Trellis-Based Optimal Baud-Rate Timing Recovery Loops for Magnetic Recording Systems

Wei Zeng[1], M. Fatih Erden[2], Aleksandar Kavčić[3], Erozan M. Kurtas[2], and Raman C. Venkataramani[3]

[1]Division of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138 USA
[2]Seagate Technology, Pittsburgh, PA 15222 USA
[3]Department of Electrical Engineering, University of Hawaii at Manoa, Honolulu, HI 96822 USA

The advent of iteratively decodable codes has allowed a decrease in tolerable signal-to-noise ratios (SNRs) in magnetic recording systems, which typically translates into an increase in the recording densities. However, at such low SNRs, conventional timing recovery loops suffer from frequent cycle slips. Typical timing recovery loops in magnetic recording applications perform data detection, timing error detection, and loop filtering in a sequential manner. This sequence of operations in the timing recovery loop performs well if the timing error is a small fraction of the bit interval. However, in the cycle-slip regions, the timing error is comparable to the bit interval, and the loop fails. In this paper, we represent the timing error in magnetic recording systems by using a Markov model that does not confine the timing error to only small fractions of the bit interval. By utilizing such a model, we give a trellis representation of the timing error process. The trellis representation permits the formulations of two optimal baud-rate timing recovery loops, according to two optimality criteria. We prove that both optimality criteria lead to solutions similar to the classical first-order phase-locked loop. The new loops do not perform data detection, timing error detection, and loop filtering in a sequential manner. Instead, the loops perform data detection and timing error detection jointly on a trellis, without the need for a loop filter. Simulation results show that the new timing recovery loops outperform the standard second-order baud-rate Mueller and Müller phase-locked loop with fine-tuned loop parameters. This performance gain is substantial if the timing error process is extremely noisy or if there is residual frequency offset resulting from inaccurate acquisition from the sector preamble on a disk drive.

Index Terms—**Intersymbol interference, phase-locked loop, timing recovery, trellis.**

I. Introduction

TIMING recovery is crucial to every communication and magnetic recording system, where the receiver needs to figure out the best instants at which to sample the received (or read-back) signal. In communication systems, the timing uncertainty may come from the random delay of the received signal, or from the slow drift of the receiver clock with respect to the transmitter clock. In magnetic recording systems, the mechanical motion fluctuation of the recording media during the writing and reading processes will lead to timing uncertainty. The purpose of the timing recovery unit (or synchronizer) is to estimate the random timing uncertainty in order to adjust the sampler. This problem has been well studied in the literature and engineering practice, and several timing recovery schemes have been proposed. A comprehensive exposition and classification of these schemes can be found in [1].

Most often, timing recovery loops are classified as *decision-directed* or *nondecision-directed* [1], depending on whether tentative decisions about the transmitted data symbols are needed for estimating the timing error. Kobayashi [2] derived a decision-directed receiver structure as an approximation of maximum-likelihood (ML) estimation [3]. Similar decision-directed detection algorithms based on ML estimation have also been studied by Qureshi and Newhall [4], and Ascheid and Meyr [5]. Non-decision-directed timing recovery strategies can also be derived using a similar ML criterion, where the log-likelihood function is averaged over random data symbols with different *a priori* distributions. Related work can be found in [6] and [7].

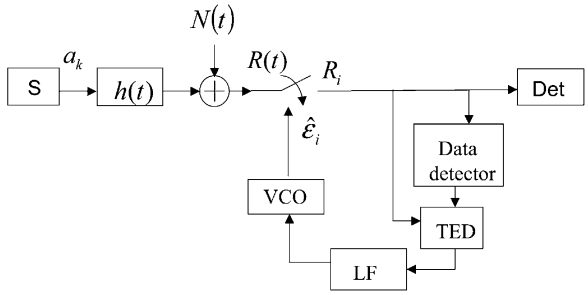

Fig. 1. A receiver with the conventional Mueller and Muller phase-locked loop. "Data detector" represents a short-delay Viterbi detector.

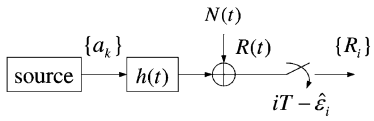

Fig. 2. Simple block diagram of the channel and signal model.

One of the most widely used *digital* decision-directed timing error detector (TED), which operates on the baud-rate samples of the baseband signal, was proposed by Mueller and Müller (M&M) [8]. The M&M TED uses a weighted sum of the samples, whose expected value defines a timing function. A typical timing recovery loop that contains such a TED consists of a data detector, a TED, a loop filter (LF), and a voltage controlled oscillator (VCO), as shown in Fig. 1.

Adaptive Kalman filtering theory has been applied by Driessen [9] and Patapoutian [10] to a specific linearized timing error model. It was shown that the Kalman filter has the same structure as the second order phase-locked loop (PLL) for such a model, with time-varying loop coefficients. Recursive and closed-form expressions of the Kalman gain are derived for

Digital Object Identifier 10.1109/TMAG.2006.881094

both acquisition and tracking modes of the PLL. A comprehensive analysis and comparison of several symbol-rate timing recovery schemes can be found in [11].

Another problem that is similar to the symbol timing recovery is the carrier phase recovery in communication systems that utilize carrier modulated signals. Many similar signal processing techniques can be applied here and have been thoroughly studied [12], [13]. Scharf *et al.* [14] derived a Viterbi algorithm to jointly estimate the data sequence and phase error sequence, where the phase error sequence was modeled as an independent increment random process with modulo 2π. In [15], Dauwels and Loeliger studied the message passing algorithms for joint decoding and phase estimation over factor graphs. A key difference between the phase error studied in [14], [15] and the timing error studied here is that phase error only rotates the signal in the complex plane. However, timing error may cause symbol deletion or insertion (cycle-slip), which is much harder to correct. For example, a phase error of one period (2π) is equivalent to no phase error, while a timing error of one period means a cycle-slip.

In recent years, timing recovery has been recognized as one of the key bottlenecks in designing efficient high-density recording systems. Accordingly, several new approaches to timing recovery (both in the acquisition mode and tracking mode) have been proposed for magnetic data storage. New *iterative timing recovery* schemes were proposed for symbol detection in the presence of timing errors and intersymbol interference [16]–[18]. Such schemes utilize the power of iteratively decodable codes to combat the residual timing error (especially cycle-slips). These iterative timing recovery schemes showed remarkable performance gains over the conventional detection schemes. Further, novel training symbol placement methods for frequency acquisition have been suggested in [19], and other timing recovery schemes [20], [21] that do not rely on the conventional PLL have also been proposed, especially for recording channels with very low SNRs.

In this paper, we study timing recovery in the classical (i.e., *noniterative*) setting, where the synchronization is performed in real-time by a *timing recovery loop*. Our goal is to design the optimal baud-rate timing recovery loop for baseband communication channels typical for magnetic recording systems. Our solutions are proposed for the timing error process that can be modeled to be a discrete-time discrete-valued Markov random process. Of course, this assumption is not valid in practice where the timing error is typically continuous-valued. However, if we finely quantize the continuous-valued timing error, we can closely approximate the continuous-valued random process and thus achieve an approximately optimal timing recovery loop.

The paper is organized as follows. The baseband signal and timing error model typical for magnetic recording systems are introduced in Section II. The problem is formulated into optimization problems in Section III using two different criteria. Sections IV and V provide solutions to the two formulations, respectively. Section VI shows the simulation results. The complexities of the proposed solutions are discussed in Section VII, and Section VIII concludes the paper.

