Extraction of Timing Error Parameters From Readback Waveforms

Wei Zeng[1], Aleksandar Kavcic[1], and Ravi Motwani[2]

[1]DEAS, Harvard University, Cambridge, MA, 02138 USA
[2]Data Storage Institute, Singapore 117608

In this paper, we consider the problem of modeling the timing error process in magnetic recording systems. We propose a discrete-valued Markov model for the timing error process, and design two methods (data-aided and nondata-aided), based on the Baum–Welch algorithm, to extract the model parameters from the readback waveforms. The channel model we consider is an intersymbol interference (ISI) channel with additive Gaussian noise. The continuous-time readback signal at the output of the channel is sampled at baud-rate. Simulation results show that the estimated parameters are close to the actual values and the convergence is attained in a few iterations of the Baum–Welch algorithm. We also demonstrate the usefulness of the accurate model extraction by comparing a fine-tuned Markov timing recovery loop to the standard Mueller and Muller detector with a tuned second-order loop filter.

***Index Terms*—Baum–Welch algorithm, cycle-slip, intersymbol interference, synchronization, timing recovery.**

I. Introduction

MAGNETIC recording (and communications) channels suffer from misalignment between the write (transmitter) clock and the read (receiver) clock. The timing recovery method at the receiver, not being ideal, produces synchronization errors.

It is widely accepted that in magnetic recording applications, the detectors/decoders need to be fine-tuned to the signal and noise characteristics in order to achieve maximal gains. However, so far this line of thinking has not penetrated into the design of timing recovery systems in magnetic recording. Typically, derivations of the synchronizers assume that the timing error is a *fractional* offset [1] of the sampling instant (which is an assumption that breaks in the cycle-slip region). This assumption leads to the design of standard first and/or second order timing recovery loops with only 2–3 tunable parameters. Albeit simple, such schemes suffer from frequent cycle-slips, especially at low signal-to-noise ratios (SNRs) or when the timing error is large. For example, when the timing error is more than a bit interval, the timing recovery loop will tend to lock at a new stable point, which is a cycle (or several cycles) away from the ideal sampling instant. Since the timing error model does not consider such "large" timing errors (cycle-slips), if the timing recovery loop experiences a cycle-slip, it does not have a mechanism to resynchronize. Clearly, if we could construct an accurate statistical model for the timing error process, we would have the opportunity to fine-tune the timing recovery loop to match the model, or even construct more accurate timing recovery loops.

In [2], [3], a discrete Markov model for the timing errors has been proposed, which led to a soft-output algorithm to jointly detect data symbols and timing errors. However, the soft-output algorithm makes the assumption that the parameters of the Markov timing error model are known, which in practice need to be estimated. In this paper, we consider the problem of estimating the parameters of the timing error model from baud-rate samples of readback waveforms.

Digital Object Identifier 10.1109/TMAG.2005.861756

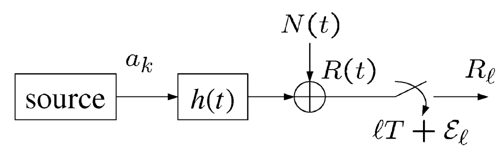

Fig. 1. Simple block diagram of the channel and signal model.

The organization of the paper is as follows. In Section II, a Markov model for the timing error process, as well as its equivalent finite-state trellis model are proposed. In Sections III and IV, we propose the data-aided and nondata-aided estimation methods, respectively. We present the simulation results in Section V, and conclude the paper in Section VI.

II. Source/Channel Model and Problem Formulation

We consider a simple system, as shown in Fig. 1. We denote the binary antipodal written symbol (channel input symbol) at time $k \in \mathbb{Z}$ by $a_k \in \{-1, 1\}$.

The channel response function $h(t)$ is modulated by the input sequence $\{a_k\}$. The readback waveform $R(t)$ has the form

$$R(t) = \sum_k a_k h(t - kT) + N(t) \tag{1}$$

where T is the symbol interval and $N(t)$ is additive Gaussian noise.

An ideal receiver would sample the readback waveform $R(t)$ at time instants $T, 2T, 3T, \ldots$. However, because of the timing errors, the receiver samples the readback waveform at the following sampling instants $T + \mathcal{E}_1, 2T + \mathcal{E}_2, \ldots, \ell T + \mathcal{E}_\ell$, and so on. That is, the ℓth sample is

$$R_\ell = R(\ell T + \mathcal{E}_\ell) + N_\ell \tag{2}$$

where N_ℓ are independent and identically distributed Gaussian random variables with mean 0 and variance σ^2, shortly denoted by $N_\ell \sim \mathcal{N}(0, \sigma^2)$. For simplicity, we assume that N_ℓ are independent of the input sequence $\{a_k\}$. Generalization to pattern-dependent noise [4] is straightforward.

