

Timing Recovery in Conjunction With Maximum Likelihood Sequence Detection in the Presence of Intersymbol Interference

Jaekyun Moon, *Fellow, IEEE*, and Jaewook Lee, *Student Member, IEEE*

Abstract—A decision-directed timing recovery scheme is discussed that maintains and updates a separate phase estimate path for each survivor data path in the maximum-likelihood (ML) sequence detection of intersymbol interference (ISI) channels. For each survivor path, the phase estimate is updated recursively using symbol-decisions implied in the path. Unlike in the existing per-survivor timing recovery approach, there exists a single global timing loop that operates on a single stream of delayed phase estimate samples released from the merged survivor path or an up-to-date best survivor path. Loop analysis, tracking behavior simulation and BER simulation validate the proposed approach.

Index Terms—Timing recovery, synchronization, maximum-likelihood sequence detection, inter-symbol interference (ISI), phase-locked loop, cycle slip.

I. INTRODUCTION

CONTINUED demands for improved throughput rates force communication systems to operate at low signal-to-noise-ratios (SNRs). At low SNRs, however, typical detectors produce excessive bit errors. Powerful codes that have been proposed recently can correct many of these errors and bring down the bit error rate (BER) of digital communication systems to a satisfactory level. These codes are based on iterative processing of soft reliability of transmitted bits. Examples include turbo codes [1] and low density parity check codes [2]. However, the detector can only operate with proper recovery of timing of each bit relative to others [3], [4]. Furthermore, the timing recovery operation also typically depends on the estimated bits fed by the detector [5]–[8]. This is true whether the timing error detector (TED) portion of the timing recovery loop is derived from the timing function approach of Mueller and Muller [6], [9], the minimum mean squared error (MMSE) principle [10] or the maximum likelihood (ML) criterion [11]. An unfortunate consequence is that if the BER is high at the detector output, as is the case at low SNRs, timing recovery loops cannot function well. This in turn degrades the detector performance further, causing even those powerful codes to fail to correct enough erroneous bits. The maximum likelihood sequence detector (MLSD) is capable of achieving

improved BERs even when the SNR is relatively low, but with this type of detector reliable decisions become available only after a considerable delay. Such a decision delay also prevents reliable timing recovery in the presence of rapid timing phase variations.

In per-survivor timing recovery of [12], [13], joint ML estimation is done by generating a sampling phase error estimate for each branch in the Viterbi trellis using the local bit sequence implied in the given branch and readjusting the sampling position of the read signal for that branch for the next symbol cycle. The scheme effectively runs a separate parallel phase-locked loop (PLL) as well as a sampling device for each of the branches in the trellis. This approach allows operation of the timing loop at low SNRs, but maintaining separate timing loops for all survivor paths leads to a complex architecture. If the latency and complexity associated with taking multiple passes over the entire observation sequence can be tolerated during equalization, as is the case in iterative turbo equalization [14], then timing recovery can also be incorporated into the existing turbo equalizer structure with relatively low added complexity, to closely approach the ideal synchronous performance even at low SNRs and virtually eliminate the cycle slips [15].

In this paper, we elaborate on the idea first discussed in [16], aiming at the same objective of providing a reliable timing loop operation at low SNRs. The idea is based on performing phase estimation for each branch separately; the delayed phase estimate associated with the merged survivor path is released to a global low pass filter whose output drives a voltage-controlled oscillator (VCO) that in turn controls the sampling time of a single analog-to-digital converter (ADC). Thus, there exist parallel phase estimates incorporated into the Viterbi trellis branch metric computation but only one global PLL exists effectively. The phase error detector of [16] is also different from existing techniques in that the phase is recursively estimated along each tentative data path implied within the Viterbi detector.

The timing function itself is derived from the least square solution that minimizes error between the observation sequence and the clean expected signal sequence and can be shown to be equivalent to maximizing the conditional probability density function of the observation sequence given the phase offset. The proposed scheme can be incorporated into any Viterbi-like trellis or tree search algorithm. This paper also presents jitter analysis for the proposed scheme as well as the existing Mueller and Muller (MM) and MMSE schemes. Numerical results are provided to facilitate the direct comparison of the proposed scheme with existing schemes.

Manuscript received May 5, 2003; revised February 29, 2008. First published April 21, 2008; current version published October 29, 2008. This paper was recommended by Associate Editor W. Namgoong.

The authors are with Communications and Data Storage Laboratory, Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: moon@umn.edu; leex2244@umn.edu).

