strictly decreasing function for $x > 0$, one concludes that minimizing Δ_e is equivalent to maximizing the ratio $\frac{T_e}{\sigma_e} = (1-\alpha)\frac{\underline{\delta}_e^T\,\underline{\delta}_e}{\sqrt{\underline{\delta}_e^T\,R_\epsilon\,\underline{\delta}_e}}$ which is proportional to the square root of the effective SNR [1]. This example illustrates, once more, that designing an equalizer that minimizes Δ_e is equivalent to maximizing the effective SNR, i.e., minimizing the BER for a given bit-error sequence. ◇

IV. Near-Minimum BER Equalizer Adaptation

In the previous section, a cost function (10), which is directly related to the BER for a given bit-error sequence, was derived. In this section, we employ (10) in order to derive the near-minimum BER (NMBER) equalizer adaptation. The basic idea of the NMBER adaptation is to minimize (10) for all relevant bit-error sequences. The different functions Δ_e for the different bit-error sequences are, then, combined with different weights so as to achieve the best overall BER. For clarity, let us first focus on a given bit-error sequence e_k and develop an adaptive equalization scheme that minimizes (10). The second part of this section combines the different minimizations of the different functions Δ_e such that the overall BER, approximated by its union-bound expression, is optimized.

For a given bit-error sequence e_k, an equalizer adaptation scheme that minimizes (10) can be based on the steepest descent algorithm. This consists of following, at each iteration, the opposite direction of the gradient of Δ_e with respect to the equalizer coefficients. The adaptation of the pth-equalizer tap can be written as follows:

$$w_p^{(k+1)} = w_p^{(k)} - \eta'(e)\left.\frac{\partial\Delta_e}{\partial w_p}\right|_{\underline{w}=\underline{w}^{(k)}} \tag{11}$$

where $w_p^{(k)}$ is the pth-equalizer tap at time kT. The coefficient $\eta'(e)$ denotes the equalizer adaptation constant. Note that this adaptation constant is, in general, dependent on the error-sequence e_k. The reasons for this dependency are explained in the next paragraph. By using (5) and the equality $\frac{\partial\epsilon_i}{\partial w_p} = r_{i-p}$, one can prove that

$$\frac{\partial X'_e}{\partial w_p}\Big|_{\underline{w}=\underline{w}^{(k)}} = \left(\sum_{i\le k}\delta_{e,i}r_{i-p}\right)\mathbb{1}_{\{X_e>T_e\}}. \tag{12}$$

Upon replacing the expectation of X'_e in (10) by its instantaneous realization, (11) can be rewritten as

$$w_p^{(k+1)} = w_p^{(k)} - \eta(e)\left(\underline{\delta}_e^T\,\underline{r}_{k-p}\right)\mathbb{1}_{\{\mathcal{M}(b+2e)-\mathcal{M}(b)<T_h(e)\}} \tag{13}$$

where $\eta(e) = \eta'(e)/T_e$, $\underline{r}_{k-p} = [r_{k-p}, r_{k-p-1}, \ldots]^T$, and $T_h(e) = 4\alpha\underline{\delta}_e^T\,\underline{\delta}_e$ and where the selection condition, i.e., $\mathbb{1}_{\{X_e>T_e\}}$ was rewritten in terms of path metrics in the VD trellis using (3).

Now, if we consider a set of bit-error sequences, the overall BER can be seen as the accumulation of conditional BERs for each bit sequence and admissible bit-error sequence, weighted differently for every bit sequence and bit-error sequence. More precisely, if we assume that transmitted sequences are of length N, then a union bound on the BER can be obtained using Bayes' rule. This is written as

$$\mathrm{BER} \le \sum_{b,e} p(b,e)\frac{H_w(e)}{N}\Pr(\underline{\delta}_e^T\,\underline{\delta}_e < X_e) \tag{14}$$

where the summation is taken over all possible bit sequences b of length N and bit-error sequences e. The probability that a bit-sequence b is transmitted and that e is an admissible bit-error sequence is denoted by $p(b,e)$. The Hamming weight of the bit-error sequence e, i.e., the number of nonzeros in e, is denoted by $H_w(e)$. In order to derive a near-optimal expression of the weights $\eta(e)$, we use the union-bound expression to approximate BER.

Averaging over all bit sequences and admissible bit-error sequences, one can see that the NMBER adaptation in (11) seeks to minimize the total cost function

$$\overline{\Delta} = \sum_{b,e} p(b,e)\eta'(e)\Delta_e. \tag{15}$$

Note that the averaging operation is inherited in the equalizer adaptation loop. If we first consider the case where $\alpha = 0$, then we have $T_e = \underline{\delta}_e^T\,\underline{\delta}_e$ and $\Delta_e = \Pr(\underline{\delta}_e^T\,\underline{\delta}_e < X_e)$ using (6), (9), and (10). It follows that, in order to make the minimization of (15) equivalent to that of the right-hand expression of (14), it is sufficient to take $\eta'(e) = \eta(e)T_e$ to be proportional to $H_w(e)$, or, equivalently,

$$\eta(e) = \eta_0\frac{H_w(e)}{\underline{\delta}_e^T\,\underline{\delta}_e} \tag{16}$$

where η_0 is a constant independent of the bit-error sequence e_k. Therefore, in order to minimize the BER, the minimization of the different cost functions Δ_e should be weighted differently for different bit-error sequences, according to (16). The division by $\underline{\delta}_e^T\,\underline{\delta}_e$ in (16) can be omitted in practice because the dominant bit-error sequences have approximately similar Euclidian weights, which are close to the minimal Euclidian weight.

When $\alpha > 0$, then the expression of $\eta(e)$ given in (16) becomes suboptimal in general. However, from our simulations, no noticeable improvement in the BER was provided by further optimization of $\eta(e)$. For this reason, we consider the expression of $\eta(e)$, given by (16), in the sequel.

The overall adaptation of the $p\mathrm{th}$-equalizer tap value is depicted in Fig. 4. At every clock cycle kT, an ACS operation is employed at every state. At every state, two quantities are derived. First, the difference in path metrics between the selected and the discarded paths is taken. Second, a bit-error sequence e_k is derived as the bitwise difference between the two sequences corresponding to the discarded and the selected paths. The bit-error sequence e_k, taken from the state where the best path ends, is used to compute the vector $\underline{\delta}_e = [(g*e)_k, (g*e)_{k-1}, \ldots (g*e)_{k-L}]^T$, where the integer value L depends on the maximum length of relevant bit-error sequences. In the sequel, we simply fix L to the backtracking depth of the VD. The equalizer adaptation is enabled only when the difference in path metrics is smaller than $T_h(e) = 4\alpha\underline{\delta}_e^T\,\underline{\delta}_e$. For simplicity, one can fix $T_h(e)$ to