We assume that $h(t)$ is a *finite-support function* that satisfies

$$h(t) = 0, \quad \text{for } |t| \geq vT. \tag{3}$$

We denote the support interval of $h(t)$ by $(-vT, vT)$, where $v \in \mathbb{Z}^+$. By using (3), we can now rewrite (2) as

$$R_\ell = \sum_{k=\ell-v+\lceil \mathcal{E}_\ell/T \rceil}^{\ell+v+\lfloor \mathcal{E}_\ell/T \rfloor} a_k h(\ell T + \mathcal{E}_\ell - kT) + N_\ell. \tag{4}$$

Equation (4) shows that at most

$$D = 2v \tag{5}$$

data symbols influence the value of each readback sample R_ℓ. For example, if we assume that the readback sample R_ℓ falls inside the kth symbol interval $((k-1)T, kT]$, the $2v$ data symbols that influence R_ℓ are $a_{k-v}, a_{k-v+1}, \ldots, a_{k+v-1}$.

Notational Conventions: Throughout the paper, we use the following notational conventions. The probability of an event A is denoted by $P(A)$. The probability of an event A, conditioned on an event B, is denoted by $P(A \mid B)$. A sequence of variables $b_i, b_{i+1}, \ldots, b_j$ is shortly denoted by b_i^j. A readback sample R_ℓ in (4) is a random variable, where its realization is denoted by r_ℓ. The positive integer K denotes the block length of written symbols (bits) a_1^K. The index k is exclusively used to count the written symbols (bits) a_k, where $1 \leq k \leq K$. For a given block of input bits a_1^K, we will take L readback samples R_1^L, whose realizations are r_1^L. Throughout the paper, the index ℓ (confined to $1 \leq \ell \leq L$) is exclusively used to count the readback samples R_ℓ (or r_ℓ). We require that $L = K - W$, where W is a known positive integer. The discrepancy between L and K is required because of the timing errors. The integer W is a safety window length whose value is chosen to guarantee that the readback samples R_1^L fall within the K symbol intervals that correspond to the K written bits a_1^K.

A. A Practical Physical Model for the Timing Error

The timing error $\mathcal{E}_\ell$ is the difference between the sampling instant of the ℓth sample of the readback waveform and the expected sampling instant ℓT. Since the writing process and the reading process are never perfectly synchronized, the timing error $\mathcal{E}_\ell$ is a random process. Often this random process is modeled to be a constant for an extended number of bit intervals [1], or is modeled as Gaussian random walk [3]. Though such models are simple, they are often not accurate enough and do not provide enough parameters to tune the synchronizer.

Here we adopt a timing error process model with several tunable parameters. We uniformly quantize the symbol interval into Q levels, and allow $\mathcal{E}_\ell$ to take values jT/Q, where j is an integer. Obviously, the quantization is an approximation, which, if chosen to be fine enough, introduces only a marginally small quantization error. We further assume that $\mathcal{E}_\ell$ can be represented by a Markov process with the probability of transition from state iT/Q to state jT/Q denoted by $q_{i,j}$. Fig. 2 illustrates an example, where nonzero transition probabilities occur only between neighboring states. Obviously, more accurate Markov models with transitions between nonneighboring states are possible (but not considered here for the simplicity of the presentation). Note that due to the cyclostationarity of the timing error process, $q_{i,j}$ must equal $q_{i+Q,j+Q}$. That is

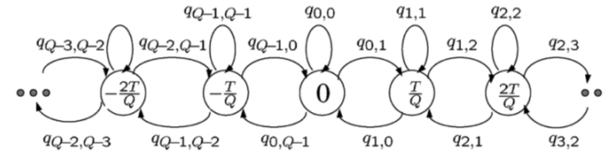

Fig. 2. A simple state-transition diagram of the timing error $\mathcal{E}_\ell$.

$$P\left(\mathcal{E}_{\ell+1} = \varepsilon_{\ell+1} \,\middle|\, \mathcal{E}_\ell = \varepsilon_\ell = \frac{i+nQ}{Q}T\right) = \begin{cases} q_{i,i-1}, & \text{if } \varepsilon_{\ell+1} = \varepsilon_\ell - \frac{T}{Q} \\ q_{i,i+1}, & \text{if } \varepsilon_{\ell+1} = \varepsilon_\ell + \frac{T}{Q} \\ 1 - q_{i,i+1} - q_{i,i-1}, & \text{if } \varepsilon_{\ell+1} = \varepsilon_\ell \end{cases} \tag{6}$$

where $i \in \{0, 1, \ldots, Q-1\}$ and n is any integer $n \in \mathbb{Z}$.