Digital Object Identifier 10.1109/TCSI.2008.920985

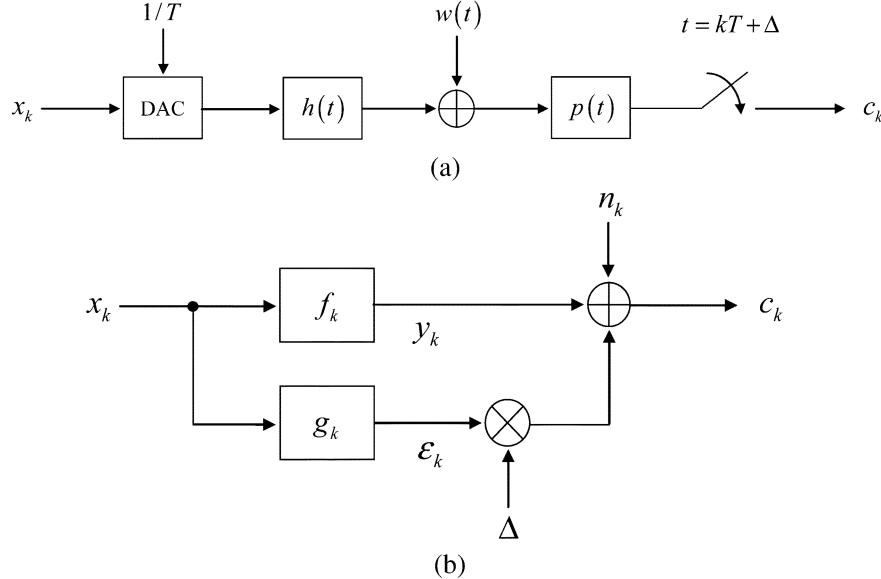


Fig. 1. Baseband model of the ISI communication channel. (a) Continuous-time model. (b) Discrete-time model.

II. SYSTEM MODEL

The proposed approach starts from a continuous-time intersymbol interference (ISI) channel model shown in Fig. 1(a), where the input bit stream x_k is converted into a continuous-time impulse train at a rate of $1/T$ bits per second, applied to the channel with impulse response $h(t)$, corrupted by additive noise white Gaussian (AWGN) noise $w(t)$, passed through a front-end filter $p(t)$, and finally sampled at $t = kT + \Delta$ with Δ representing a phase error that varies slowly compared to the symbol rate $1/T$. Letting $f(t) = h(t)*p(t)$ and $n(t) = w(t)*p(t)$, the sampled signal at the receiver is given by

$$\begin{aligned} c_k &= \left[\sum_i x_i f(t - iT) + n(t) \right]_{t=kT+\Delta} \\ &= \sum_i x_i f((k-i)T + \Delta) + n_k \\ &\approx \sum_i x_i [f((k-i)T) + \Delta \cdot f'((k-i)T)] + n_k \\ &= \sum_{j=0}^L f_j x_{k-j} + \Delta \cdot \sum_{j=-P_1}^{P_2} g_j x_{k-j} + n_k \\ &= y_k + \Delta \cdot \varepsilon_k + n_k \end{aligned} \quad (1)$$

where $g(t) = f'(t) = h'(t)*p(t)$ and the approximation is due to considering only the first-order effect of the phase error.¹ The first summation (denoted by y_k) represents the signal component that includes ISI whereas the second term (denoted by $\Delta \cdot \varepsilon_k$) arises due to the sampling phase error. Note $\{g_j\} = \{g(t=jT)\}$ contains noncausal components (i.e., $P_1 > 0$) and its causal portion is also longer than $\{f_j\} = \{f(t=jT)\}$ (i.e., $P_2 > L$) in general.

Fig. 1(b) shows the corresponding block diagram for the discrete-time channel model. Here x_k is assumed to be binary and

¹Although the algorithm development and analysis are based on the first-order model, the actual simulation results presented later do not make such approximation. Thus, the validation of the algorithm through simulation points to the reasonable accuracy of the first-order model for our purposes.

real-valued for ease of presentation, although the algorithm developed in this paper is applicable to multilevel or complex-valued inputs. In the sequel, for notational simplicity and ease of presentation, we shall also assume all signals are real-valued, although this assumption is not necessary for the development of the proposed algorithm. Although $p(t)$ represents a bandlimiting front-end filter in a very general sense and may affect the noise correlation properties, we shall simply assume here that it satisfies a certain pulse shaping condition so that $n_k = w(t = kT + \Delta)$ are AWGN samples. In this paper, we shall also assume that $\{f_j\}$ and $\{g_j\}$ are completely known at the receiver side (via either estimation or prior characterization).