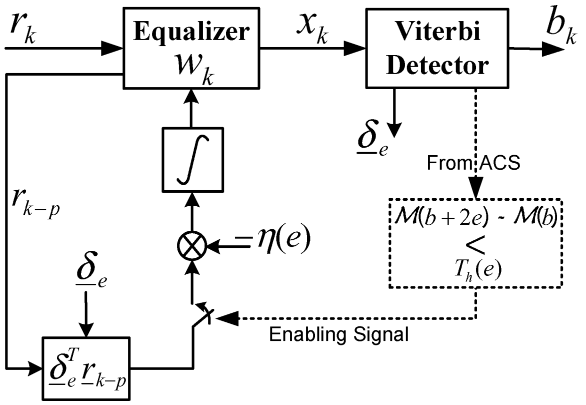

Fig. 4. NMBER adaptation. Only the adaptation of the $p\mathrm{th}$-equalizer tap is shown.

$$T_h(e) = T_h = 4\alpha\min_e \underline{\delta}_e^T\,\underline{\delta}_e$$

without any significant loss in performance. When the adaptation is enabled, the scalar product of the vector $\underline{\delta}_e$ with the equalizer input vector $\underline{r}_{k-p} = [r_{k-p}, r_{k-p-1}, \ldots r_{k-p-L}]^T$ is computed, scaled with $-\eta(e)$, and then passed to an ideal discrete-time integrator that produces the updated $p\mathrm{th}$-equalizer tap value.

A geometrical interpretation of the NMBER algorithm, which provides an intuitive explanation, is given in Appendix B.

A. Efficient Realization of Near-Minimum BER Equalizer Adaptation

In Fig. 4, the scalar product operation $\underline{\delta}_e^T\,\underline{r}_{k-p}$ can be interpreted as focusing equalizer adaptation on the frequency region that is of interest for the bit-error sequence e_k. The amplitude response of g_k in the calculation of $\underline{\delta}_e^T\,\underline{r}_{k-p}$ can be interpreted as only a modification of the adaptation open-loop gain per frequency. Therefore, one can replace, in $\underline{\delta}_e^T\,\underline{r}_{k-p}$, g_k by any response g'_k that has the same phase response as g_k. This DOF in the choice of the amplitude response of g'_k can be used to further simplify the NMBER algorithm. In order to illustrate the principle, we consider optical recording and magnetic recording systems. Any other system can be treated similarly.

Because target responses for optical recording and perpendicular magnetic recording systems are, often, symmetric, a simple response $g'(z) = z^{-D_g}$, where D_g denotes the delay in bits of the target response g_k, can be used to compute $\underline{\delta}_e^T\,\underline{r}_{k-p}$. For longitudinal magnetic recording systems, the target response is antisymmetric and is of the form $g(z) = (1-z^{-1})(1+z^{-1})^n$, where $n = 1$, $n = 2$, or $n = 3$, corresponding to PR4, EPR4, and E2PR4 classes of targets. In this case, the response $g'(z) = (1-z^{-1})z^{-n/2}$ if n is even and $g'(z) = (1-z^{-2})z^{-(n-1)/2}$ if n is odd captures the phase response of the target response $g(z)$. This choice of $g'(z)$ can, thus, be used to compute $\underline{\delta}_e^T\,\underline{r}_{k-p}$. For the sake of clarity, let us focus in the sequel only on optical recording and perpendicular magnetic recording channels. The simplified NMBER (SNMBER) equalizer adaptation rule is, then, obtained by replacing in (13) g_k by $g'_k = \delta(k - D_g)$. This can be written as

$$w_p^{(k+1)} = w_p^{(k)} - \eta(e)\left(\underline{e}_{k-D_g}^T \underline{r}_{k-p}\right) \mathbb{1}_{\{\mathcal{M}(b+2e)-\mathcal{M}(b)<T_h\}} \tag{17}$$

where D_g is the delay of the target response g_k and $\underline{e}_{k-D_g}^T = [e_{k-D_g}, e_{k-D_g-1}, \ldots, e_{k-D_g-L}]$. This adaptation algorithm presents the advantage of further improved efficiency. In fact, because, in practice, relevant bit-error sequences span only few bits, the scalar products with $\underline{e}$ can be realized with only few additions. As an example, single bit-errors are given by $e = \pm[1, 0, 0]$, the simplified equalizer update boils down, except for the selection mechanism, to $\underline{e}_{k-D_g}^T \underline{r}_{k-p} = \pm r_{k-p+D_g}$. In the case of a double-bit error, given by $e = [1, 0, -1]$, the equalizer update is simply given by $\underline{e}_{k-D_g}^T \underline{r}_{k-p} = r_{k-p+D_g} - r_{k-p-2+D_g}$.

B. Extension of the NMBER Algorithm to NPML Systems

Noise-predictive maximum-likelihood (NPML) detectors arise by imbedding a noise prediction/whitening process into the branch metric computation of the Viterbi detector [10], [11]. This boils down to modifying the path metric in (1) by replacing the target response by $g'_k = g_k - \sum_{i=1}^{M} p_i g_{k-i}$ and the detector input by $y_k = x_k - \sum_{i=1}^{M} p_i x_{k-i}$, where p_i denotes an M-tap noise prediction filter. The NPML path metric becomes $\mathcal{M}'(a) = \sum_i (y_i - (g' * a)_i)^2$. Therefore, the NMBER algorithm, in this case, can be derived by simple analogy to the PRML case. One can check that the NMBER adaptation for NPML systems can be obtained by simply replacing in (13) g_k by g'_k and applying a whitening filter to the delayed equalizer input, i.e., by replacing in (13) r_{k-p} by $r'_{k-p} = r_{k-p} - \sum_{i=1}^{M} p_i r_{k-p-i}$. The equalizer adaptation rule then becomes

$$w_p^{(k+1)} = w_p^{(k)} - \eta(e)\left(\underline{\delta}'^T_e \underline{r}'_{k-p}\right) \mathbb{1}_{\{\mathcal{M}'(b+2e)-\mathcal{M}'(b)<T_h(e)\}}$$

where $\underline{\delta}'_{e,i} = (g' * e)_i$ and $\underline{r}'_{k-p} = [r'_{k-p}, r'_{k-p-1}, \ldots, r'_{k-p-L}]^T$.