Notation: Throughout the paper, uppercase letters denote random processes, while lowercase letters denote their realizations. The notation $P(\mathcal{E}_k = \varepsilon_k)$ denotes the probability of the event $\mathcal{E}_k = \varepsilon_k$. Similarly, $P(\mathcal{E}_k = \varepsilon_k \,|\, R_m = r_m)$ denotes the conditional probability. When no confusion can arise, we will use short notation $P(\varepsilon_k)$ and $P(\varepsilon_k \,|\, r_m)$ to denote $P(\mathcal{E}_k = \varepsilon_k)$ and $P(\mathcal{E}_k = \varepsilon_k \,|\, R_m = r_m)$, respectively. We will use ε_1^k to denote the vector $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_k)$. Notation $A \sim \mathcal{N}(\mu, \sigma^2)$ denotes that the random variable A is normal (Gaussian) with mean μ and variance σ^2.

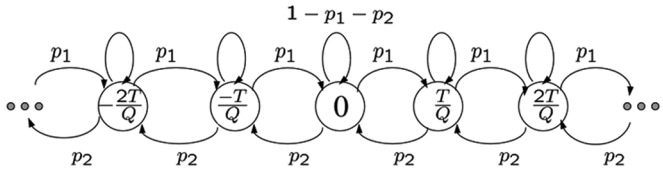

Fig. 3. State transition diagram of the Markov timing error $\mathcal{E}_i$.

II. Signal and Timing Error Models

Denote by a_k the binary antipodal channel input symbol $(a_k \in \{-1, 1\})$ at time $k \in \mathbb{Z}$. The channel response function $h(t)$ is modulated by the channel input sequence $\{a_k\}$. To make the analysis simple, we assume that $\{a_k\}$ is a sequence of independent and identically distributed (i.i.d.) equiprobable random variables.[1] The received waveform $R(t)$ is assumed to be a baseband signal typical for magnetic recording systems

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

where T is the symbol interval and $N(t)$ is additive noise. (Throughout the text, we use integer k as the index for input symbols, and integer i as the index for output samples.) If no timing error exists, the receiver will sample $R(t)$ at $t = iT$, for $i \in \mathbb{Z}$. However, if timing uncertainty exists, the actual sampling instant will be $t = iT + \mathcal{E}_i$. The random variable $\mathcal{E}_i$ represents the unknown timing uncertainty of the ith symbol in the sequence, and is independent of the channel input $\{a_k\}$.

We assume that the timing error $\mathcal{E}_i$ is a discrete-time, discrete-valued random process, that can take one of countably many values $(jT)/(Q)$, where j is an arbitrary integer and Q is a fixed positive integer. Clearly, Q is the number of quantization levels in each symbol interval. We further assume that the timing error $\mathcal{E}_i$ is *slowly* varying with time, and can be represented by the following independent increment random process:

$$\mathcal{E}_{i+1} = \mathcal{E}_i + \Delta_{i+1} \tag{2}$$

$$P(\Delta_i = \delta_i) = \begin{cases} p_1, & \text{if } \delta_i = \frac{T}{Q} \\ p_2 & \text{if } \delta_i = -\frac{T}{Q} \\ 1 - p_1 - p_2, & \text{if } \delta_i = 0 \end{cases}. \tag{3}$$

The initial value of this random process is $\mathcal{E}_0 = 0$. The increments of the timing error, Δ_i, are assumed to be i.i.d. and are independent of all previous samples R_j and previous timing errors $\mathcal{E}_j$, for $j < i$. Fig. 3 shows the state transition diagram of $\mathcal{E}_i$. We consider a slowly time-varying process $\mathcal{E}_i$, which permits the following assumption:

$$P(\Delta_i = 0) = 1 - p_1 - p_2 > \max(p_1, p_2). \tag{4}$$

[1]Extensions to Markov processes are straightforward.

When Δ_i has nonzero mean, i.e., $p_1 \neq p_2$, the (2) and (3) model a timing error process with frequency offset. If there is no frequency offset, we have $p_1 = p_2$.

We want to underline that the Markov model given by (2)–(3) and by Fig. 3 is only an *example* that we find very useful for illustrating the principle of trellis-based timing recovery. If needed, more accurate Markov models can easily be adopted without changing the nature of the problem. For example, without changing the structure of the solution that follows, we can allow transitions to nonadjacent states, say from state 0 to states $\pm 2Q/T$ or to states $\pm 3Q/T$, thus effectively allowing faster timing error processes to be tracked. Hence, the "bandwidth" of the modeled process is by no means limited to the choice of the quantization step T/Q.

Let us assume that the receiver has an estimate $\hat{\varepsilon}_i$ of $\mathcal{E}_i$. (Of course, our task in this paper will be to design the estimator that generates $\hat{\varepsilon}_i$, but for the time being, let us assume that the estimator $\hat{\mathcal{E}}_i$ is known.) In that case, the receiver samples the received signal at time instants $iT + \mathcal{E}_i - \hat{\varepsilon}_i$. Denote by R_i the ith sample at the receiver

$$\begin{aligned} R_i &= R(iT + \mathcal{E}_i - \hat{\varepsilon}_i) \\ &= \sum_k a_k h(iT - kT + \mathcal{E}_i - \hat{\varepsilon}_i) + N_i \end{aligned} \tag{5}$$

where N_i is the ith sample of the noise. For simplicity, we shall assume that $N_i \sim \mathcal{N}(0, \sigma_N^2)$ are i.i.d. Gaussian noise random variables. The model in (5) assumes that the timing error $\mathcal{E}_k$ is introduced by the receiver (i.e., by the reading process in storage channels). That is, we consider that the transmitted signal is a series of perfectly synchronized pulses, while the receiver samples the received signal with timing error.

III. Problem Formulation

A. Basic Problem Statement

The basic problem can be formulated as follows: upon receiving R_i, we need to make an estimate $\hat{\varepsilon}_{i+1}$ of the timing error $\mathcal{E}_{i+1}$. The value $\hat{\varepsilon}_{i+1}$ will be used to sample the received waveform at time instant $(i+1)T - \hat{\varepsilon}_{i+1}$.

Obviously, there are many choices of objective functions when designing the estimator for $\mathcal{E}_{i+1}$. In this section, we present two possible criteria. The first method is based on estimating the most likely sequence ε_1^{i+1}, and the second is based on estimating the most likely value ε_{i+1} without regarding the entire sequence. These two problem formulations are covered in detail in Sections III-C and III-D. However, before embarking on these detailed problem formulations, we introduce some more necessary notation.

B. More Notation

Definition 1: [Residual Timing Error]: The residual timing error $\mathcal{T}_i$ (with realization τ_i) is defined as

$$\mathcal{T}_i = \mathcal{E}_i - \hat{\varepsilon}_i. \tag{6}$$

■

Definition 2: [Finite Support Function $h(t)$]: We assume that the channel response function $h(t)$ satisfies

$$h(t) = 0, \quad \text{for } |t| \geq qT. \tag{7}$$

We denote the support interval of $h(t)$ by $(-qT, qT)$. ■

By using (6) and (7), We can now rewrite (5) as

$$R_i = \sum_{i-q+\lceil \mathcal{T}_i/T \rceil}^{i+q+\lfloor \mathcal{T}_i/T \rfloor} a_k h(iT - kT + \mathcal{T}_i) + N_i. \tag{8}$$

The upper limit of the summation in (8) will prove to be an important variable, so we denote it by a single letter L_i.

Definition 3 $[L_i]$: The random variable L_i, whose realization is ℓ_i, is defined as

$$\begin{aligned} L_i &= i + q + \left\lfloor \frac{\mathcal{T}_i}{T} \right\rfloor \\ &= i + q + \left\lfloor \frac{\mathcal{E}_i - \hat{\varepsilon}_i}{T} \right\rfloor. \end{aligned} \tag{9}$$

■

L_i is random because the residual timing error $\mathcal{T}_i = \mathcal{E}_i - \hat{\varepsilon}_i$ is random. However, $L_i = \ell_i$ is deterministic if ε_i (the realization of $\mathcal{E}_i$) and $\hat{\varepsilon}_i$ are known, as we can see from (9).