III. PROPOSED METHOD

A. Background

Sequence detection, whether based on trellis search or tree search, typically consists of the computation of a branch metric given by $\lambda_k = (c_k - y_k)^2$ and accumulation of it to form path metrics. The detection problem amounts to finding a path of data bits with the smallest path metric. Assuming Gaussian noise and ignoring innocuous constants, the problem can be stated as one of maximizing the conditional probability $f(\mathbf{c}|\mathbf{x})$ or the log-likelihood function of the form:

$$l(\mathbf{c}|\mathbf{x}) = - \sum_k (c_k - y_k)^2. \quad (2)$$

In the presence of phase error Δ , a detection method based on the straightforward maximum-likelihood criterion will attempt to find the sequence \mathbf{x} (the bold letter is used to denote the entire sequence) that maximizes the log-likelihood function

$$l(\mathbf{c}, \Delta | \mathbf{x}) \simeq - \sum_k [c_k - y_k(\mathbf{x}) - \Delta \cdot \varepsilon_k(\mathbf{x})]^2 \quad (3)$$

$$\begin{aligned} &= - \sum_k [c_k - y_k(\mathbf{x})]^2 - \sum_k \Delta^2 \varepsilon_k^2(\mathbf{x}) \\ &\quad + \sum_k 2\Delta \cdot \varepsilon_k(\mathbf{x}) [c_k - y_k(\mathbf{x})]. \end{aligned} \quad (4)$$

The difficulty is that this log-likelihood function is not known since Δ is not known. This problem can be circumvented using an expectation-maximization type of approach, as has been done for the full response system in the presence of timing-error-induced ISI in [19]. Instead of maximizing $l(\mathbf{c}, \Delta | \mathbf{x})$, the expectation of $l(\mathbf{c}, \Delta | \mathbf{x})$ given the known observation \mathbf{c} and some preliminary estimate of \mathbf{x} can be maximized. The conditional expectation or the estimation of $l(\mathbf{c}, \Delta | \mathbf{x})$ itself is then improved using the improved version of \mathbf{x} that is obtained as a result of this maximization step. The process is repeated until there is a reason to believe that the detected sequence \mathbf{x} is reliable. Let \mathbf{x}^i denote the detected data sequence at i th iteration. The procedure can be summarized as a two-step iterative process that repeats itself until convergence is achieved.

- 1) E-step: Compute the conditional expectation

$$E\{l(\mathbf{c}, \Delta | \mathbf{x}) | \mathbf{c}, \mathbf{x}^i\}.$$

- 2) M-step: Find a new estimate

$$\mathbf{x}^{i+1} = \arg \max_{\mathbf{x}} E\{l(\mathbf{c}, \Delta | \mathbf{x}) | \mathbf{c}, \mathbf{x}^i\}.$$

Examining (4), we see that the E-step above can be replaced by the step of estimating Δ and Δ^2 using \mathbf{c} and \mathbf{x}^i . Once the estimates are found, then a Viterbi-like algorithm can be used to obtain \mathbf{x}^{i+1} based on the branch metric of the form

$$\lambda_k = [c_k - y_k(\mathbf{x})]^2 + \widehat{\Delta}^2 \varepsilon_k^2(\mathbf{x}) - 2\widehat{\Delta} \varepsilon_k(\mathbf{x}) [c_k - y_k(\mathbf{x})] \quad (5)$$

where $\widehat{\Delta}$ and $\widehat{\Delta}^2$ are the estimates of Δ and Δ^2 based on \mathbf{c} and \mathbf{x}^i . The improved sequence \mathbf{x}^{i+1} obtained as a result will in turn be used to improve $\widehat{\Delta}$ and $\widehat{\Delta}^2$. The major drawback of this approach is that multiple passes have to be taken repeatedly over the entire observation sequence \mathbf{c} each time the estimates $\widehat{\Delta}$ and $\widehat{\Delta}^2$ are adjusted. The need to estimate $\widehat{\Delta}^2$ also represents extra computation.