V. Simulation Results

By way of illustration, we consider an idealized optical storage channel according to the Braat–Hopkins model [12]

$$H(f) = \begin{cases} \frac{\sin(\pi f T)}{\pi f T}\left(\cos^{-1}|\frac{f}{f_c}| - \frac{f}{f_c}\sqrt{1-(\frac{f}{f_c})^2}\right), & |f| < f_c \\ 0, & |f| \geq f_c \end{cases}$$

where f_c denotes the normalized optical cutoff frequency. Data b_k is taken to be run-length-limited [13] with run-length parameters $(d, k) = (1, 7)$. The channel noise is AWG with a variance σ_n^2. Channel SNR is defined as $\mathrm{SNR} = \sum_k h_k^2/\sigma_n^2$. The normalized cutoff frequency f_c of an optical recording channel depends on the laser wavelength λ, the numerical aperture (NA) of the objective lens, and the channel bit-length T_{bit} and is given by $f_c = \frac{2\mathrm{NA}}{\lambda} T_{\mathrm{bit}}$. We use here the Blu-ray optical parameters, i.e., $\mathrm{NA} = 0.85$, a laser wavelength $\lambda = 405$ nm, and a track pitch of 320 nm [14]. We consider two different disk capacities that are 23 GB and 30 GB on a single-layer 12-cm disk. The corresponding channel bit-lengths are, respectively, $T_{\mathrm{bit}} = 81$ nm and $T_{\mathrm{bit}} = 62$ nm and the resulting normalized cutoff frequencies are, respectively, $f_c = 0.34$ and $f_c = 0.26$. The comparison of the NMBER with respect to the LMS algorithm is done at both capacities. To compare the NMBER and the LMSAM algorithms, the 30-GB channel is considered where a more pronounced improvement can be pointed out. To allow fair comparison between the different adaptation algorithms, all schemes are run first in the DA mode where the prior knowledge of the transmitted bit sequence is used in all adaptation loops. For LMS this is used to extract the error signal ϵ_k and for the NMBER and the SNMBER it is used to select the state that corresponds to the correct bits from where to extract the bit-error sequence e_k. Simulation results of NMBER performances in the DD mode are then shown at the end of this section.

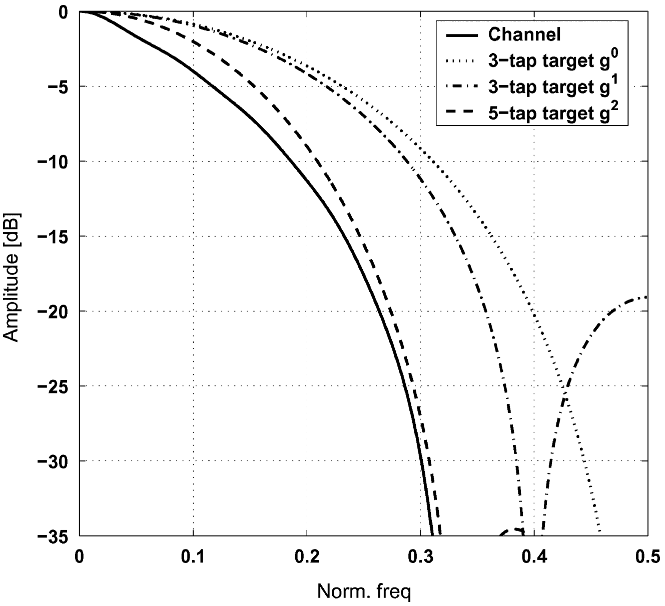

FIg. 5. Amplitude–frequency of idealized optical channel having a normalized cutoff $f_c = 0.34$, 3-tap targets $g^0 = [1, 2, 1]$ and $g^1 = [1, 1.6, 1]$ and 5-tap target $g^2 = [0.17, 0.5, 0.67, 0.5, 0.17]$. For clarity of the plot, the different targets are normalized to have the same DC.

In order to demonstrate the benefits gained by employing the NMBER equalizer adaptation over the conventional LMS adaptation, three target responses are considered. The first one is a 3-tap target response with integer coefficients given by $g^0 = [1, 2, 1]$. The second one, $g^1 = [1, 1.6, 1]$, provides a better match to the channel response. A 4-state VD is employed for g^0 and g^1. The third target response is a 5-tap response given by $g^2 = [0.17, 0.5, 0.67, 0.5, 0.17]$. Because of the $d = 1$ constraint that excludes some bit patterns, e.g., $(+ + - +)$, the number of states in the Viterbi trellis for g^2 is equal to 10. The response of g^2 approximates the in-band characteristics and cutoff frequency of the channel quite well. Amplitude responses of $h(t)$, g_k^0, g_k^1, and g_k^2 are depicted in Fig. 5.

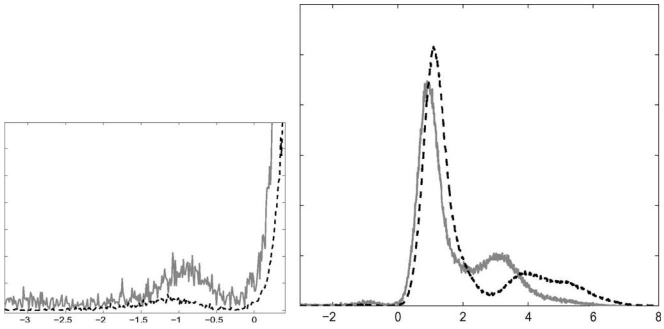

Fig. 6. SAM distribution, at $\mathrm{SNR} = 13$ dB, is shown in the right plot for LMS adaptation (solid) and NMBER adaptation (dashed). A zoom of the SAM histogram around zero is shown in the left plot.

To illustrate the concept of the NMBER adaptation, Fig. 6 shows the SAM histograms for both LMS and NMBER adaptations using the target response g^1. The SAM histogram is the accumulation of the different probability distribution functions of $S(e)$ for the different bit-error sequences. The area below the tail of this histogram, below zero, determines the BER. It can be seen that, below zero, the SAM histogram with NMBER adaptation is below the one with LMS adaptation. Moreover, because the SAM distribution on the positive axis is irrelevant for the BER, our adaptation scheme uses this DOF and does not spend any equalization effort there.

For the 23-GB channel, Fig. 7 shows the simulated BER as a function of the SNR for different targets and adaptation algorithms. The equalizer length N_w is fixed to 9 and $\alpha = 0.4$.

For the target response g^0, Fig. 7 shows that, on the one hand, the NMBER algorithm outperforms the LMS algorithm by 1.5 dB at $\mathrm{BER} = 10^{-5}$. On the other hand, the simplified algorithm SNMBER is indistinguishable, in terms of BER, from the NMBER algorithm. For the target response g^1, the NMBERalgorithm outperforms LMS by 0.6 dB. Moreover, whereas with the latter, the difference in SNR between g^0 and g^1 is ∼1 dB, it is reduced to less than 0.1 dB using the NMBER algorithm. The SNR difference between the two targets in the case of LMS is explained by the fact that g^1 is better matched to the channel than g^0 in the in-band frequencies, i.e., for $f < f_c$.

The 5-tap target response g^2 presents a good match to the channel response, as shown in Fig. 5. For this reason, the LMS adaptation is already very close to optimal in the case of additive white noise. In this case, the NMBER algorithm is practically identical to its LMS counterpart over the whole SNR range. In addition, using LMS, the 3-tap target g^1 presents a loss in SNR of 1 dB compared to the 5-tap target g^2. This gap in SNR between g^1 and g^2 is reduced to only 0.4 dB using the NMBER algorithm. Such improvement in SNR for short target responses, i.e., less states in the VD trellis, makes the NMBER algorithm very attractive for practical systems.