C. Problem Formulation Using an Optimal-Path Criterion

Denote by

$$P\left(\varepsilon_1^{i+1}, a_1^{\ell_i} \;\middle|\; r_1^i, \hat{\varepsilon}_1^i\right) = P\left(\varepsilon_1^{i+1}, a_1^{\ell_i} \;\middle|\; r_1^i\right) \tag{10}$$

the conditional probability of the event $(\mathcal{E}_1^{i+1} = \varepsilon_1^{i+1}, L_i = \ell_i, A_1^{L_i} = a_1^{\ell_i})$ given the observations of samples $R_1^i = r_1^i$ and the prior timing estimates $\hat{\varepsilon}_1^i$. Note that $\hat{\varepsilon}_1^i$ is a function of prior observations r_1^{i-1}, so we can simply omit it from the conditioning. Nevertheless, we keep it as a "dummy" argument in the conditioning on the left-hand side of (10) in order to emphasize the fact that $(\hat{\varepsilon}_1, \ldots, \hat{\varepsilon}_i)$ are known before we make the $(i+1)$th estimate $\hat{\varepsilon}_{i+1}$.

Let $\tilde{\varepsilon}_1^{i+1}(i) = (\tilde{\varepsilon}_1(i), \tilde{\varepsilon}_2(i), \ldots, \tilde{\varepsilon}_{i+1}(i))$ denote the estimated timing error sequence obtained by maximizing the joint *a posteriori* probability in (10), after observing the first i samples r_1^i

$$\left(\tilde{\varepsilon}_1^{i+1}(i), \tilde{\ell}_i, \tilde{a}_1^{\tilde{\ell}_i}\right) = \arg \max_{\varepsilon_1^{i+1}, \ell_i, a_1^{\ell_i}} P\left(\varepsilon_1^{i+1}, a_1^{\ell_i} \;\middle|\; r_1^i, \hat{\varepsilon}_1^i\right). \tag{11}$$

Observe that it is possible to have $\tilde{\varepsilon}_1^i(i) \neq \hat{\varepsilon}_1^i$. Since the value ℓ_i is fully determined if ε_i and $\hat{\varepsilon}_i$ are known, we take advantage of this fact, and slightly abuse the notation to restate (11) as

$$\left(\tilde{\varepsilon}_1^{i+1}(i), \tilde{a}_1^{\tilde{\ell}_i}\right) = \arg \max_{\varepsilon_1^{i+1}, a_1^{\ell_i}} P\left(\varepsilon_1^{i+1}, a_1^{\ell_i} \;\middle|\; r_1^i, \hat{\varepsilon}_1^i\right). \tag{12}$$

From the sequence $\tilde{\varepsilon}_1^{i+1}(i)$, we derive $\hat{\varepsilon}_{i+1}$ by simply equating

$$\hat{\varepsilon}_{i+1} \triangleq \tilde{\varepsilon}_{i+1}(i). \tag{13}$$

Our task is to find the estimate $\hat{\varepsilon}_{i+1}$. Criterion (12) may suggest that the complexity of this task grows with i. However, as we will show in Section IV, the solution $\hat{\varepsilon}_{i+1} = \tilde{\varepsilon}_{i+1}(i)$ can be recursively obtained without explicitly solving for $\tilde{\varepsilon}_1^i(i)$ and $\tilde{a}_1^{\tilde{\ell}_i}$, and the complexity at each step does not grow with the elapsed time i.

D. Problem Formulation Using an Optimal-State Criterion

In formulating the second criterion for estimating $\mathcal{E}_{i+1}$, we take advantage of the assumption that $\mathcal{E}_i$ is a slowly time-varying process. Thus, we can assign to $\hat{\varepsilon}_{i+1}$ the most likely value of $\mathcal{E}_i$ after observing r_1^i. That is

$$\begin{aligned}\hat{\varepsilon}_{i+1} &= \arg\max_{\varepsilon_i} P\left(\varepsilon_i \,\middle|\, r_1^i, \hat{\varepsilon}_1^i\right) \\ &= \hat{\varepsilon}_i + \arg\max_{\tau_i} P\left(\tau_i \,\middle|\, r_1^i, \hat{\varepsilon}_1^i\right) \end{aligned} \tag{14}$$

where the second equality follows from (6) and the fact that the value of $\hat{\varepsilon}_i$ is given in the conditioning, so we only need to optimize over the variable τ_i. Equation (14) shows that the optimal loop is actually formulated as a first-order PLL, where $\arg\max_{\tau_i} P(\tau_i \,|\, r_1^i, \hat{\varepsilon}_1^i)$ represents the output of a TED. Note that the formulation in (14) is simpler than the one in (12)–(13). However, we shall see that the computational complexity of solving (14) is higher than that of solving (12)–(13). In Section V, we give a recursive trellis-based method for solving the problem in (14).

IV. Solution to the Optimal-Path Estimation Problem

A. The Conditional Density

We now study the *a posteriori* probability in (12) more carefully. From (2)–(4), we have

$$\begin{aligned}&\max_{\varepsilon_1^{i+1}, a_1^{\ell_i}} P\left(\varepsilon_1^{i+1}, a_1^{\ell_i} \,\middle|\, r_1^i, \hat{\varepsilon}_1^i\right) \\ &\overset{(a)}{=} \max_{\varepsilon_1^{i+1}, a_1^{\ell_i}} P\left(\varepsilon_1^i, a_1^{\ell_i} \,\middle|\, r_1^i, \hat{\varepsilon}_1^i\right) \cdot P\left(\varepsilon_{i+1} \,\middle|\, \varepsilon_1^i, a_1^{\ell_i}, r_1^i, \hat{\varepsilon}_1^i\right) \\ &\overset{(b)}{=} \max_{\varepsilon_1^{i+1}, a_1^{\ell_i}} P\left(\varepsilon_1^i, a_1^{\ell_i} \middle| r_1^i, \hat{\varepsilon}_1^i\right) \cdot P\left(\varepsilon_{i+1} | \varepsilon_1^i\right) \\ &\overset{(c)}{=} \max_{\varepsilon_1^i, a_1^{\ell_i}, \delta_{i+1}} P\left(\varepsilon_1^i, a_1^{\ell_i} \,\middle|\, r_1^i, \hat{\varepsilon}_1^i\right) \cdot P(\Delta_{i+1} = \delta_{i+1}) \\ &\overset{(d)}{=} P(\Delta_{i+1} = 0) \cdot \max_{\varepsilon_1^i, a_1^{\ell_i}} P\left(\varepsilon_1^i, a_1^{\ell_i} \,\middle|\, r_1^i, \hat{\varepsilon}_1^i\right) \\ &\overset{(e)}{=} (1 - p_1 - p_2) \cdot \max_{\varepsilon_1^i, a_1^{\ell_i}} P\left(\varepsilon_1^i, a_1^{\ell_i} \,\middle|\, r_1^i, \hat{\varepsilon}_1^i\right) \\ &\overset{(f)}{=} (1 - p_1 - p_2) \cdot \max_{\tau_1^i, a_1^{\ell_i}} P\left(\tau_1^i, a_1^{\ell_i} \,\middle|\, r_1^i, \hat{\varepsilon}_1^i\right). \end{aligned} \tag{15}$$