Another approach based on a significantly different concept is to take a multi-parameter estimation view [20]. According to this view, the likelihood function is jointly maximized with respect to \mathbf{x} and Δ , which is the same as maximizing the log-likelihood function $l(\mathbf{c} | \Delta, \mathbf{x})$. Note that in the EM point of view, Δ is a “nuisance” parameter that must be dealt with in the quest for a reliable estimate of \mathbf{x} , whereas in the multi-parameter estimation view, both Δ and \mathbf{x} are target parameters against which maximization is pursued. One way of solving the multi-parameter estimation problem is to construct a joint ISI and timing error trellis based on quantization of Δ and assuming a closed-loop tracking of Δ [18]. The joint ML problem then can be solved by running a Viterbi algorithm on this expanded trellis. In this approach, however, the size of the trellis increases by a factor roughly equal to the number of the quantization levels for Δ . A suboptimal but simpler way is to use the already detected symbol sequence $\widehat{\mathbf{x}}$ in a decision-directed mode in simply maximizing the function $l(\mathbf{c} | \Delta, \widehat{\mathbf{x}})$ against the single parameter Δ , as was done in [17]. Also, in [20], this joint ML problem was simplified by assuming a latency-free symbol-by-symbol decision followed by appropriate equalization.

B. Proposed Algorithm: Suboptimal Joint ML

Our approach here is also based on the multi-parameter estimation view but is different from the approaches of [17], [20] or [18] both conceptually and in terms of practical implication. To understand the proposed algorithm, we first observe

$$\max_{\Delta, \mathbf{x}} l(\mathbf{c} | \Delta, \mathbf{x}) = \max_{\widehat{\mathbf{x}}} \max_{\Delta} l(\mathbf{c} | \Delta, \widehat{\mathbf{x}}). \quad (6)$$

While this nested maximization problem on the right hand side is not any easier than the original joint ML problem, the nested formulation points to a very efficient suboptimal algorithm. Namely, we perform the inner maximization $\max_{\Delta} l(\mathbf{c} | \Delta, \widehat{\mathbf{x}})$ on every one-step extension of each survivor path $\widehat{\mathbf{x}}$ in the Viterbi trellis in a given symbol cycle, using only the up-to-date data. In other words, we find the value of Δ that would maximize $-\sum_{i=1}^k [c_i - y_i(\widehat{\mathbf{x}}_k) - \Delta \cdot \varepsilon_i(\widehat{\mathbf{x}}_k)]^2$ for every competing path $\widehat{\mathbf{x}}_k$ at symbol cycle k . The value of Δ that maximizes this accumulated metric will be in general different from one competing path to next. In our scheme, the outer maximization then corresponds to selecting a survivor path out of the two competing paths arriving at each node by maximizing with respect to j the function $-\sum_{i=1}^k [c_i - y_i(\widehat{\mathbf{x}}_k(j)) - \Delta(j) \cdot \varepsilon_i(\widehat{\mathbf{x}}_k(j))]^2$, where j points to a particular branch. Note that this outer maximization is done for every node in the trellis. In practice, to cope with the usually time-varying nature of Δ , only a finite window of observations are used in estimating Δ . Also, in practice the estimated Δ is fed through a phase-locked loop to maintain a closed timing loop. We note that the idea of estimating the sampling phase for a given survivor path and then extending the survivor path based on branch metric computation now incorporating the currently updated phase information for that survivor path has also been explored in [12], [13] in a similar manner. The main difference lies in that here we choose to update and maintain the phase estimate trajectory associated with each survivor and then release the reliable phase estimate sample with a certain delay to the global PLL that resides outside the Viterbi computational loop. In contrast, the work of [12], [13] is based on running a separate PLL each branch and instantly adjusting the sampling phase of the observation signal as the branch metric is computed. The practical implication is that the scheme of [12], [13] does not require storing and maintaining phase estimate trajectories with the survivor paths, whereas the present scheme does not require running separate PLLs including sampling devices for each branch.

We now present the proposed joint data and phase detection algorithm in whole while providing sufficient details for implementation. While the proposed approach can be applied to any sequence detection scheme based on the computation of branch metrics, we shall specifically use the Viterbi detector to present our algorithm. Consider a window of symbols that are implied in each branch of the channel trellis: $\{x_{k-P_2}, \dots, x_k, \dots, x_{k+P_1}\}$. The parameters P_1 and P_2 are chosen such that the $P_1 + P_2 + 1$ symbols are sufficient to compute both $y_k = (x * f)_k$ and $\varepsilon_k = (x * g)_k$, given $\{f_j\}$ and $\{g_j\}$. In fact, P_1 is in general set to greater than 0 to account for the noncausal samples in $\{g_j\}$. Let v specify a particular branch, i.e., $1 \leq v \leq 2^{P_1+P_2+1}$. For a given branch v in the channel trellis, assume that there exist some initial phase