One advantage of this discrete Markov model for the timing error process is that it does *not* assume the timing error to be only a fraction of the bit interval. Rather, it allows the scenario where the timing error is greater than the bit interval, i.e., the model allows cycle-slips to occur. Thus, the model permits the design of advanced timing recovery loops and channel detectors that are resilient to cycle-slips [5], [3], [6].

It is not hard to verify that under this model, the number of possible values of the timing error $\mathcal{E}_\ell$ grows linearly with time ℓ. Obviously this creates problems, because the number of timing error states is unbounded (see Fig. 2). We circumvent this problem by proposing an equivalent *finite-state* trellis model for the timing error process.

B. Equivalent Finite-State Trellis Model

A state s_k at time k is an element of the finite-size set $\mathcal{T}$ of all possible states

$$s_k \in \mathcal{T} \tag{7}$$

where

$$\mathcal{T} = \{0, 1, 2, \ldots, Q-1, Q, Q+1\}. \tag{8}$$

We next assign a meaning to the elements of the set $\mathcal{T}$. A state s_k at time k is associated with the kth symbol interval

$$((k-1)T, kT].$$

The model in (6) implies that one of the following three events must occur, depending on the realization of the timing error process.

1) No sampling instant falls in the kth symbol interval. This event is denoted by a state

$$s_k = 0.$$

2) Exactly one sampling instant falls in the kth symbol interval. The value of this single sampling instant is confined to one of the Q possible quantized realizations $(k-1)T + iT/Q$, where $i \in \{1, 2, \ldots, Q\}$. Each of the quantized realizations is denoted by a state

$$s_k = i.$$

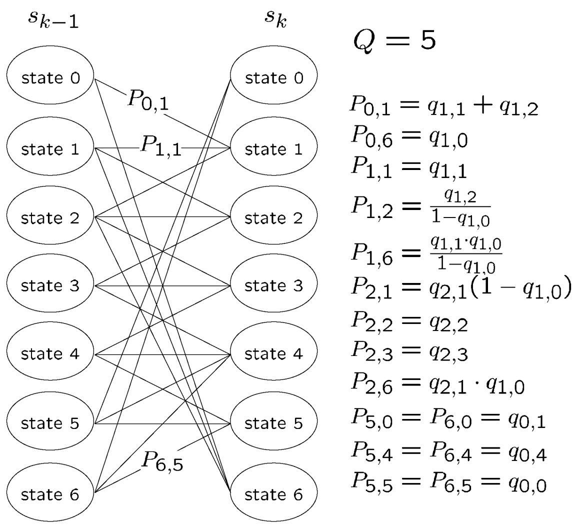

Fig. 3. Section of the equivalent timing error trellis for $Q = 5$.

3) Exactly two sampling instants fall in the kth symbol interval. These two sampling instants are

$$(k-1)T + \frac{T}{Q} \quad \text{and} \quad kT.$$

This is denoted by a state

$$s_k = Q + 1.$$

The state sequence $s_k \in \mathcal{T}$ is a homogeneous Markov process with

$$P\left(s_k \,\middle|\, s_0^{k-1}\right) = P(s_k \,|\, s_{k-1}).$$

Denote by $P_{i,j}$ the probability

$$P_{i,j} = P(s_k = j \,|\, s_{k-1} = i).$$

The values $P_{i,j}$ can be derived from the probabilities $q_{i,j}$ introduced in Section II-A. For $3 \leq i \leq Q-1$, the probabilities $P_{i,j}$ are

$$P_{i,j} = q_{i,j}, \quad \text{if } 3 \leq i \leq Q-1. \tag{9}$$

However, there are special cases that do not obey rule (9) and must be handled with care. For example, Fig. 3 shows a trellis section for $Q = 5$ and lists the values $P_{i,j}$ for all i that do not conform to (9).

The timing error trellis is thus described by the set of states $\mathcal{T}$ and by the state transition probabilities $P_{i,j}$. The probabilities $P_{i,j}$ are the tunable parameters that describe the timing error process $\mathcal{E}_\ell$. For notational purposes, we group these parameters $P_{i,j}$ into the parameter set $\mathcal{P}$

$$\mathcal{P} = \{P_{i,j} : i \in \mathcal{T} \text{ and } j \in \mathcal{T}\}. \tag{10}$$

The advantage of the trellis model for the timing error process is that the state sequence is essentially a homogeneous *finite-state* Markov chain. There are various estimation techniques that we can apply to solve many problems associated with this finite-state Markov model. For example, we can do maximum *a posteriori* probability symbol detection under timing uncertainty [5], or we can do iterative timing recovery [3] by calculating soft-outputs from this finite-state trellis model. In this paper, we study the problem of determining the transition probabilities $P_{i,j}$ from readback waveform samples (channel outputs) that are corrupted by timing errors.