Equality *(a)* is the Bayes rule. Equalities *(b)* and *(c)* come from the facts that the process $\mathcal{E}_i$ is Markovian and the increment Δ_{i+1} is independent of all previous observations. Equalities *(d)* and *(e)* are consequences of (3) and (4). The last equality (f) is derived by using (6). The result in (15) indicates that

- in order to maximize the *a posteriori probability* in (12), it is sufficient to maximize $P(\tau_1^i, a_1^{\ell_i} \,|\, r_1^i, \hat{\varepsilon}_1^i)$, and
- the estimate given by (12) and (13) can be rewritten as

$$\left(\tilde{\tau}_1^i(i), \tilde{a}_1^{\tilde{\ell}_i}\right) = \arg\max_{\tau_1^i, a_1^{\ell_i}} P\left(\tau_1^i, a_1^{\ell_i} \,\middle|\, r_1^i, \hat{\varepsilon}_1^i\right) \tag{16}$$

and

$$\hat{\varepsilon}_{i+1} \triangleq \tilde{\varepsilon}_{i+1}(i) \overset{(g)}{=} \tilde{\varepsilon}_i(i) = \hat{\varepsilon}_i + \tilde{\tau}_i(i) \tag{17}$$

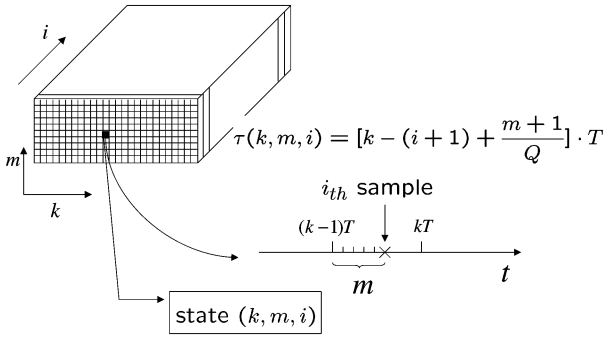

Fig. 4. Definition of the timing trellis states.

where (g) follows from (2) since $\tilde{\delta}_{i+1} = 0$, as shown by (d) in (15). Here $\tilde{\tau}_1^i(i)$ denotes the estimated sequence of residual timing errors after observing the first i samples r_1^i. Later, we will simply use $\tilde{\tau}_1^i$ when no confusion arises.

Equation (17) shows that, again, the optimal detector is actually a first-order PLL, where $\tilde{\tau}_i(i)$ represents the output of a TED. In the next subsection, we explain how to efficiently implement the TED by using state propagation rules in a joint timing and inter-symbol interference (ISI) trellis.

B. Trellis-Based Solution

In this section, we illustrate how to compute $\tilde{\tau}_i(i)$ in (17) through the recursive maximization of the *a posteriori* probability given in (16). In the sequel, to simplify the notation, we drop the argument in the parentheses of $\tilde{\tau}_i(i)$ and set $\tilde{\tau}_i \triangleq \tilde{\tau}_i(i)$. Since both the timing error and the channel have memory, it is helpful to give a graphical interpretation of how the timing error sequence and the channel state propagate. We first define a timing trellis.

Timing Error States: From our previous assumption (3), the sampling instants must fall on integer multiples of $(T)/(Q)$. Now, we partition the time axis into nonoverlapping semi-open intervals $((k-1)T, kT]$, where $k \in \mathbb{Z}^+$. Each interval corresponds to a transmitted symbol. There are Q possible positions (quantization levels) where a sample can be taken within each interval.

We construct the timing trellis by representing each state (or, node) in the trellis with three variables (k, m, i). The physical interpretation of the state is: the ith sample falls inside the kth input interval $((k-1)T, kT]$ at the $(m+1)$th quantization level, $m \in \{0, 1, \ldots, Q-1\}$, as illustrated in Fig. 4. We observe that each state (k, m, i) is in a one-to-one correspondence with the residual timing error τ_i for the ith sample, i.e.,

$$\tau_i = \tau(k, m, i) = \left[k - (i+1) + \frac{m+1}{Q}\right] \cdot T \tag{18}$$

where $m \in \{0, 1, \ldots, Q-1\}$. Conversely, for a given sample r_i, any value of τ_i also uniquely corresponds to a state (k, m, i).

Definition 4 [The ith Plane]: We define the set of states (k, m, i) with the same index i as the *ith plane*. Different states (nodes) in the ith plane are mapped to different values of τ_i. ■

We next need to figure out how the timing states propagate from the $(i-1)$th (predecessor) plane to the ith (successor)

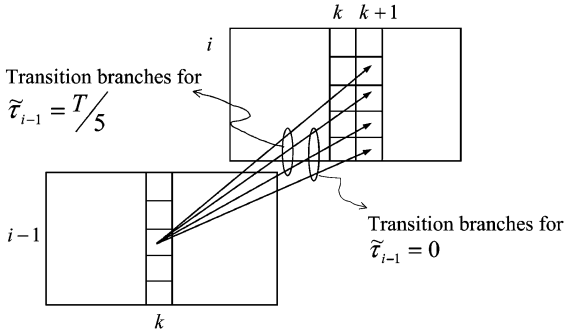

Fig. 5. Illustration of the state propagation in the timing trellis.

plane after the sample r_i is obtained. From (2), (6), and (17), we have

$$\begin{aligned}\mathcal{T}_i &= \mathcal{E}_i - \hat{\varepsilon}_i \\ &= \mathcal{E}_{i-1} + \Delta_i - (\hat{\varepsilon}_{i-1} + \tilde{\tau}_{i-1}) \\ &= \mathcal{T}_{i-1} + \Delta_i - \tilde{\tau}_{i-1}. \end{aligned} \tag{19}$$

Note that when the sample r_{i-1} is obtained, we can compute $\tilde{\tau}_{i-1}$, so $\tilde{\tau}_{i-1}$ can be regarded as a constant in (19). We observe two things from (19). First, starting from a given predecessor node in the $(i-1)$th plane, we can only go to three possible successor nodes in the ith plane, corresponding to the three possible values of Δ_i, respectively [see (3)]. Second, the branches that link the states in the predecessor plane $i-1$ to the states in the successor plane i clearly depend on the value of $\tilde{\tau}_{i-1}$. Thus, the state propagation needs to be dynamically established at different steps i, after computing $\tilde{\tau}_{i-1}$.

For example, consider the case where we have $Q = 5$ levels of quantization; see Fig. 5 for reference to this example. Suppose that the predecessor state is $(k, 2, i-1)$, which means that the $(i-1)$th sample is taken at time $(k-1)T+(3)/(5)T$. If the best estimate of the residual timing error is $\tilde{\tau}_{i-1} = 0$, then the possible successor states in the ith plane are $((k+1), 1, i), ((k+1), 2, i), ((k+1), 3, i)$, as shown in Fig. 5. However, if the best estimate of the residual timing error is $\tilde{\tau}_{i-1} = (T)/(5)$, then by (18) and (19), the successor states in the ith plane will be shifted by 1 node. That is, they are $((k+1), 2, i), ((k+1), 3, i), ((k+1), 4, i)$, as illustrated in Fig. 5.

Joint Data and Timing Error States: In order to construct a graphical model that can *jointly* represent the timing error and the *transmitted data*, we need to expand our previous definition of the states in the timing trellis to a *joint* timing-data trellis.

From our finite-support assumption (7), at most

$$D = 2 \cdot q \tag{20}$$

data symbols are needed to calculate the value of each sample as shown by (8). They are $(a_{k-q}, \ldots, a_{k+q-1})$. To account for these D data symbols as well as the timing error, we expand the state variable m into a vector $(a_{k-q}, \ldots, a_{k+q-1}, m)$. Since $0 \le m < Q$, and $a_i \in \{-1, 1\}$ are binary integers, we can define a one-to-one mapping

$$(a_{k-q}, \ldots, a_{k+q-1}, m) \overset{1-to-1}{\longleftrightarrow} \bar{m} \tag{21}$$

where

$$\bar{m} = m + Q\left[\sum_{j=k-q}^{k+q-1} 2^{k+q-1-j}\left(\frac{a_j+1}{2}\right)\right]. \tag{22}$$

Thus, in the new trellis, the state is defined as follows.