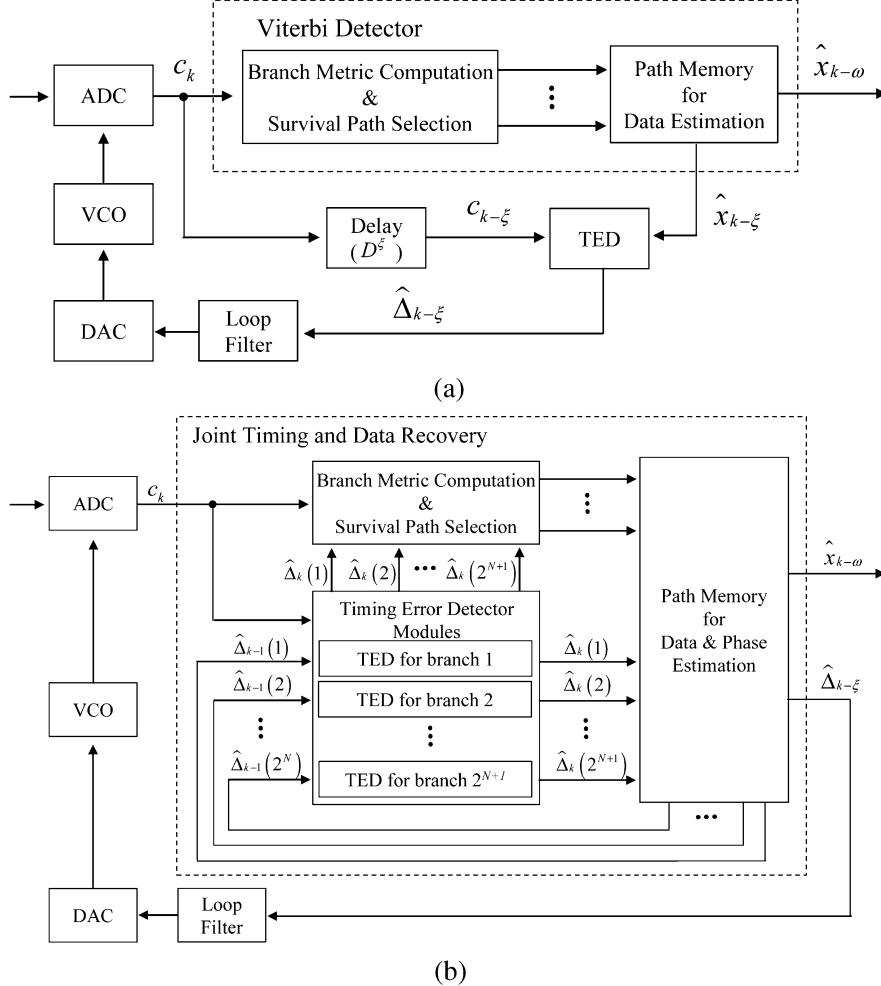


Fig. 2. Block diagrams of timing recovery schemes. (a) Conventional timing recovery. (b) Proposed timing recovery. $N = P_1 + P_2$.

estimate, $\hat{\Delta}(v)$. For a particular branch v , $y_k(v)$ and $\varepsilon_k(v)$ are also completely specified; to compute $\lambda_k(v)$, these known values along with the received (or the observation) sample c_k are used in accordance with (3), i.e.,

$$\lambda_k(v) = [c_k - y_k(v) - \hat{\Delta}(v)\varepsilon_k(v)]^2. \quad (7)$$

Once λ_k 's are computed for all branches in the given cycle, the survivor paths and their metrics are updated in the same way as in the conventional Viterbi detector. A significant modification relative to the conventional architecture at this point is that the phase estimate sequence is also updated and separately stored for each survivor path. In the beginning of the next cycle, the phase estimate $\hat{\Delta}(v)$ for each branch is updated, using the phase update rule that will be discussed in the next subsection.

The whole procedure is repeated in every cycle. In the meantime, the phase estimate sample associated with the merged survivor path or the best up-to-date survivor path is released (one sample per symbol cycle, assuming a synchronous fixed-latency architecture) with a certain delay to the loop filter, which in turn feeds the VCO. The VCO will yield a new clock phase according to which the observation sample c_k (or pre-equalized version of it, in case an additional discrete-time equalizer exists within the timing loop) is generated.