For the 30-GB channel, Fig. 8 shows the simulated BER as a function of the SNR for different targets and adaptation algorithms. The parameter α is fixed here to $\alpha = 0.3$. Fig. 8 shows clearly that, as density increases, the short-length target responses g^0 and g^1 become completely impractical using the LMS algorithm. Nevertheless, using the NMBER algorithm allows significant performance improvements for these short target responses. This improvement amounts to 3.4 dB for g^1 and to even more for g^0. However, because of their short length, g^0 and g^1 still lag few decibells behind the 5-tap target response g^2. Furthermore, for the target g^2, the NMBER allows an improvement of 1.2 dB in the SNR with respect to the LMS algorithm.

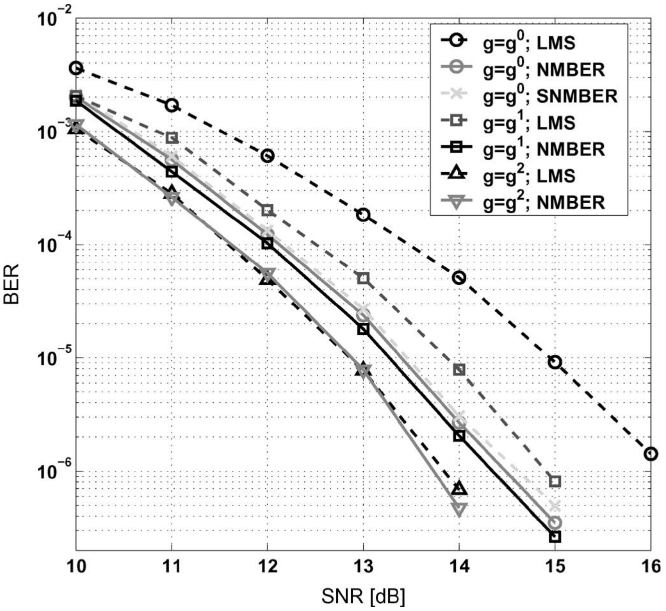

Fig. 7. Simulated BER versus SNR for the different target responses and adaptation schemes at a disk capacity of 23 GB.

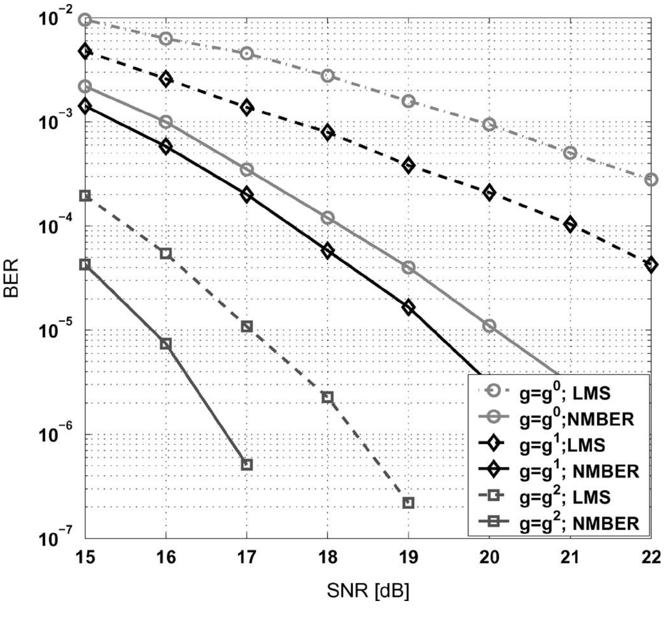

Fig. 8. Simulated BER versus SNR for the different target responses and adaptation schemes at a disk capacity of 30 GB.

It is apparent from Figs. 7 and Figs. 8 that the NMBER algorithm can be very useful in practice. First, in order to limit detection complexity, which grows exponentially with the target length, short target responses are preferably employed. For

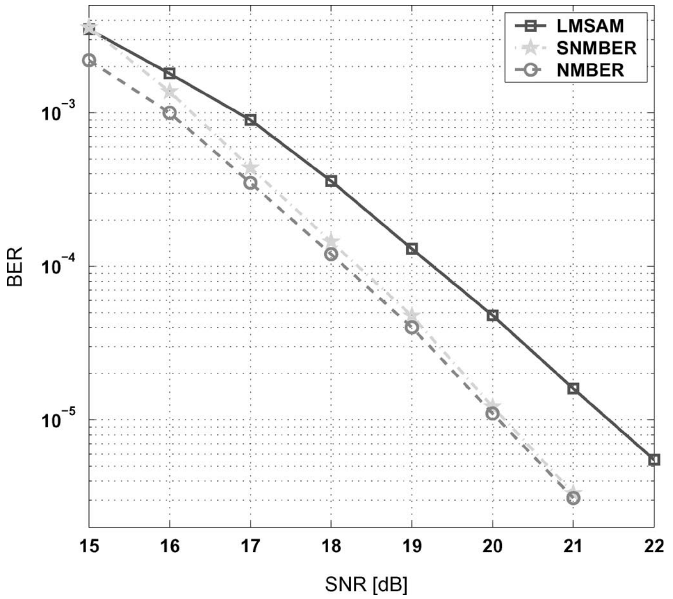

Fig. 9. Simulated BER versus SNR for $g = g^0$ and the different adaptation schemes at a disk capacity of 30 GB.

these targets, the LMS adaptation becomes suboptimal and the NMBER adaptation allows significant performance improvements. Second, at a given complexity, i.e., target length, the SNR improvement of the NMBER equalization with respect to LMS increases with storage density. This should help to achieve higher storage densities without sacrificing complexity.

Next, also the LMSAM is taken into account. The LMSAM scans the data for particular patterns and adapts the equalizer in order to minimize $E[X_e^2]$ for single-bit errors at data transitions. However, as storage capacity increases, other error events, e.g., the double-bit errors $e = \pm[1, 0, -1]$, become substantial. The difference in predetection SNR between the LMSAM and the NMBER then becomes more pronounced. In order to illustrate the suboptimality of the LMSAM algorithm, Fig. 9 shows simulated BER as a function of SNR for the target response g^0 at a disk capacity of 30 GB and $N_w = 9$. The LMSAM algorithm is implemented in the DA mode where the transmitted data is scanned for the patterns $(- - + + +)$, $(- - - + +)$, $(+ + + - -)$, and $(+ + - - -)$. LMSAM equalizer adaptation is implemented, as explained in [7]. For NMBER and SNMBER adaptations, α is taken to be equal to 0.3. Fig. 3 shows that the LMSAM algorithm yields a loss of 1.4 dB compared to the NMBER or the SNMBER algorithm at the capacity of 30 GB. This loss will increase at higher storage capacities.