Definition5: [Joint Timing and ISI State]: A joint timing and ISI state is denoted by the triplet $s_i = (k, \bar{m}, i)$. ■

With this definition, the full joint trellis has the following properties.

- For each state (node) $(k, \bar{m}, i)$ in the full trellis, the timing state is captured by $m = (\bar{m} \bmod Q)$. Thus, the corresponding residual timing error is

$$\begin{aligned}\tau_i &= \tau(k, \bar{m}, i) \\ &= \left[k - (i+1) + \frac{(\bar{m} \bmod Q) + 1}{Q}\right] \cdot T. \end{aligned} \tag{23}$$

  The ISI information of the state $(k, \bar{m}, i)$ is captured by

$$\sum_{j=k-q}^{k+q-1} 2^{k+q-1-j} \cdot \left(\frac{a_j+1}{2}\right) = \left\lfloor \frac{\bar{m}}{Q} \right\rfloor. \tag{24}$$

- Each valid path through the trellis that leads to a state in the ith plane uniquely determines a data sequence a_1^{k+q-1} and a residual timing error sequence τ_1^i simultaneously. Conversely, given a timing sequence τ_1^i and a data sequence a_1^{k+q-1}, we can uniquely find a path leading to state $(k, \bar{m}, i)$.

Recursive State Propagation: Let us denote s_i as an arbitrary state (node) in the ith plane of the full trellis, i.e., $s_i \in \{(k_i, \bar{m}_i, i)\}$, where $k_i \in \mathbb{Z}^+$ and $\bar{m}_i \in \{0, 1, \ldots, Q \cdot 2^D - 1\}$. Since the state sequence s_1^i and the joint sequence $(\tau_1^i, a_1^{\ell_i})$ uniquely determine each other, we have

$$\max_{\tau_1^i, a_1^{\ell_i}} P\left(\tau_1^i, a_1^{\ell_i} \,\middle|\, r_1^i, \hat{\varepsilon}_1^i\right) = \max_{s_1^i} P\left(s_1^i \,\middle|\, r_1^i, \hat{\varepsilon}_1^i\right). \tag{25}$$

Thus, the estimation rules given by (16) and (17) are equivalent to

$$\tilde{s}_1^i = \arg\max_{s_1^i} P\left(s_1^i \,\middle|\, r_1^i, \hat{\varepsilon}_1^i\right) \tag{26}$$

$$\hat{\varepsilon}_{i+1} = \hat{\varepsilon}_i + \tau(\tilde{s}_i) \tag{27}$$

where $\tau(\tilde{s}_i)$ is obtained by substituting $\tilde{s}_i$ into (23).

We next manipulate the term on the right-hand side of (26) to derive a recursive formula. Using the Bayes rule, we have

$$P\left(s_1^i \,\middle|\, r_1^i, \hat{\varepsilon}_1^i\right) = C \cdot \gamma(s_{i-1}, s_i, r_i) \cdot P(s_1^{i-1} \,|\, r_1^{i-1}, \hat{\varepsilon}_1^{i-1}) \tag{28}$$

where

$$\gamma(s_{i-1}, s_i, r_i) = P(r_i \,|\, s_i) \cdot P(s_i \,|\, s_{i-1}) \tag{29}$$

$$C = \frac{P\left(r_1^{i-1}, \hat{\varepsilon}_1^{i-1}\right)}{P\left(r_1^i, \hat{\varepsilon}_1^i\right)} = \frac{P\left(r_1^{i-1}\right)}{P\left(r_1^i\right)}. \tag{30}$$

(The proof is provided in the Appendix.) We notice that C does not depend on s_i and can be viewed as a constant at the ith step. Since this constant C does not have any effect on the maximization in (26), we can simply ignore it. The recursive formula (28) and the estimation rule (26) prompt the following state metric definition in log-domain.

Definition 6: [Accumulated Metric $c(s_i)$]: We define the metric $c(s_i)$ of an arbitrary state $s_i \in \{(k_i, \bar{m}_i, i)\}$ in the ith plane as

$$c(s_i) \triangleq \min_{s_{i-1}} [c(s_{i-1}) - \ln \gamma(s_{i-1}, s_i, r_i)] \tag{31}$$

with the initial condition

$$c(s_0) = \begin{cases} 0, & \text{if } s_0 = (0, Q-1, 0), \\ +\infty, & \text{otherwise} \end{cases}. \tag{32}$$

■

The term $-\ln \gamma(s_{i-1}, s_i, r_i)$ can be viewed as the branch metric between states s_{i-1} and s_i in two adjacent planes, and $c(s_i)$ is the minimum accumulated metric among all the paths in the full trellis that end in state s_i.

Equations (28) and (31) indicate that our estimation rules (26) and (27) are equivalent to

$$\tilde{s}_i = \arg\min_{s_i} c(s_i) \tag{33}$$

$$\hat{\varepsilon}_{i+1} = \hat{\varepsilon}_i + \tau(\tilde{s}_i). \tag{34}$$

where $\tau(\tilde{s}_i)$ is calculated by (23).

Recursive Estimation Algorithm I: We can now formulate the recursive timing recovery algorithm based on the optimal-path criterion.

1) The system is assumed to be perfectly synchronized at time $i = 0$ (i.e., $\varepsilon_0 = 0$), and we assume $a_k = 0$ for $k < q$. The initial condition of our algorithm is (32). In practice, this assumption is satisfied by using preambles. (Initialization with nonperfect synchronization at time $i = 0$ is also possible.)
2) Whenever the loop detector receives the ith sample r_i, we calculate all the state metrics $c(s_i)$ in the ith plane according to (31).
3) The next-step timing error estimate $\hat{\varepsilon}_{i+1}$ is computed by (33) and (34).
4) For all states s_{i-1}, delete $c(s_{i-1})$ from the memory.
5) Use $\hat{\varepsilon}_{i+1}$ to take the $(i+1)$th sample, and go back to step 2.

Comments: The above recursive estimation algorithm propagates similarly as the standard Viterbi algorithm [22], however, we *never need to trace back* as we do in the Viterbi algorithm. Thus, a trellis structure is actually not necessary. At any step i, we only need to remember the ith plane metrics $c(s_i)$ and thus conserve the memory. This explains step 4 in the above algorithm. The complexity and memory cost of the algorithm is discussed in Section VII.

V. Solution to the Optimal-State Estimation Problem

We consider the *a posteriori probability* in (14). From our previous definition of the state s_i, we get

$$P\left(\tau_i \mid r_1^i, \hat{\varepsilon}_1^i\right) = \sum_{s_i:\tau(s_i)=\tau_i} P(s_i | r_1^i, \hat{\varepsilon}_1^i) = \sum_{s_i:\tau(s_i)=\tau_i} \left[\sum_{s_1^{i-1}} P\left(s_1^i | r_1^i, \hat{\varepsilon}_1^i\right) \right]. \tag{35}$$

The summation within the brackets in (35) can be derived recursively by using (28)

$$\begin{aligned} &\sum_{s_1^{i-1}} P\left(s_1^i \mid r_1^i, \hat{\varepsilon}_1^i\right) \\ &\quad = C \cdot \sum_{s_1^{i-1}} \gamma(s_{i-1}, s_i, r_i) P\left(s_1^{i-1} \mid r_1^{i-1}, \hat{\varepsilon}_1^{i-1}\right) \\ &\quad = C \cdot \sum_{s_{i-1}} \gamma(s_{i-1}, s_i, r_i) \left[\sum_{s_1^{i-2}} P\left(s_1^{i-1} \mid r_1^{i-1}, \hat{\varepsilon}_1^{i-1}\right) \right] \end{aligned} \tag{36}$$

where C is given by (30). As before, C does not depend on the states and is considered as a constant at the ith step. We therefore ignore this constant and introduce the following recursive accumulated metric definition based on (36).