The block diagrams of the conventional timing recovery scheme and the proposed timing recovery scheme are given in Fig. 2. In contrast to the conventional scheme, the proposed method incorporates the parallel phase estimators into the detection block. While the conventional timing loop can generate a reliable TED output only after some latency ξ , the proposed scheme effectively has an additional tracking loop associated with each branch that allows phase tracking without latency. In the proposed scheme, the phase estimate is updated recursively (as will be shown in the next section) for a given branch. The estimate $\hat{\Delta}_{k-\xi}$ is generated either from the common tail of the survivor paths (when $\xi \geq \omega$, where ω denotes the data decision latency) or the best survivor path to date (when $\xi < \omega$). In practice, the TED latency ξ is generally set to smaller than the data decision latency ω to guarantee the stability of the timing recovery loop and ensure the maximum overall performance. A pseudocode for the proposed algorithm is given at the end of the next subsection.

C. Phase Update and Pseudo-Code

The phase estimate is updated for each branch in the following way. Assume that a window of M observation samples over time $k - M + 1$ to time k is to be utilized for the estimation purpose. Given a one-step extension of a survivor input path, the

value of Δ that maximizes $-\sum_{i=k-M+1}^k [c_i - y_i - \Delta \cdot \varepsilon_i]^2$ is easily obtained either by the least square fit or, equivalently, by taking derivative with respect to Δ and setting the result to zero. We obtain

$$\hat{\Delta}_k = \frac{\sum_{i=k-M+1}^k e_i \varepsilon_i}{\sum_{i=k-M+1}^k \varepsilon_i^2} \quad (8)$$

where $e_i = c_i - y_i$ and the subscript k in $\hat{\Delta}_k$ emphasizes that this estimator is for the k^{th} symbol interval. The error variance of this estimator is $\sigma_{\hat{\Delta}}^2 = E[(\Delta - \hat{\Delta}_k)^2] = \sigma_n^2 / \sum_{i=k-M+1}^k E[\varepsilon_i^2]$, where σ_n^2 is the variance of the additive white Gaussian noise (AWGN) n_k . The error variance tends to zero as σ_n^2 decreases (i.e., as SNR increases) and/or M increases.

We also mention that the estimator of (8) coincides with the data-dependent TED output of [17] when the noise becomes independent of the data pattern. We note, however, that the TED of [17] is applied to a single global data path making up of already released decisions. In contrast, we apply (8) to each tentative data path (i.e., survivor path) within the Viterbi detector separately.

For our purposes, a recursive formulation of (8) provides a more convenient phase update equation. Letting

$$\gamma_k = 1 / \sum_{i=k-M+1}^k \varepsilon_i^2 \quad (9)$$

we write (8) as

$$\begin{aligned} \hat{\Delta}_k &= \gamma_k \sum_{i=k-M+1}^k e_i \varepsilon_i \\ &= \gamma_k \left[\sum_{i=k-M}^{k-1} e_i \varepsilon_i + e_k \varepsilon_k - e_{k-M} \varepsilon_{k-M} \right] \\ &= \gamma_k \left[(\hat{\Delta}_{k-1} / \gamma_{k-1}) + e_k \varepsilon_k - e_{k-M} \varepsilon_{k-M} \right] \\ &= (\gamma_k / \gamma_{k-1}) \hat{\Delta}_{k-1} + \gamma_k (e_k \varepsilon_k - e_{k-M} \varepsilon_{k-M}) \end{aligned} \quad (10)$$

where γ_k itself can be obtained recursively

$$\gamma_k = [\gamma_{k-1}^{-1} + \varepsilon_k^2 - \varepsilon_{k-M}^2]^{-1}. \quad (11)$$

When M is sufficiently large, a good approximation for the recursive phase update is

$$\hat{\Delta}_k \approx \hat{\Delta}_{k-1} + \gamma_k (e_k \varepsilon_k - e_{k-M} \varepsilon_{k-M}). \quad (12)$$

A suboptimal phase-update equation is possible by setting

$$\begin{aligned} \hat{\Delta}_k &= \gamma \sum_{i=k-M+1}^k e_i \varepsilon_i \\ &= \hat{\Delta}_{k-1} + \gamma (e_k \varepsilon_k - e_{k-M} \varepsilon_{k-M}) \end{aligned} \quad (13)$$

in which case a reasonable value for the constant γ would be

$$\begin{aligned} \gamma &= 1 / \sum_{i=k-M+1}^k E[\varepsilon_i^2] = 1 / (M \cdot E[\varepsilon_i^2]) \\ &= 1 / (M \sigma_x^2 \eta) \end{aligned} \quad (14)$$

where $\eta = \sum_i g_i^2$.