C. Parameter Estimation Problem

Although the model in Fig. 2 is physically intuitive (and therefore appealing), direct estimation of $q_{i,j}$ in Fig. 2 is not easy. On the other hand, estimating the parameters $P_{i,j}$ of the *equivalent* finite-state Markov model is a more attractive approach for two reasons:

1) the model is a finite-state model, and hence only a finite number of parameters $P_{i,j}$ needs to be estimated;
2) the model can be directly used to design optimal detectors for channels with synchronization errors [5], [3].

Our modeling task is to observe (capture) L readback samples r_1^L (sampled with timing errors), and estimate the trellis model parameters $P_{i,j}$ from the captured samples. We assume that the channel impulse (or transition) response $h(t)$ is *a priori* known; if it is not known, it can be easily measured. To perform the modeling task, we will consider two scenarios:

- the *data-aided* scenario, where the written bits are known (for example, blocks of training symbols are written);
- the *nondata-aided* scenario, where the written bits are not known and are generated randomly with a known probability distribution prior to the writing.

The basic tool that we use for the parameter estimation task is the Baum–Welch algorithm [7]. In its classical form, the algorithm performs the estimation of transition probabilities on a trellis representation of a Markov-memory process. Here, however, the classical Baum–Welch algorithm must be modified. To justify this, consider the example in Fig. 3. The state $s_k = 0$ denotes that the kth symbol interval is not sampled at all. Hence, no sample r_ℓ can be associated with the state $s_k = 0$. This illustrates that there may not be a one-to-one correspondence between the states and readback waveform samples (channel output samples) as in the ordinary trellises used for running BCJR (forward-backward) algorithms [8]. Hence, we need to modify the forward-backward component of the Baum–Welch algorithm [7].

III. Data-Aided Iterative Estimation

The Baum–Welch algorithm is an iterative algorithm, where each iteration consists of two steps [7]:

1) the forward-backward (sum-product) algorithm;
2) the re-estimation algorithm.

To describe both components of the Baum–Welch procedure for our specific trellis, we introduce the following notation.

- Let Z_k denote the vector of readback samples that correspond to the k-th symbol interval. Obviously, the length of Z_k is not fixed, but rather it depends on the realization of the state s_k.

—If $s_k = 0$, the kth symbol interval is not sampled at all, and hence Z_k is an empty vector, $Z_k = \emptyset$.

—If $s_k \in \{1, 2, \ldots, Q\}$, the kth symbol interval is sampled once, and hence Z_k is a vector of length 1.

—If $s_k = Q+1$, the kth symbol interval is sampled twice, and hence Z_k is a vector of length 2.

A. Forward-Backward Algorithm

Given the set of state transition probabilities $\mathcal{P}$, and the realizations of readback samples r_1^L, we wish to calculate the following two probabilities:

$$\zeta_k(s_k, s_{k+1}, \mathcal{P}) = P(s_k, s_{k+1}, r_1^L | \mathcal{P}) \tag{11}$$

$$\mu_k(s_k, \mathcal{P}) = P(s_k, r_1^L | \mathcal{P}) \tag{12}$$

where $s_k \in \mathcal{T}$ and $s_{k+1} \in \mathcal{T}$.

We define the following functions (for $0 \le \ell \le L$):

$$\alpha_k(m, \ell) = P(s_k = m, Z_1^k = r_1^\ell) \tag{13}$$

$$\beta_k(m, \ell) = P(Z_{k+1}^K = r_{\ell+1}^L | s_k = m) \tag{14}$$

where $m \in \mathcal{T}$ is any timing error state. In words, $\alpha_k(m, \ell)$ is the joint probability that the timing error state at time k is m, and that there are ℓ samples, r_1^ℓ, that fall inside the first k symbol intervals $(0, kT]$. We also define the following branch metric function, for $m \in \mathcal{T}$ and $m' \in \mathcal{T}$ and $0 \le \ell \le L$:

$$\begin{aligned} \gamma_k(m, m', \ell) &= P(s_k = m' | s_{k-1} = m) \cdot P(Z_k | s_{k-1} = m, s_k = m') \\ &= \begin{cases} P(S_k = m', Z_k = r_\ell | S_{k-1} = m), & \text{if } 1 \le m' \le Q \\ P(S_k = m', Z_k = r_{\ell-1}^\ell | S_{k-1} = m), & \text{if } m' = Q+1 \\ P(S_k = m', Z_k = \emptyset | S_{k-1} = m), & \text{if } m' = 0. \end{cases} \end{aligned} \tag{15}$$

Obviously, if we know the state transition probabilities $\mathcal{P}$, as well as the statistics of the channel noise N_ℓ in (4), it is straightforward to calculate the above branch metrics $\gamma_k(m, m', \ell)$.