A. Stability and Convergence Behavior of the NMBER Algorithm

Because of the nonlinear and selective nature of the NMBER algorithm, a theoretical analysis of its stability and convergence behavior is quite fastidious. The convergence behavior of the NMBER algorithm depends on the adaptation constant η_0 and on the threshold T_h. The higher the threshold T_h, the more frequent the NMBER adaptation is enabled and the smaller η_o should be taken into consideration in order to ensure convergence of the algorithm. In order to highlight the dependence of the NMBER performance as a function of η_0, Fig. 10 shows the BER as a function of η_0 for the 30-GB channel at different SNR values for the target responses $g = g^0$ and $g = g^2$. The threshold, or, equivalently, α, is optimized to achieve the best BER for the smallest value of η_0. Fig. 10 illustrates that the performance of the NMBER algorithm is basically independent of η_0 if this latter is smaller than a given value $\eta_{\max}$ ($\approx 10^{-3}$, in this case) and that if $\eta_0 > \eta_{\max}$, the NMBER algorithm can become unstable.

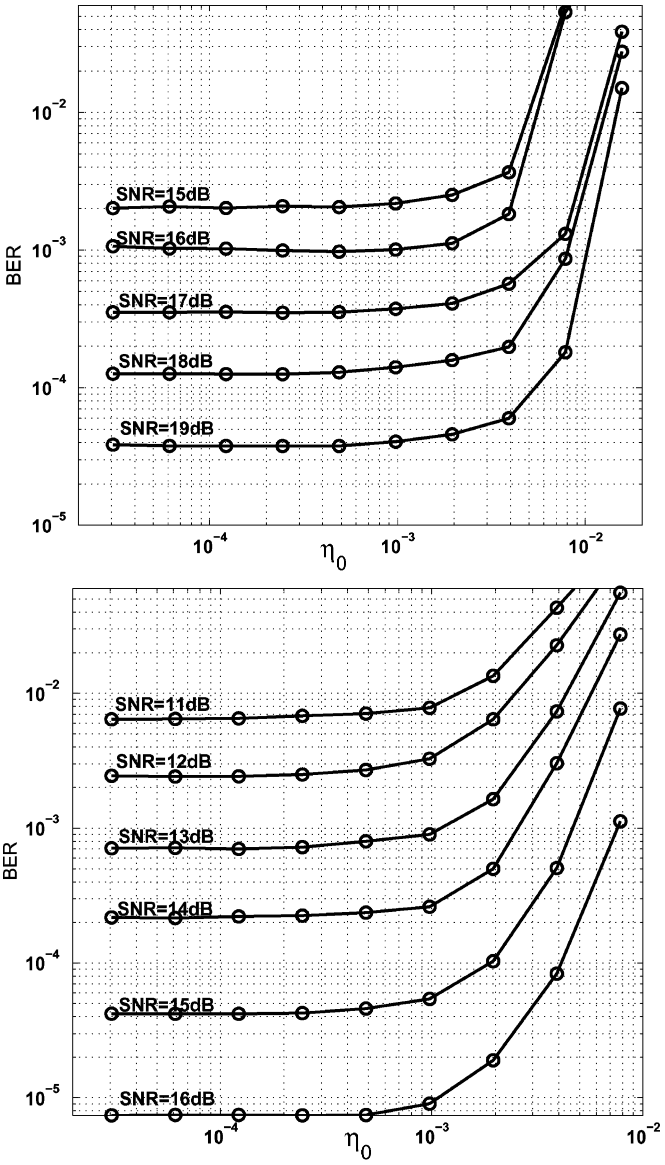

Fig. 10. Simulated BER versus η_0 at a capacity of 30 GB and different SNR values for the target responses $g = g^0$ (upper plot) and $g = g^2$ (lower plot).

The NMBER performance independence of η_0 for $\eta_0 < \eta_{\max}$ and the independence of $\eta_{\max}$ on SNR make the choice of the adaptation constant rather easy in practice. In fact, as optical storage channels have a well-known behavior [12], the value of η_0 for a practical system can be simply precomputed and optimized based on numerical simulations.

B. Behavior of the NMBER Algorithm in the Decision-Directed Mode

The previous simulation results were conducted in the DA mode where the prior knowledge of the transmitted bits was used to extract the necessary control signals for the different algorithms. In many practical systems, prior knowledge about the transmitted bits is not available and (preliminary) VD decisions have to be used instead, i.e., the scheme must be run in the DD mode.

In the DD mode, the choice of α is crucial. In fact, if $\alpha \approx 0$, then the NMBER algorithm will mainly adapt on wrong decisions, which causes the algorithm to diverge. From this perspective, α has to be as high as possible to minimize the probability of adapting on wrong decisions. However, in order to limit BER degradations, α has to be chosen as small as possible such that the Gaussian tail approximation holds. Therefore, α must realize a tradeoff between these two criteria.

To implement the NMBER algorithm in the DD mode, a bit-error sequence and a Boolean variable need to be stored at every state of the trellis up to the decision backtracking depth L. The Boolean variable tells whether the difference in path metrics between the selected and discarded paths by the ACS unit is smaller or bigger than the threshold T_h. At every clock cycle, a bit decision is taken from the VD trellis, at a decoding state, following a selected path at a depth L. The decoding state is also used to extract a bit-error sequence and one Boolean variable. The equalizer adaptation is, then, performed, according to Fig. 4, where the equalizer input r_{k-p} is delayed to compensate for the backtracking delay prior to correlation with δ_e.

Fig. 11 shows the simulated BER for the target responses $g = g^0$ and $g = g^2$ and, respectively, the 23- and 30-GB channels, using the NMBER adaptation in both DA and DD modes. This shows that the performance of the NMBER adaptation in the DD mode is within a fraction of a decibel from its DA counterpart, which proves the practical value of the NMBER algorithm. The SNMBER algorithm has a similar behavior. The performance degradation of the DD mode, compared to the DA mode, increases with storage density, as illustrated in Fig. 11. This is not surprising as system sensitivity increases with density [4], i.e., performance becomes more sensitive to small system parameter deviations.

Remark: In a practical optical storage system, choosing the threshold to be very small, can cause serious problems to the NMBER algorithm. In fact, small values of the difference in VD path metrics can be caused, for example, by media defects, scratches, or fingerprints. Adapting the equalizer when these artifacts occur, will cause the NMBER algorithm to diverge. A simple remedy to this issue is to add a second smaller threshold $T_{h2} < T_h$ and freeze the NMBER adaptation when the VD path metrics difference is smaller than T_{h2}. This threshold should serve also to freeze all adaptation loops, e.g., DC, AGC, PLL, to prevent them from divergence.