The phase update method of (13) is clearly simpler to implement than (8) and also allows theoretical steady-state loop analysis. Expression (12) or (13) also points to the formulation of a “leaky” version of the phase update equation [21]. For the steady state loop analysis presented in the next section, we will make use of (13), while we will employ either (12) or (13) in simulation.

A pseudocode for the proposed algorithm is provided for clarity. Focus on the k^{th} stage of the trellis diagram. Let us consider two competing paths v_1 and v_2 leaving two different states a_1 and a_2 , respectively, on the left side and arriving at a given state b on the right side. The branch metric computation, path arbitration and path metric updates for this basic structure are simply repeated for every one of $2^{P_1+P_2}$ ending states in this one stage of the trellis. We assume (13) is used for phase update.

For $b = 1 : 2^{P_1+P_2}$

For $i = 1 : 2$

— Phase estimate update: $\hat{\Delta}_k(v_i) = \hat{\Delta}_{k-1}(a_i) + \gamma [e_k(v_i) \varepsilon_k(v_i) - e_{k-M}(a_i) \varepsilon_{k-M}(a_i)]$
— Branch metric computation: $\lambda_k(v_i) = [c_k - y_k(v_i) - \hat{\Delta}_k(v_i) \varepsilon_k(v_i)]^2$

End

Add, compare and select:

$$m = \arg \min_{1 \leq i \leq 2} P_{k-1}(a_i) + \lambda_k(v_i)$$

Path metric update: $P_k(b) = P_{k-1}(a_m) + \lambda_k(v_m)$

Reset the most recent phase estimate: $\hat{\Delta}_k(b) = \hat{\Delta}_k(v_m)$

Survivor path update:

$$\hat{x}_k(b) = [\hat{x}_{k-\omega}(a_m), \dots, \hat{x}_{k-1}(a_m); \hat{x}_k(v_m)]$$

Phase trajectory update:

$$\hat{\Delta}_k(b) = [\hat{\Delta}_{k-\xi}(a_m), \dots, \hat{\Delta}_{k-1}(a_m); \hat{\Delta}_k(b)]$$

End

Select the best up-to-date survivor path:

$$l = \arg \min_{1 \leq b \leq 2^{P_1+P_2}} P_k(b)$$

Symbol decision: release the oldest symbol in $\hat{x}_k(l)$

Phase estimation: release the oldest phase estimate in $\hat{\Delta}_k(l)$

Reset all other $\hat{x}_k(b)$'s and $\hat{\Delta}_k(b)$'s by dropping the oldest elements.

Repeat the above steps with $k \leftarrow k + 1$.

Note that the term $e_{k-M}(a_i) \varepsilon_{k-M}(a_i)$ in the phase estimate update equation is specific to the state a_i and can be obtained

from the decisions stored in its survivor path. For a leaky phase update, the need to compute or maintain this term disappears. In a given cycle, the order of branch metric computation and phase revision can be reversed without affecting results.

Had (12) been used for phase update instead, the constant γ in the phase estimate update equation would have been replaced by a branch-dependent and time-dependent parameter $\gamma_k(v_i)$. An extra update equation would also have been added

$$\gamma_k(v_i) = [\gamma_{k-1}^{-1}(a_i) + \varepsilon_k^2(v_i) - \varepsilon_{k-M}^2(a_i)]^{-1} \quad (15)$$

along with a reset equation

$$\gamma_k(b) = \gamma_k(v_m) \quad (16)$$

after the add/compare/select operation.

Notice that in the pseudocode description of the algorithm, the extra computation and memory burden relative to the conventional Viterbi detector is due to the phase update step and the need to maintain the phase estimate trajectories. In practice, however, the phase latency ξ can be made very small, meaning that the extra storage requirement is typically negligible. We note that overall, the main complexity increase relative to the conventional Viterbi detector arises from the need to specify more symbols for each branch to predetermine the term $\varepsilon_k(v)$, which results in an increased Viterbi trellis by about a factor of four. The factor four comes from our empirical observation that the “derivative” response $\{g_i\}$, counting only the significant samples, is longer than the ISI response $\{f_i\}$ by 2 samples in the channel responses investigated in this work.

It is possible to use tentative decisions implied in a given survivor path to reduce the size of trellis, i.e., each branch of the channel trellis can be associated with $\{\tilde{x}_{k-P_2}, \dots, \tilde{x}_{k-Q-1}, x_{k-Q}, \dots, x_k, \dots, x_{k+P_1}\}$, where $\{\tilde{x}_{k-P_2}, \dots, \tilde{x}_{k-Q-1}\}$ denote the symbols implied in the survivor path. At this point, how much performance loss is incurred as the trellis size is reduced using this approach is subject to further investigation.