Definition 7: [Accumulated Metric $c'(s_i)$]: We define the metric $c'(s_i)$ of an arbitrary state $s_i \in \{(k_i, \bar{m}_i, i)\}$ in the ith plane as

$$c'(s_i) \triangleq \sum_{s_{i-1}} \gamma(s_{i-1}, s_i, r_i) \cdot c'(s_{i-1}) \tag{37}$$

with the initial condition

$$c'(s_0) = \begin{cases} 1, & \text{if } s_0 = (0, Q-1, 0) \\ 0, & \text{otherwise} \end{cases}. \tag{38}$$

■

Based on the above definition and the result in (35), the estimation rule given by (14) is equivalent to

$$\hat{\varepsilon}_{i+1} = \hat{\varepsilon}_i + \arg\max_{\tau_i} \sum_{s_i:\tau(s_i)=\tau_i} c'(s_i). \tag{39}$$

Recursive Estimation Algorithm II: We now formulate the recursive timing recovery algorithm based on the optimal-state criterion.

1) The system is assumed to be perfectly synchronized at time $i = 0$ (i.e., $\varepsilon_0 = 0$), and we assume $a_k = 0$ for $k < q$. The initial condition of our algorithm is (38). In practice, this assumption is satisfied by using preambles. (Initialization with nonperfect synchronization at time $i = 0$ is also possible.)

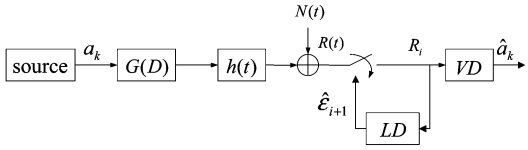

Fig. 6. The un-coded system used in simulation. LD represents the loop detector. VD represents the Viterbi detector for data detection.

2) Whenever the loop detector receives the ith sample r_i, for $i \geq 1$, we calculate all the state metrics $c'(s_i)$ in the ith plane according to (37). Normalize the state metrics such that $\sum c'(s_i) = 1$.

3) The next-step timing error estimate $\hat{\varepsilon}_{i+1}$ is computed by (39).

4) For all states s_{i-1}, delete $c'(s_{i-1})$ from the memory.

5) Use $\hat{\varepsilon}_{i+1}$ to take the $(i+1)$th sample, and go back to step 2.

Comments: We notice that the above algorithm actually uses the same state (or trellis) structure as the one in Section IV. The state propagation in the algorithm is similar to the forward recursion of the Bahl–Cocke–Jelinek–Raviv (BCJR) algorithm [23]. Since we *do not need backward recursions*, the previous state metrics are always deleted in step 4. The complexity and memory cost of this algorithm is discussed in Section VII.

VI. Simulation Results

To assess the quality of the derived timing recovery methods, both algorithms in Sections IV and V are compared to the conventional M&M timing recovery loop as well as to the perfect timing scenario. The data symbols are generated by an equiprobable binary source, and are independent and identically distributed (i.i.d.). The symbols are first passed through the filter $G(D) = 1 - D^2$, as shown by Fig. 6. We assume that $h(t)$ is a truncated $\mathrm{sinc}(\cdot)$ function of the form $h(t) = \mathrm{sinc}((t)/(T))[u(t+T) - u(t-T)]$, where $u(t)$ is the unit step function. If there is no timing error, the channel is equivalent to the $PR4$ channel [24]. In the simulations, we create waveforms according to (5), where the timing error $\{\mathcal{E}_i\}$ injected into the waveform is a Gaussian independent increment process $\mathcal{E}_i = \mathcal{E}_{i-1} + W_i$, and $W_i \sim \mathcal{N}(\mu_w, \sigma_w^2)$ are i.i.d. Gaussian random variables. Therefore, this timing error process has a frequency offset

$$\Delta f_1 = \frac{1}{T + \mu_w} - \frac{1}{T}.$$

For the purpose of designing the loop detector for this timing error, we approximate the timing error by the model in (2) and (3). This discrete model has a frequency offset

$$\Delta f_2 = \frac{1}{T + \frac{T}{Q}(p_1 - p_2)} - \frac{1}{T}.$$

In practice, the parameters p_1 and p_2 can be iteratively estimated from the received signal (or training symbols) using the Baum–Welch algorithm [25]. (Note that we could use a more sophisticated Markov timing model, but it suffices to use the simple one in (2) and (3) to illustrate the principle.) The final *residual* frequency offset of the system is

$$\begin{aligned}\Delta f &= \Delta f_1 - \Delta f_2 \\ &= \frac{1}{T + \mu_w} - \frac{1}{T + \frac{T}{Q}(p_1 - p_2)} \\ &\approx -\frac{1}{T}\left(\frac{\mu_w}{T} - \frac{p_1 - p_2}{Q}\right). \end{aligned} \tag{40}$$

This approximation is based on the assumption that $|\mu_w| \ll T$ and $|(p_1 - p_2)/(Q)| \ll 1$. In our simulations, we set $p_1 = p_2$. Therefore, the residual frequency offset is $\Delta f \approx -(\mu_w)/(T^2)$. Since the method works at *baud-rate*, we compare it with the standard second-order phase-locked loop (PLL) using the Mueller and Müller (M&M) detector [8], for systems with and without residual frequency offset.

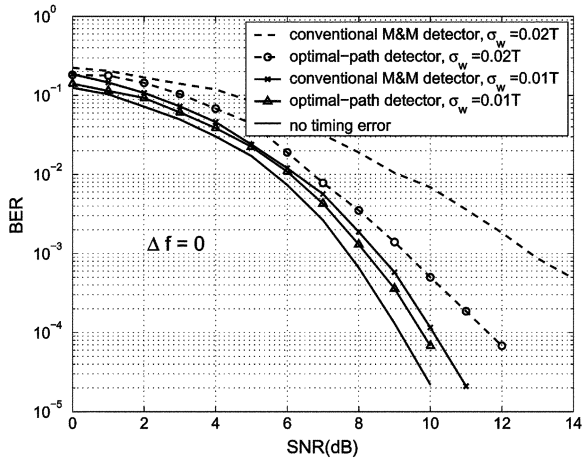

Fig. 7. Bit-error-rate (BER) performance using the sequence-based state propagation detector. Timing error increments are i.i.d Gaussian random variables.

Fig. 7 compares the bit-error-rate (BER) performance of the optimal-path timing error detector when there is *no residual frequency offset*, i.e., $\mu_w = 0$ and $p_2 = p_1$, to the standard first-order PLL with the M&M detector (a first-order PLL suffices since there is no frequency offset). Both the filter coefficient in the M&M detector, as well as the delay length of the Viterbi detector inside the PLL were exhaustively optimized for every SNR, respectively. The quantization level for the proposed loop detector is fixed to $Q = 10$. We observe that for $\sigma_w = 0.01T$, the proposed optimal-path timing error detector barely outperforms the M&M phase-locked loop. However, as the timing error increases to $\sigma_w = 0.02T$, there is a large performance gain attained by the new detector in all SNR regions.