From the Markovian property of the trellis, we can compute $\alpha_k(m, \ell)$ recursively, for $m \in \mathcal{T}$ and $1 < k \le K$ and $k - W \le \ell \le k + W$

$$\begin{aligned} \alpha_k(m, \ell) &= \begin{cases} \sum_{m' \in \mathcal{T}} \alpha_{k-1}(m', \ell-1) \gamma_k(m', m, \ell), & \text{if } 1 \le m \le Q \\ \sum_{m' \in \mathcal{T}} \alpha_{k-1}(m', \ell) \gamma_k(m', m, \ell), & \text{if } m = 0 \\ \sum_{m' \in \mathcal{T}} \alpha_{k-1}(m', \ell-2) \gamma_k(m', m, \ell), & \text{if } m = Q+1. \end{cases} \end{aligned} \tag{16}$$

Notice that here we adopt a fixed integer $W \in \mathbb{Z}^+$ as a safety window length that specifies the maximum number of cycle-slips considered by the algorithm. In practice, this number can be adjusted by the user.

We assume that the sampling clock starts with zero phase-offset (perfect clock acquisition), and set the initial conditions for $\alpha_k(\cdot, \cdot)$ as

$$\alpha_1(m, \ell) = \begin{cases} \gamma_1(Q, m, 1), & \text{if } m \in \{Q-1, Q\} \text{ and } \ell = 1 \\ \gamma_1(Q, m, 0), & \text{if } m = 0 \text{ and } \ell = 0 \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

Similarly, using the Markovian property of the trellis, we have a recursion for $\beta_k(\cdot, \cdot)$ for $1 \le k < L - W$ and $k - W \le \ell \le k + W$

$$\begin{aligned} \beta_k(m, \ell) &= \sum_{1 \le m' \le Q} \gamma_{k+1}(m, m', \ell+1) \beta_{k+1}(m', \ell+1) \\ &\quad + \gamma_{k+1}(m, 0, \ell) \beta_{k+1}(0, \ell) \\ &\quad + \gamma_{k+1}(m, Q+1, \ell+2) \beta_{k+1}(Q+1, \ell+2). \end{aligned} \tag{18}$$

We set a simple (approximate) initial condition for $\beta_k(\cdot, \cdot)$ at $k = L - W = K - 2W$ to be uniform for all $m \in \mathcal{T}$

$$\beta_{K-2W}(m, \ell) = \begin{cases} 1, & L - 2W \le \ell \le L; \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

The two joint probabilities in (11) and (12) are computed as follows: for $m \in \mathcal{T}$, and $m' \in \mathcal{T}$ and $1 \le k < K - 2W$

$$\begin{aligned} \zeta_k(m, m', \mathcal{P}) &= \begin{cases} \sum_{\ell=k-W}^{k+W} [\alpha_k(m, \ell) \gamma_{k+1}(m, m', \ell+1) \beta_{k+1}(m', \ell+1)] \\ \quad \text{if } 1 \le m' \le Q \\ \sum_{\ell=k-W}^{k+W} [\alpha_k(m, \ell) \gamma_{k+1}(m, m', \ell+2) \beta_{k+1}(m', \ell+2)] \\ \quad \text{if } m' = Q+1 \\ \sum_{\ell=k-W}^{k+W} [\alpha_k(m, \ell) \gamma_{k+1}(m, m', \ell) \beta_k(m', \ell)] \\ \quad \text{if } m' = 0, \end{cases} \end{aligned} \tag{20}$$

$$\mu_k(m, \mathcal{P}) = \sum_{\ell=k-W}^{k+W} \alpha_k(m, \ell) \beta_k(m, \ell). \tag{21}$$

Similar to the BCJR algorithm [8], this algorithm has forward and backward recursions, given by (16) and (18), respectively. Just like in the BCJR algorithm [8], normalizations of the coefficients α_k and β_k are required here as well.

B. Re-Estimation Using the Baum–Welch Algorithm

The forward-backward algorithm given above computes the probabilities ζ_k and μ_k assuming that the set of state transition probabilities $\mathcal{P}$ is known. In practice, however, the set $\mathcal{P}$ is generally not known and thus needs to be estimated from the readback samples. We now introduce the iterative Baum–Welch method for re-estimating these transition probabilities $\mathcal{P}$.