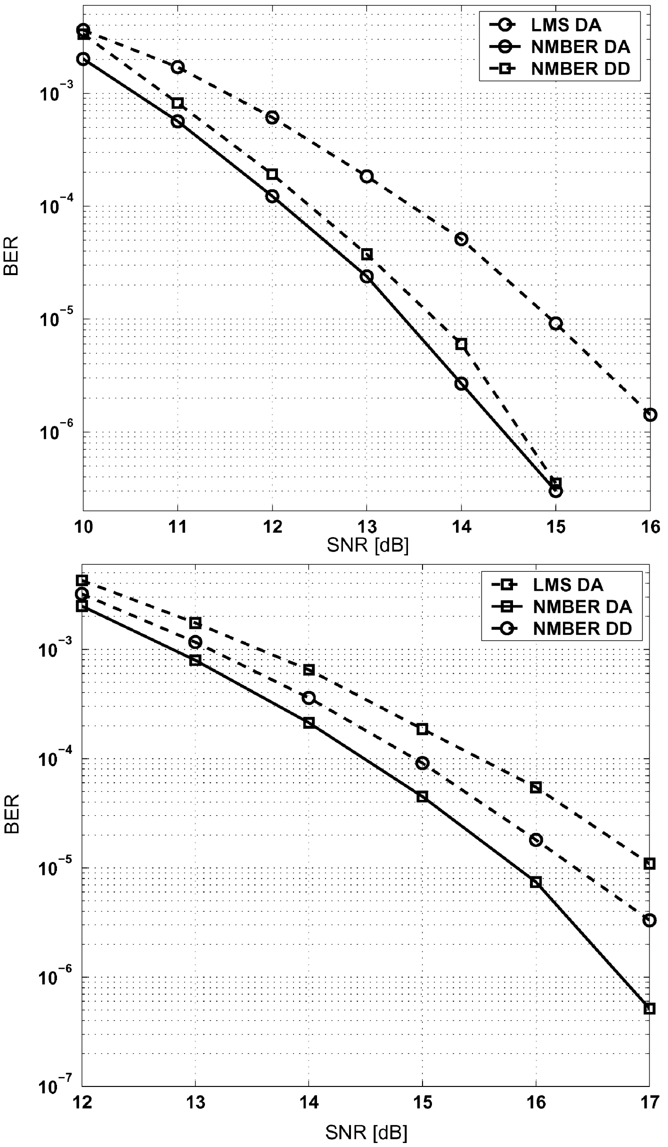

Fig. 11. Simulated BER versus SNR using the NMBER algorithm in both the DA and DD modes for $g = g^0$ and the 23-GB channel (upper plot; $\alpha = 0.4$) and for $g = g^2$ and the 30-GB channel (lower plot; $\alpha = 0.3$). As a basis of reference, the LMS performance in DA mode is also shown.

VI. Conclusion

A new equalizer adaptation scheme has been proposed for PRML systems. This new scheme seeks to minimize directly the BER. Based on an analysis of Viterbi detection performance, we highlighted a practical cost function for equalizer adaptation. This function was used to realize a remarkably simple equalizer adaptation scheme. The proposed scheme incorporates a selection mechanism that enables equalizer adaptation only if the difference in path metrics, between selected and discarded paths from the Viterbi trellis, is smaller than a prescribed threshold. The actual version of the new adaptation scheme is essentially as simple as the LMS. A simplified scheme that allows further improved efficiency was also presented. Because of the selection mechanism, the proposed schemes present an advantage in terms of power consumption, especially for long equalizers.

Simulation results for an idealized optical storage system showed that our scheme outperforms significantly the existing adaptation schemes, especially at high storage densities or short target response lengths.

Appendix

A. Impact of Residual ISI on the Sequenced Amplitude Margin

In order to develop a better understanding of the impact of residual ISI on the SAM, let us consider the case where the channel noise n_k is data-independent, additive, and Gaussian. In this case, the error signal ϵ_k is composed of two components. The first component is time-invariant and linearly dependent on the bit-sequence b_k, i.e., RISI, and the second one is a data-independent zero-mean and Gaussian noise. For simplicity of the analysis, we assume that the binary data is uncoded. The error signal is given by

$$\epsilon_k = (m * b)_k + u_k \tag{18}$$

where $u_k = (w * n)_k$ denotes the noise component. The RISI component is characterized by the impulse response m_k, where $m_k = (w * h)_k - g_k$.

In order to evaluate $\Pr(\underline{\delta}_e^T \underline{\delta}_e < X_e)$, let us consider a bit-error sequence e_k and compute $E[X_e]$ and $E[X_e^2]$, where the expectations are taken over all possible realizations of u_k and b_k, such that $b + 2e$ is an admissible bit sequence. Plugging (18) in $X_e = \underline{\delta}_e^T \underline{\epsilon}$ and substituting $\delta_{e,k}$ by $(g * e)_k$, we can write

$$X_e = \sum_k (g * e)_k (m * b)_k + \sum_k (g * e)_k u_k. \tag{19}$$

Since u_k is independent of e_k and is zero on average, we have $E[\sum_k (g * e)_k u_k] = 0$. The average of X_e is, then, equal to

$$E[X_e] = E\left[\sum_k (g * e)_k (m * b)_k\right] = \sum_{k,i,j} g_{k-i} m_{k-j} E[e_i b_j].$$

In order to evaluate $E[e_i b_j]$, we introduce the set $I(e)$ of indices i such that $e_i \neq 0$, i.e., ($i \in I(e) \Leftrightarrow e_i \neq 0$). The summation over j in the previous equality is split into two terms depending on $j \in I(e)$ or not

$$\begin{aligned} E[X_e] = &\sum_{k,i,j \in I(e)} g_{k-i} m_{k-j} E[e_i b_j] \\ &+ \sum_{k,i,j \in I(e)} g_{k-i} m_{k-j} E[e_i b_j]. \end{aligned}$$

When $j \in I(e)$, b_j becomes deterministic. In fact, because $b + 2e$ is an admissible bit sequence, the only possibility for b_j, when $e_j \neq 0$, is $b_j = -e_j$. In this case $E[e_i b_j] = -e_i e_j$. However, when $j \in I(e)$, it is easy to prove that $E[e_i b_j] = 0$ because the data is assumed to be uncoded. It follows that

$$E[X_e] = -\sum_{k,i,j \in I(e)} g_{k-i} m_{k-j} e_i e_j.$$

Because $e_j = 0$ for $j \in I(e)$, the previous summation can be taken over all values of j. It is, then, straightforward to show that

$$E[X_e] = -\sum_k (g * e)_k (m * e)_k = -\underline{\delta}_e^T \underline{m}_e \tag{20}$$

where the vector $\underline{m}_e$ is given by $(\underline{m}_e)_k = (m * e)_k = (w * h * e)_k - (g * e)_k$.

In a similar manner as we derived (20), one can prove that $E[X_e^2]$ can be written as follows:

$$E[X_e^2] = (\underline{\delta}_e^T \underline{m}_e)^2 + \underline{\delta}_e^T (M^e + R_{uu}) \underline{\delta}_e \tag{21}$$

where R_{uu} is the autocorrelation matrix of u_k and the matrix M^e is defined by

$$\begin{aligned} M_{k,k'}^e &= \sum_{j \in I(e)} m_{k-j} m_{k'-j} \\ &= (m * m^*)_{k-k'} - \sum_{j \in I(e)} m_{k-j} m_{k'-j}, \end{aligned}$$

where m^* is defined by $m_i^* = m_{-i}$.