Fig. 8 compares the BER performance of the optimal-state timing error detector when there is *no residual frequency offset* to the standard first-order PLL with the M&M detector. The simulation parameters were chosen to be identical to those of Fig. 7. Again, we observe that for $\sigma_w = 0.01T$, the proposed optimal-state timing error detector barely outperforms the conventional PLL. However, as the timing error increases to $\sigma_w = 0.02T$, there is a large performance gain attained by the new detector in all SNR regions.

Fig. 9 compares the BER performance of the proposed recursive timing error detectors when there is *residual frequency*

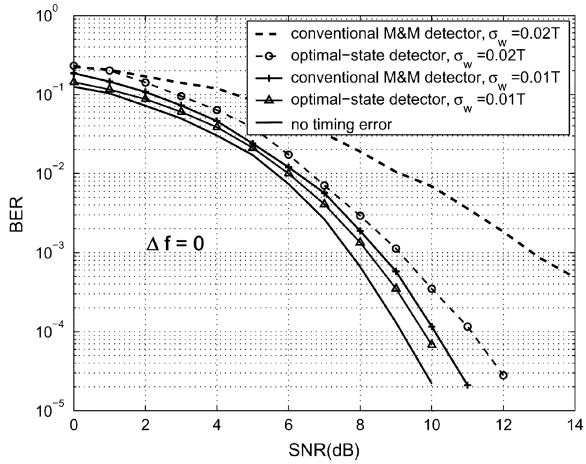

Fig. 8. BER performance using the single-value-based state propagation detector. Timing error increments are i.i.d Gaussian random variables.

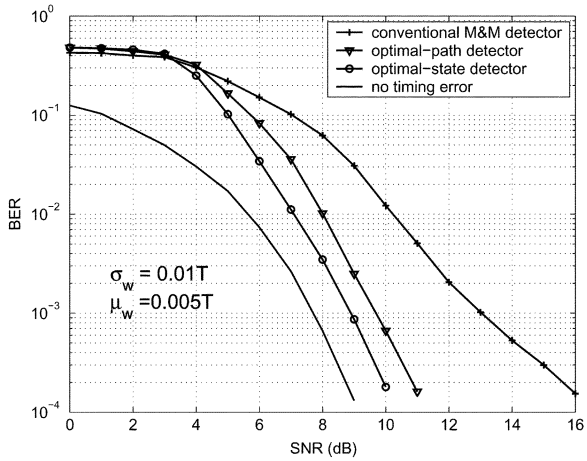

Fig. 9. BER performance using the recursive detectors when there is residual frequency offset in the system. Timing error increments are i.i.d Gaussian random variables with nonzero mean.

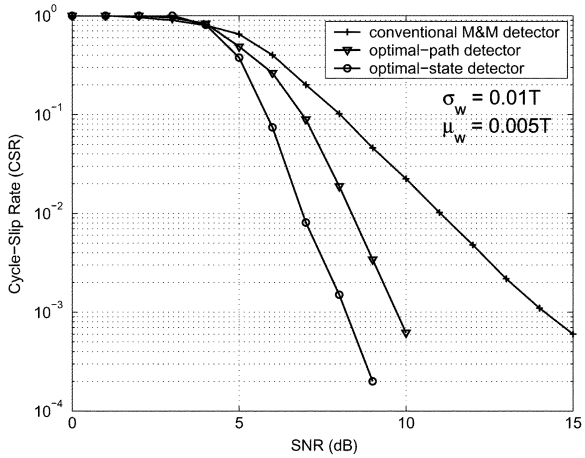

Fig. 10. Cycle-slip rate (CSR) performance using the recursive detectors when there is residual frequency offset in the system. Timing error increments are i.i.d Gaussian random variables with nonzero-mean.

offset, i.e., $\Delta f \neq 0$, to the standard second-order PLL with the M&M detector. The parameters of the timing error increment are set to $\sigma_w = 0.01T$ and $\mu_w = 0.005T$. Both filter coefficients in the M&M detector, as well as the delay length of the Viterbi detector before the M&M detector were exhaustively optimized for every SNR, respectively. We observe that both of the proposed recursive detectors are less sensitive to the frequency offset, and provide large performance gains over the conventional PLL.

Sector-by-sector simulation results show that the probability of having a cycle-slip is greatly reduced by using the proposed recursive timing error detectors. Fig. 10 compares the *cycle-slip rate* (fraction of sectors that experience cycle-slips) of the proposed timing recovery loops to the second-order PLL with the M&M detector. We notice that the optimal-state detector outperforms the optimal-path detector when there is residual frequency offset in the system. The optimal-state algorithm has a lower cycle-slip rate. A simple explanation to this is that in the optimal-state algorithm, we explicitly calculate the probability of each *value* of the timing error τ_i (or each state) at every step. This is achieved by summing over all possible paths at each state during the state propagation, which is similar as the forward recursion of the BCJR algorithm [23]. In the optimal-path algorithm, what we compute is the probability of each joint data and timing error *sequence* (or path). We only keep the most likely path at each state during the state propagation, and delete the other paths. Therefore, we may encounter state error propagation when running the optimal-path detector.

VII. Discussion

As we have discussed in Sections IV and V, neither algorithm requires *backward recursions* or *trace-back operations* to make the decision. Hence, previous state metrics are always deleted from the memory in step 4 of each algorithm. Therefore, the memory costs of the proposed algorithms do not grow with block length n. Further, it is not necessary to evaluate all the state metrics $c(s_i)$ (or $c'(s_i)$) in the ith plane in step 2. In practice, if we assume that the timing error estimate is not too bad, i.e., $|\tau_i| = |\mathcal{E}_i - \hat{\varepsilon}_i| \leq d \cdot T$ for $1 \leq i \leq \ell$ and $d \in \mathbb{Z}^+$ is a fixed integer, we only need to consider those states $s_i = (k, \bar{m}, i)$ that satisfy $|k - i| \leq d$. Simulation results show that for most cases, it is sufficient to choose $d = 2$ without any loss in performance. The amount of memory (i.e., the number of state metrics) required for both of the proposed algorithms is $2d \cdot Q \cdot 2^{2q}$, which is $4 \cdot d \cdot Q$ times the amount of memory of the conventional Viterbi algorithm.

Now we look at the computational complexity of the two algorithms. Obviously, the number of computations at each step is proportional to the memory cost $2d \cdot Q \cdot 2^{2q}$. However, for the optimal-path algorithm, we only need the operations of "addition" and "comparison" during the state propagation if we operate in the log-domain, as shown by (31). For the optimal-state algorithm, turning to the log-domain will not help, because of the "addition" in (37). Therefore, we will need the "multiplication" operation, which is more complex in terms of hardware implementation.

Both algorithms adopt the simple Markov model shown in Fig. 3, which only allows state transitions to the adjacent neighboring states. More accurate Markov processes that allow state transitions to nonadjacent states can be adopted without changing the nature of the algorithms. Algorithms derived based on such Markov models would be able to track faster

changes of the timing errors, albeit at the cost of increasing the complexity.

VIII. Conclusion

In this work, we proposed two optimization criteria for timing error estimation from baud-rate samples, and derived state propagation based timing recovery algorithms, under the assumption that the timing error can be modeled as a discrete random walk process. We compared the bit error rate performance of the newly derived detectors to the conventional timing recovery loop (with a Mueller and Müller loop detector), and observed a considerable performance gain when the timing error is large or when there is residual frequency offset in the system. There is only a slight performance gain when the timing error is small and when there is no residual frequency offset. The performances of the proposed timing recovery methods in *coded* systems that can operate at very low SNRs need to be further studied.