Denote by $\hat{\mathcal{P}}^{(n)}$ the estimate of the set of transition probabilities $\mathcal{P}$ in the n-th iteration of the algorithm. Our task is to estimate a new set $\hat{\mathcal{P}}^{(n+1)}$ of transition probabilities using the computed probabilities $\mu_k(i, \hat{\mathcal{P}}^{(n)})$ and $\zeta_k(i, j, \hat{\mathcal{P}}^{(n)})$. Notice that the expected number of visits to state i is given by $\sum_{k=1}^{K-2W} \mu_k(i, \hat{\mathcal{P}}^{(n)})$ and the expected number of transitions from state i to state j is given by $\sum_{k=1}^{K-2W} \zeta_k(i, j, \hat{\mathcal{P}}^{(n)})$. Then the state transition probabilities can be estimated as the ratio of the expected counts

$$\hat{P}_{i,j}^{(n+1)} = \frac{\sum_{k=1}^{K-2W-1} \zeta_k \left(i, j, \hat{\mathcal{P}}^{(n)}\right)}{\sum_{k=1}^{K-2W-1} \mu_k \left(i, \hat{\mathcal{P}}^{(n)}\right)} \tag{22}$$

and the new set of estimated transition probabilities is

$$\hat{\mathcal{P}}^{(n+1)} = \left\{ \hat{P}_{i,j}^{(n+1)} : i \in \mathcal{T} \text{ and } j \in \mathcal{T} \right\}.$$

We now formulate the data-aided Baum–Welch algorithm to estimate the state transition probabilities:

1) Write a known sequence of data bits a_1^K.
2) Sample a long readback sequence r_1^L, where $L = K - W$.
3) Set an initial estimate of the transition probabilities $\hat{\mathcal{P}}^{(0)}$.
4) In the n-th iteration of the Baum-Welch algorithm, use the estimate $\hat{\mathcal{P}}^{(n)}$ as the state transition probabilities of the timing error trellis. Calculate the α and β coefficients according to Eqns (16), (17), (18) and (19).
5) Calculate $\zeta_k(m, m', \hat{\mathcal{P}}^{(n)})$ and $\mu_k(m, \hat{\mathcal{P}}^{(n)})$ according to Eqns. (20) and (21) for $1 \le k < K - 2W$ and $m \in \mathcal{T}$ and $m' \in \mathcal{T}$.
6) The $(n+1)$-th estimate of the transition probabilities $\hat{\mathcal{P}}^{(n+1)}$ is calculated by (22).
7) If no convergence occurs, increase the iteration count n by 1 and go to step 4.

IV. Non-Data-Aided Iterative Estimation

In this case, the channel input symbols are not known, so there are two unknown processes with memory that need to be modeled: 1) the timing error process $\mathcal{E}_\ell$ and 2) the channel input process a_k observed through an intersymbol interference (ISI) channel. Hence, we need to build a trellis representation of the joint timing-error/ISI process. The structure of such a joint trellis, as well as the forward-backward algorithm for this joint trellis have been described in [5], [3].

The Baum–Welch algorithm for the nondata-aided scenario is thus similar to the data-aided scenario. Using a forward-backward algorithm, properly applied to the joint timing-error/ISI trellis [5], [3], we can compute the probabilities ζ_k and μ_k that are needed in the first half of each Baum–Welch iteration. The second half of each Baum–Welch iteration (the re-estimation step) is identical to (22). Thus, the difference between the data-aided and the nondata-aided Baum–Welch algorithm is only in the implementation of the forward-backward algorithm.

V. Simulation Results

In order to evaluate the performance of the Baum–Welch estimation algorithm, we compare the estimated state transition probabilities $\hat{\mathcal{P}}^{(n)}$ after the nth iteration of the Baum–Welch algorithm to the true values $\mathcal{P}$. We consider the simple case where we have $Q = 5$ quantization levels per symbol interval, and set the parameters

$$q_{i,j} = \begin{cases} 0.98, & i = j \\ 0.01, & i = j \pm 1 \end{cases}. \tag{23}$$

The channel is a simple PR4 channel [9]. The length of the input sequence was selected to be $K = 3000$ symbols and SNR is 4 dB. The number of readback samples is $L = 2996$ (i.e., $W = 4$). Table I shows the corresponding true state transition probabilities $\mathcal{P}$, as well as the estimated transition probabilities $\hat{\mathcal{P}}^{(n)}$ after the nth Baum–Welch iteration using the *data-aided* method.