Equations (20) and (21) give a closed-form expression of $E[X_e]$ and $E[X_e^2]$. In order to link these quantities to ACS error probabilities, let us assume that the distribution of X_e can be approximated by a Gaussian. This assumption is not valid, in general, because of the data-dependent component of the error signal ϵ_k. However, in the limiting case of a small amount of residual ISI, this approximation is acceptable. Note that the approximation is only used in this portion of the appendix to provide more insights and that the other results of this paper are more general. With this assumption, one can write

$$\Pr(\text{ACS error}|e) \simeq \mathrm{Q}\left(\frac{\underline{\delta}_e^T \underline{\delta}_e + \underline{\delta}_e^T \underline{m}_e}{\sqrt{\underline{\delta}_e^T (M^e + R_{uu}) \underline{\delta}_e}}\right) \tag{22}$$

where $\Pr(\text{ACSerror}|e)$ equals the average of $\Pr(\text{ACSerror}|b, e)$ over all possible bit sequences b_k such that $b + 2e$ is an admissible bit sequence.

The impact on $\Pr(\text{ACSerror}|e)$ of the RISI differs significantly from the impact of the channel noise. The RISI has basically two different impacts. First, compared to the case of $m = 0$, it induces a modification in the nominator of the Q-function argument in (22). We name this nominator the effective Euclidian weight of the bit-error sequence e_k. The effective Euclidian weight can be either bigger or smaller than $\underline{\delta}_e^T \underline{\delta}_e$ (constructive or destructive ISI for the bit-error sequence e_k) depending on the sign of $\underline{\delta}_e^T \underline{m}_e = -E[X_e]$. Second, the denominator of the argument of the Q-function in (22) is also modified. One can check that the matrix M^e is positive and, therefore, the denominator increases when $m \neq 0$ compared to $m = 0$. The impact of M^e in (22) can be seen as an increase in effective channel noise power.

An expression of the effective predetection SNR ρ_{VD} can be extracted from (22)

$$\sqrt{\rho_{\mathrm{VD}}} = \min_e \frac{\underline{\delta}_e^T \underline{\delta}_e + \underline{\delta}_e^T \underline{m}_e}{\sqrt{\underline{\delta}_e^T (M^e + R_{uu}) \underline{\delta}_e}}. \tag{23}$$

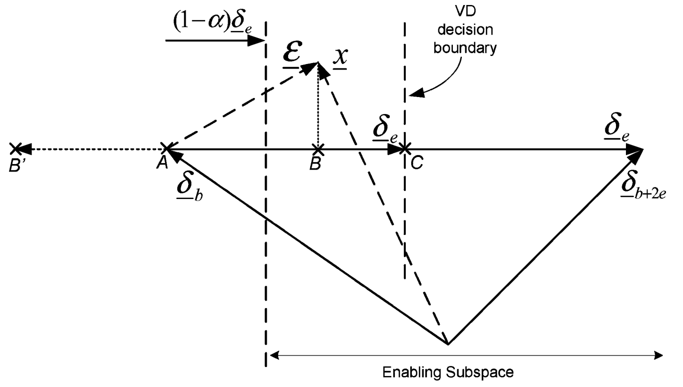

Fig. 12. Geometrical representation of the enabling condition of the NMBER algorithm.

Note that if there is no residual ISI, i.e., $m = 0$, and u_k is white with a variance σ^2, ρ_{VD} boils down to the known expression $\rho_{\mathrm{VD}} = \min_e \frac{\underline{\delta}_e^T \underline{\delta}_e}{\sigma^2}$.

Application to Equalizer Adaptation: Designing the equalizer response to minimize $E[X_e^2]$ (21) does not necessarily minimize $\Pr(\text{ACS error}|e)$ because of two reasons. First, the impact of $\underline{\delta}_e^T \underline{m}_e$ on $\Pr(\text{ACS error}|e)$ is different than that of $\underline{\delta}_e^T (M^e + R_{uu})\underline{\delta}_e$, as explained earlier. Thus, minimizing (21) would not, in general, be optimal because this does not provide the optimal tradeoff between $\underline{\delta}_e^T \underline{m}_e$ and $\underline{\delta}_e^T (M^e + R_{uu})\underline{\delta}_e$. Second, and more important, minimizing $E[X_e^2]$ does not provide any constraint on the sign of $E[X_e]$, i.e., the opposite sign of $\underline{\delta}_e^T \underline{m}_e$, whereas it has been shown that this sign is of capital importance for $\Pr(\text{ACS error}|e)$. We conclude that minimizing $E[X_e^2]$, as suggested in [7], is not optimal in general. Simulation results of Fig. 9 confirm this conclusion.

B. Geometrical Interpretation of the NMBER Algorithm

In order to develop an intuitive understanding of the NMBER algorithm, let us collect the received samples r_k in a vector $\underline{r} = [r_0, \ldots, r_{N-1}]^T$ that we call the received vector. We denote by $\underline{x} = [x_0, \ldots, x_{N-1}]^T$ the column vector of equalized samples $x_k = (w * r)_k$. For simplicity, let us focus on one admissible bit-error sequence e_k. This means that we consider for detection only the two sequences b_k and $(b + 2e)_k$. The VD will decide for the bit-sequence b_k if and only if the vector $\underline{x}$ is closer to $\underline{\delta}_b$ than to the vector $\underline{\delta}_{b+2e}$, where $\underline{\delta}_a = [(g * a)_0, \ldots, (g * a)_{N_b-1}]^T$ for a bit-sequence a_k (see Fig. 12). The distance between two vectors is computed using the L2-norm given by: $\|\underline{X}\|^2 = \underline{X}^T \underline{X}$. Fig. 12 shows also the vector $\underline{\delta}_e = \frac{1}{2}(\underline{\delta}_{b+2e} - \underline{\delta}_b)$ and the boundary decision of the VD. Let us then see what happens to the vector $\underline{x}$ after the NMBER equalizer adaptation. For this purpose, let us assume that we receive a vector $\underline{r}$ and that the NMBER equalizer adaptation is enabled. The same vector $\underline{r}$ is assumed to be received again after equalizer adaptation.

First of all, one needs to note that what matters for detection is the orthogonal projection of $\underline{\epsilon} = \underline{x} - \underline{\delta}_b$ over the vector $\underline{\delta}_e$, i.e. $\overline{AB} = \underline{\delta}_e^T \underline{\epsilon}$.