Appendix

Proof of (28): By using the Bayes rule, we have

$$\begin{aligned} &P\left(s_1^i \,\middle|\, r_1^i, \hat{\varepsilon}_1^i\right) \\ &\quad = P\left(s_1^i, r_1^i, \hat{\varepsilon}_1^i\right) / P\left(r_1^i, \hat{\varepsilon}_1^i\right) \\ &\quad = P\left(s_1^{i-1} \,\middle|\, r_1^{i-1}, \hat{\varepsilon}_1^{i-1}\right) \frac{P(r_1^{i-1}, \hat{\varepsilon}_1^{i-1})}{P(r_1^i, \hat{\varepsilon}_1^i)} \\ &\qquad \cdot P\left(s_i, r_i \,\middle|\, s_1^{i-1}, r_1^{i-1}, \hat{\varepsilon}_1^{i-1}\right). \end{aligned} \tag{41}$$

The last term in (41) can be further simplified as

$$\begin{aligned} &P\left(s_i, r_i \middle| s_1^{i-1}, r_1^{i-1}, \hat{\varepsilon}_1^{i-1}\right) \\ &\quad = P\left(s_i \middle| s_1^{i-1}, r_1^{i-1}, \hat{\varepsilon}_1^{i-1}\right) \cdot P\left(r_i \,\middle|\, s_1^i, r_1^{i-1}, \hat{\varepsilon}_1^{i-1}\right). \end{aligned} \tag{42}$$

Since s_i contains both the timing and ISI information for the ith sample r_i, we have

$$P\left(r_i \,\middle|\, s_1^i, r_1^{i-1}, \hat{\varepsilon}_1^{i-1}\right) = P(r_i \,|\, s_i). \tag{43}$$

Next by using the one-to-one correspondence between the state s_i on one side and the timing error and ISI information as a pair on the other, we have

$$\begin{aligned} &P\left(s_i \,\middle|\, s_1^{i-1}, r_1^{i-1}, \hat{\varepsilon}_1^{i-1}\right) \\ &\quad = P\left(\tau_i, a_{\ell_i - D+1}^{\ell_i} \,\middle|\, \tau_1^{i-1}, a_1^{\ell_{i-1}}, r_1^{i-1}, \hat{\varepsilon}_1^{i-1}\right) \\ &\quad \overset{(h)}{=} P\left(\tau_i \,\middle|\, \tau_{i-1}, \hat{\varepsilon}_{i-1}^i\right) \cdot P\left(a_{\ell_{i-1}+1}^{\ell_i} \,\middle|\, a_{\ell_{i-1}}\right) \qquad (44) \\ &\quad \overset{(i)}{=} P(s_i \,|\, s_{i-1}) \qquad (45) \end{aligned}$$

where (h) follows from the assumption that $\mathcal{E}_i$ is Markov, a_k are i.i.d., and the fact that $\hat{\varepsilon}_i$ is determined by r_1^{i-1}. Equality (i) is based on the observation that s_{i-1} actually contains all the information in the conditioning. By substituting (42), (43), and (45) into (41), we obtain the recursive formula (28).

Acknowledgment

This work was supported by a grant from Seagate Technology. W. Zeng was at Seagate when parts of this work were conducted.

References

[1] H. Meyr, M. Moeneclaey, and S. A. Fechtel, *Digital Communication Receivers*. New York: Wiley, 1998.
[2] H. Kobayashi, "Simultaneous adaptive estimation and decision algorithm for carrier modulated data transmission systems," *IEEE Trans. Commun.*, vol. COM-19, no. 3, pp. 268–280, Jun. 1971.
[3] H. L. Van Trees, *Detection, Estimation and Modulation Theory*. New York: Wiley, 1968.
[4] S. U. Qureshi and E. E. Newhall, "Adaptive receiver for data transmission over time-dispersive channels," *IEEE Trans. Inf. Theory*, vol. 19, pp. 448–457, Jul. 1973.
[5] G. Ascheid and H. Meyr, "Maximum likelihood detection and synchronization by parallel digital signal processing," in *Proc. IEEE GLOBECOM*, 1993, pp. 804–810.
[6] L. E. Franks, "Carrier and bit synchronization in data communication—A tutorial review," *IEEE Trans. Commun.*, vol. 28, no. 8, pp. 1107–1121, Aug. 1980.
[7] M. H. Meyers and L. E. Franks, "Joint carrier phase and symbol timing for pam systems," *IEEE Trans. Commun.*, vol. 28, pp. 1121–1129, Aug. 1980.
[8] K. H. Mueller and M. Muller, "Timing recovery in digital synchronous data receivers," *IEEE Trans. Commun.*, vol. 24, no. 5, pp. 516–531, May 1976.
[9] P. F. Driessen, "DPLL bit synchronizer with rapid acquisition using adaptive Kalman filtering techniques," *IEEE Trans. Commun.*, vol. 42, no. 9, pp. 2673–2675, Sep. 1994.
[10] A. Patapoutian, "On phase-locked loops and Kalman filters," *IEEE Trans. Commun.*, vol. 47, no. 5, pp. 670–672, May 1999.
[11] P. M. Aziz and S. Surendran, "Symbol rate timing recovery for higher order partial response channels," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 4, pp. 635–648, Apr. 2001.
[12] W. C. Lindsey and M. K. Simon, *Telecommunication Systems Engineering*. Eaglewood Cliffs, NJ: Prentice-Hall, 1973.
[13] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 2001.
[14] L. L. Scharf, D. D. Cox, and C. J. Masreliez, "Modulo-2π phase sequence estimation," *IEEE Trans. Inf. Theory*, vol. 26, no. 5, pp. 615–620, Sep. 1980.
[15] J. Dauwels and H.-A. Loeliger, "Joint decoding and phase estimation: An exercise in factor graphs," in *Proc. IEEE Int. Symp. Information Theory*, Yokohama, Japan, Jun. 2003, p. 231.
[16] J. R. Barry, A. Kavčić, S. W. McLaughlin, A. Nayak, and W. Zeng, "Iterative timing recovery," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 89–102, Jan. 2004.
[17] A. R. Nayak, J. R. Barry, and S. W. McLaughlin, "Joint timing recovery and turbo equalization for coded partial response channels," *IEEE Trans. Magn.*, vol. 38, no. 5, pp. 2295–2297, Sep. 2002.
[18] P. Kovintavewat, J. R. Barry, M. F. Erden, and E. M. Kurtas, "Robustness of per-survivor iterative timing recovery in perpendicular recording channels," in *Dig. IEEE Int. Magnetics Conf. (INTERMAG)*, Apr. 4–8, 2005, pp. 1613–1614.
[19] A. R. Nayak, J. R. Barry, and S. W. McLaughlin, "Optimal placement of training symbols for frequency acquisition: A Cramer–Rao bound approach," *IEEE Trans. Magn.*, vol. 42, no. 6, pp. 1730–1742, Jun. 2006.
[20] W. Zeng and A. Kavčić, "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.
[21] J. Liu, H. Song, and B. V. K. Vijaya Kumar, "Timing acquisition for low-SNR data storage channels," *IEEE Trans. Magn.*, vol. 39, no. 5, pp. 2558–2560, Sep. 2003.
[22] D. G. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, 1973.
[23] 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, pp. 284–287, Mar. 1974.
[24] P. Kabal and S. Pasupathy, "Partial-response signaling," *IEEE Trans. Commun.*, vol. COM-23, no. 9, pp. 921–934, Sep. 1975.
[25] W. Zeng, R. Motwani, and A. Kavčić, "Estimation of the timing error process parameters using the Baum-Welch algorithm," *IEEE Trans. Magn.*, vol. 42, no. 2, pp. 194–199, Feb. 2006.

Manuscript received June 7, 2005; revised July 7, 2006 (e-mail: Wei.Zeng@post.harvard.edu).