To illustrate how fast the Baum–Welch algorithm converges to the true state transition probabilities $\mathcal{P}$, we define

$$E^{(n)} = \sum_{0 \le i,j \le Q+1} \left[\hat{P}_{i,j}^{(n)} - P_{i,j} \right]^2 \tag{24}$$

TABLE I
Iterative Estimation of Timing State Transition Probabilities $\mathcal{P}$

True transition probabilities $\mathcal{P}$							
	j=0	j=1	j=2	j=3	j=4	j=5	j=6
i=0		0.99					0.01
i=1		0.98	0.01				0.01
i=2		0.01	0.98	0.01			0.00
i=3			0.01	0.98	0.01		
i=4				0.01	0.98	0.01	
i=5	0.01				0.01	0.98	
i=6	0.01				0.01	0.98	
Initial estimate $\hat{\mathcal{P}}^{(0)}$							
	j=0	j=1	j=2	j=3	j=4	j=5	j=6
i=0		0.9					0.1
i=1		0.8	0.11				0.09
i=2		0.09	0.8	0.1			0.01
i=3			0.1	0.8	0.1		
i=4				0.1	0.8	0.1	
i=5	0.1				0.1	0.8	
i=6	0.1				0.1	0.8	
Estimated probabilities $\hat{\mathcal{P}}^{(2)}$ after the 2-nd iteration							
	j=0	j=1	j=2	j=3	j=4	j=5	j=6
i=0		0.95					0.05
i=1		0.91	0.04				0.05
i=2		0.02	0.95	0.03			0.00
i=3			0.02	0.95	0.03		
i=4				0.05	0.92	0.01	
i=5	0.03				0.02	0.95	
i=6	0.03				0.02	0.95	
Estimated probabilities $\hat{\mathcal{P}}^{(8)}$ after the 8-th iteration							
	j=0	j=1	j=2	j=3	j=4	j=5	j=6
i=0		0.99					0.01
i=1		0.97	0.02				0.01
i=2		0.01	0.97	0.02			0.00
i=3			0.01	0.98	0.01		
i=4				0.02	0.96	0.01	
i=5	0.01				0.01	0.98	
i=6	0.01				0.01	0.98	

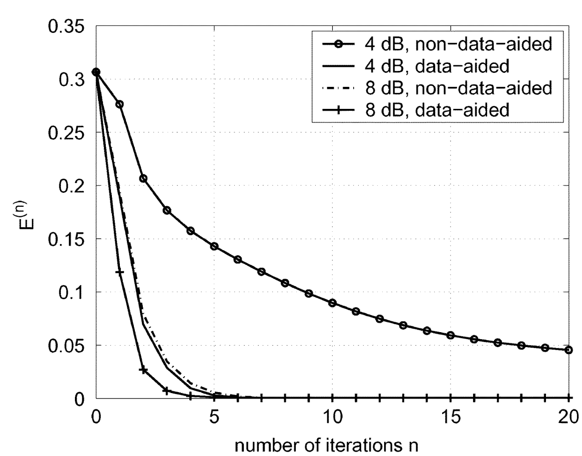

Fig. 4. Squared estimation error $E^{(n)}$ between the state transition probabilities $\hat{\mathcal{P}}^{(n)}$ and the true values $\mathcal{P}$.

as the total squared parameter estimation error in the nth iteration of the Baum–Welch algorithm. Fig. 4 plots $E^{(n)}$ versus the iteration number n for the case where the true parameters $\mathcal{P}$ and the initial estimate $\hat{\mathcal{P}}^{(0)}$ are specified in Table I. Both data-aided and nondata-aided estimation are evaluated. The block length of the input sequence was selected to be $K = 3000$, the number of readback samples is $L = 2996$, and the number of quantization levels was chosen to be $Q = 5$. We notice that the data-aided algorithm converges in less than six iterations for both 4 and

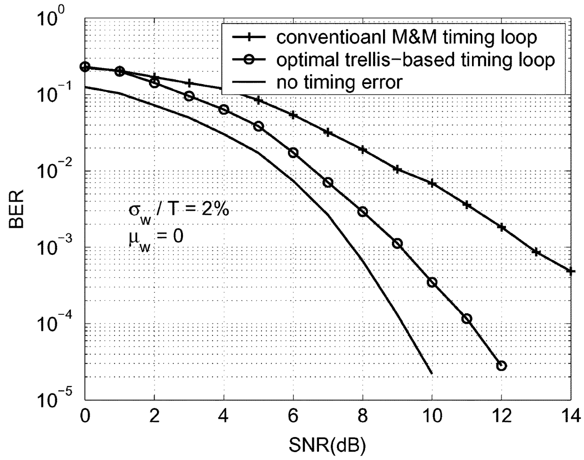

Fig. 5. Bit error rates when using the conventional M&M timing recovery detector with an optimized second-order loop filter [10] and the optimal trellis-based timing recovery loop [6] fine tuned to the Markov timing error model.