The NMBER adaptation is enabled when $\underline{\delta}_e^T \underline{\epsilon} > (1 - \alpha)\underline{\delta}_e^T \underline{\delta}_e$. This defines an enabling subspace, as shown in Fig. 12. When the vector $\underline{x}$ falls in the enabling subspace, the adaptation is enabled and the equalizer tap values are changed according to (13). A correction response ∂w, given by $\partial w_p = -\eta(e)\underline{\delta}_e^T \underline{r}_{k-p}$, is, then, added to the equalizer response. After adaptation and reception of the same vector $\underline{r}$, the vector $\underline{x}$ will change with $\underline{\partial x}$ and, more importantly, its orthogonal projection on $\underline{\delta}_e$ changes as follows:

$$\partial\overline{AB} = \underline{\delta}_e^T \underline{\partial x}.$$

Using the fact that $\partial x_k = \sum_p \partial w_p r_{k-p}$ and that $\partial w_p = -\eta(e) \sum_k \delta_{e,k} r_{k-p}$, one can easily prove that

$$\partial\overline{AB} = -\eta(e) \sum_p (\underline{\delta}_e^T \underline{r}_{k-p})^2 \leq 0. \tag{24}$$

It is, then, visible that the NMBER adaptation tries to shift the vector $\underline{x}$ outside the enabling subspace and as far as possible from the VD decision boundary so as to increase detection reliability (reliability can be seen, here, as the distance BC in Fig. 12, between $\underline{x}$ and the VD decision boundary). When the vector $\underline{x}$ falls outside the enabling subspace, the VD will output the bit-sequence b_k with a high reliability. In this case, the NMBER equalizer adaptation is disabled. However, because the LMSAM minimizes $E[\underline{\delta}_e^T \underline{\epsilon}^2]$ (for single-bit errors), it does not make any distinction, in Fig. 12, between the point B $(\underline{\delta}_e^T \underline{\epsilon} > 0)$ and its mirror B' with respect to A $(\underline{\delta}_e^T \underline{\epsilon} < 0)$ whereas these points correspond to completely different reliabilities.

Compared to the LMSAM or the LMS, the NMBER algorithm does not spend equalization effort when this does not improve detection reliability and, moreover, it is clear from (24) that when the NMBER adaptation is enabled, it always acts toward improved reliability.

References

[1] G. D. Forney, "Maximum-Likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inf. Theory*, vol. IT-18, no. 3, pp. 363–378, May 1972.

[2] S. U. H. Qureshi and E. E. Newhall, "An adaptive receiver for data transmission over time-dispersive channels," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 4, pp. 363–378, Jul. 1973.

[3] R. D. Cideciyan, F. Dolivo, R. Hermann, W. Hirt, and W. Schott, "A PRML system for digital magnetic recording," *IEEE J. Sel. Areas Commun.*, vol. 10, no. 1, pp. 38–56, Jan. 1992.

[4] J. W. M. Bergmans, *Digital Baseband Transmission and Recordin*. Norwell, MA: Kluwer Academic, 1996.

[5] C. Yeh and J. Barry, "Adaptive minimum bit-error rate equalization for binary signaling," *IEEE Trans. Commun.*, vol. 48, no. 7, pp. 1226–1235, Jul. 2000.

[6] S. Chen, E. Chng, B. Mulgrew, and G. Gibson, "Minimum-BER linear-combiner DFE," in *Proc. Int. Conf. Commun.*, 1996, pp. 1173–1177.

[7] T. Okumura, J. Akiyama, S. Maeda, T. Yamaguchi, and A. Takahashi, "New adaptive equalization method for partial response maximum likelihood system optimizing error rate performance," *Jpn. J. Appl. Phys.*, vol. 43, no. 10, pp. 7097–7101, 2004.

[8] T. Perkins and Z. A. Keirn, "A window-margin-like procedure for evaluating PRML channel performance," *IEEE Trans. Magn.*, vol. 31, no. 2, pp. 1109–1114, Mar. 1995.

[9] T. Okumura, T. Numata, J. Akiyama, S. Maeda, T. Yamaguchi, and A. Takahashi, "Method for evaluating partial response maximum likelihood system performance using sequenced amplitude margin," *Jpn. J. Appl. Phys.*, vol. 41, no. 3B, pp. 1783–1784, Mar. 2002.

[10] E. Eleftheriou and W. Hirt, "Noise-predictive maximum-likelihood (NPML) detection for the magnetic recording channel," in *Proc. IEEE Conf. Rec., ICC'96*, Jun., vol. 1, pp. 23–27.

[11] J. D. Coker, E. Eleftheriou, R. L. Galbraith, and W. Hirt, "Noise-predictive maximum likelihood (NPML) detection," *IEEE Trans. Magn.*, vol. 34, no. 1, pp. 110–117, Jan. 1998.

[12] A. Huijser, J. Pasman, G. van Rosmalen, G. Bouwhuis, J. Braat, and K. S. Immink, *Principles of Optical Disc Systems*. Bristol, UK: Adam Hilger, 1985.

[13] K. A. S. Immink, *Codes for Mass Data Storage Systems.*. Rotterdam, The Netherlands: Shannon Foundation, 1999.

[14] S. Furumiya, S. Kobayashi, B. Stek, H. Ishibashi, T. Yamagamim, and K. Schep, "Wobble-address format of the Blu-ray Disc," in *Proc. Int. Symp. Opt. Mem. Opt. Data Storage Top. Meeting*, Jul. 2002, pp. 266–268.

Jamal Riani was born in Tetouan, Morocco, in 1977. He received the degrees in engineering from the Ecole Polytechnique, Palaiseau, France, in 1999, and the Ecole National Supéieure de Télécommunications (ENST), Paris, France, in 2001. He is currently working toward the Ph.D. degree at the Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands.

His current research interests include signal processing for digital transmission and recording systems.

Steven van Beneden received the degree of Burgerlijk Electrotechnisch Ingenieur from the Katholieke Universiteit Leuven, Leuven, Belgium, in 2001, and the Post-Master in technological design of ICT-based systems from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2003. From 2003 to 2006, he was a Ph.D. student at the Eindhoven University of Technology, where he was engaged in research on signal processing techniques for digital storage systems.

Since December 2006, he is a fixed-income Quantitative Analyst at Fortis Bank, Utrecht, The Netherlands. His current research interests include the development of interest rate models and algorithmic trading systems.

Jan W. M. Bergmans (M'85–SM'91) received the degree of Elektrotechnisch Ingenieur (*cum laude*), in 1982 and the Ph.D. degree in electrical engineering, in 1987, both from Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands.

From 1982 to 1999, he was with Philips Research Laboratories, Eindhoven, where he was engaged in research on signal-processing techniques and integrated circuit (IC) architectures for digital transmission and recording systems. In 1988 and 1989, he was an Exchange Researcher at Hitachi Central Research Laboratories, Tokyo, Japan. Since 1999, he is a Full Professor and Chairman of the Signal Processing Systems Group at TU/e. He is the author or coauthor of several research papers published in refereed journals and is the author of *Digital Baseband Transmisison and Recording* (Kluwer Academic, 1996, 652 pp.). He is the holder of around 30 U.S. patents.

Andre H. J. Immink received the degree of Electrotechnisch Ingenieur (*cum laude*) from the University of Twente, Enschede, The Netherlands, in 1995, and the Ph.D. degree from Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands, in 2005.

From 1995 to 2006, he was with Philips Research Laboratories, Eindhoven, where he was engaged in research on signal processing techniques and mixed-signal integrated circuit (IC) design for optical storage and later on biosensors using giant magnetoresistive sensors. Since January 2007, he is a Principal System Architect in a Philips Healthcare Incubator, NY working on biosensor technology based on magnetic nanoparticle labels.