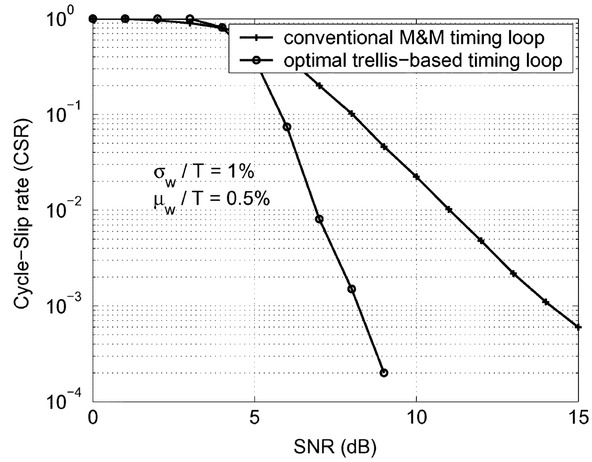

Fig. 6. Cycle-slip rates when using the conventional M&M timing recovery detector with optimized second-order loop filter [10] and the optimal trellis-based timing recovery loop [6] fine tuned to the Markov timing error model.

8 dB, while the nondata-aided algorithm converges in less than six iterations only at 8 dB.

To underline the advantage of properly modeling the timing error process, we illustrate the performances of two timing recovery loops: one that does not take the model into account, and the other that does. The real timing error in the system is a Gaussian random walk process [3] whose Gaussian increment has mean μ_w and variance σ_w^2. The two compared timing recovery loops are:

1) the standard M&M timing error detector [10] with a second-order loop filter (whose filter coefficients are optimized for each tested SNR);
2) the optimal baud-rate timing recovery loop [6], fine tuned to the Markov parameters of the timing error process, where the Markov timing error parameters are extracted from the readback waveforms using the Baum–Welch algorithm.

In Fig. 5, we compare the bit error rate (BER) performance of the Viterbi detectors when the two timing recovery loops are used, and the timing error parameters are $\mu_w = 0$ and $\sigma_w/T = 2\%$. Clearly, the timing loop that utilizes the estimated Markov timing error process parameters performs visibly better.

In Fig. 6, we compare the cycle-slip rate of the two timing loops under the scenario $\mu_w/T = 0.5\%$ and $\sigma_w/T = 1\%$. Clearly, the loop that utilizes the estimates of the Markov timing error parameters has a much lower cycle-slip rate. This is because the Markov model for the timing error allows for cycle-slips. Consequently, if a timing loop is fine-tuned to such a Markov model, it has the ability to recover from occasional cycle-slips, which is well illustrated in Fig. 6.

VI. Conclusion

In this paper, we presented a Baum–Welch method for estimating the parameters of a finite-state Markov timing error process. Both data-aided and nondata-aided methods have been proposed. The proper extraction of the timing error parameters is required for fine-tuning the timing recovery and data detection algorithms to the characteristics of a head-disk interface in magnetic recording systems. Simulation results were presented that show: 1) the ability of the Baum–Welch algorithm to correctly estimate the timing error parameters and 2) the superior performance of timing recovery loops that are tuned to the estimated timing error parameters.

Acknowledgment

This work was supported by a grant from the Information Storage Industry Consortium (INSIC).

References

[1] H. Meyr, M. Moeneclaey, and S. A. Fechtel, *Digital Communication Receivers*. New York: Wiley, 1998.
[2] W. Zeng and A. Kavčič, "MAP detection in noisy channels with synchronization errors (including the insertion/deletion channel)," in *Proc. IEEE Int. Symp. Information Theory*, Yokohama, Japan, Jun. 2003, p. 232.
[3] J. R. Barry, A. Kavčič, S. W. McLaughlin, A. Nayak, and W. Zeng, "Iterative timing recovery," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 89–102, Jan. 2004.
[4] A. Kavčič and J. J. M. Moura, "The Viterbi algorithm and Markov noise memory," *IEEE Trans. Inf. Theory*, vol. 46, no. 1, pp. 291–301, Jan. 2000.
[5] W. Zeng and A. Kavčič, "Optimal (MAP) soft-output detector for channels with intersymbol interference and timing recovery errors," *IEEE Trans. Magn.*, vol. 39, no. 5, pp. 2555–2557, Sep. 2003.
[6] W. Zeng, M. F. Erden, A. Kavčič, E. M. Kurtas, and R. C. Venkataramani, "Optimal baud-rate timing recovery loops using trellises," *IEEE Trans. Magn.*, submitted for publication.
[7] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
[8] 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. 2, pp. 284–287, Mar. 1974.
[9] P. Kabal and S. Pasupathy, "Partial-response signaling," *IEEE Trans. Commun.*, vol. COM-23, no. 9, pp. 921–934, Sep. 1975.
[10] K. H. Mueller and M. Müller, "Timing recovery in digital synchronous data receivers," *IEEE Trans. Commun.*, vol. COM-24, no. 5, pp. 516–531, May 1976.

Manuscript received July 1, 2005 (e-mail: wzeng@deas.harvard.edu).