IV. PHASE LOCKED LOOP ANALYSIS

In this section we perform loop analysis to compare the performance potential of the proposed algorithm relative to the timing loops based on the traditional MMSE principle [7], [10] as well as the popular MM approach [6]. We first focus on the TED performance and then analyze the overall loop behavior.

A. Comparison of TEDs

1) *TED Gain to Noise Ratio:* The effectiveness of a TED is typically evaluated based on a linearized, small-input model for the TED input/output behavior. Assuming a small timing error, the linearized TED output model assumes that the phase estimate can be approximated as

$$\begin{aligned} \hat{\Delta}_k &= A \cdot \Delta + p_k \\ &= A(\phi - \tau_k) + p_k \end{aligned} \quad (17)$$

where A represents the TED gain, Δ the residual phase error, ϕ the incoming phase error, τ_k the VCO output, and p_k any error in the linear approximation. Here, the residual phase error Δ

is defined as the difference between the incoming phase error ϕ and the VCO output $\tau(\Delta \triangleq \phi - \tau)$. At steady-state, p_k in (17) coincides with $\hat{\Delta}_k$ as $\tau_k = \phi$. Further assuming that the symbol decisions used in constructing $\hat{\Delta}_k$ are all correct, p_k can be directly related to the additive input noise n_k . In this sense, p_k is considered as the effective TED output noise.

An existing figure-of-merit for the efficiency of the TED is based on the ratio of A^2 to the dc component of the power spectral density (PSD) of p_k [11]. Unfortunately, however, this ratio fails to bring out the performance advantage of the proposed TED. The reason for this is that unlike conventional TED schemes, the proposed TED is incorporated into the Viterbi metric computation and path arbitration processes, resulting in a considerably improved linear range of the TED output. The performance potential associated with the improved linear range of the TED output obviously cannot be accounted for in the existing analysis based on the ideal linearized TED model.

To make the point clear about the inadequacy of the conventional figure-of-merit in analyzing our scheme, it is instructive to evaluate the TED gain to the dc noise ratio of the proposed scheme versus the MMSE scheme.

Let us first consider the MMSE timing loop. The MMSE TED is defined [10] as

$$\hat{\Delta}_k = (c_k - \hat{y}_k)c'(kT + \Delta) \quad (18)$$

where $c'(kT + \Delta) = [dc(t)/dt]_{t=kT+\Delta}$ with $c(t) = \sum_{j=0}^L x_j f(t - jT) + w(t) * p(t)$. A “zero-forcing” approximation to this MMSE TED results if $c'(kT + \Delta)$ is replaced with a sampled signal that would arise in the absence of noise and sampling phase [11]

$$\hat{\Delta}_k = (c_k - \hat{y}_k)\hat{\varepsilon}_k = [c_k - (\hat{x} * f)_k](\hat{x} * g)_k. \quad (19)$$

The “S-curve” is the expectation of $\hat{\Delta}_k$ for a given value of Δ . Calculations are performed assuming \hat{x}_k is correct, which implies that τ is close to the incoming phase error ϕ . If the input symbols are uncoded or coded with good interleaving, we can assume that the input symbols $\{x_k\}$ are zero-mean and independent. Then, taking the expectation of (19), utilizing (1) and using the above assumptions produce the S-curve as a function of Δ

$$\begin{aligned} S(\Delta) &= E[\hat{\Delta}_k] = E[(c_k - \hat{y}_k)\hat{\varepsilon}_k] \\ &= E[\{(y_k + \Delta \cdot \varepsilon_k + n_k) - \hat{y}_k\}\hat{\varepsilon}_k] \\ &\approx \Delta \cdot E[\varepsilon_k \cdot \varepsilon_k] + E[n_k] \cdot E[\varepsilon_k] \\ &= \Delta \cdot E \left[\sum_{i=-\infty}^{\infty} x_{k-i} g_i \cdot \sum_{j=-\infty}^{\infty} x_{k-j} g_j \right] \\ &= \Delta \cdot \sum_{i=-\infty}^{\infty} E[x_{k-i}^2] g_i^2 = \Delta \cdot \sigma_x^2 \cdot \sum_{i=-\infty}^{\infty} g_i^2 \\ &= \Delta \cdot \sigma_x^2 \eta \end{aligned} \quad (20)$$

where σ_x^2 is the input symbol variance. This S-curve crosses zero with a positive slope at the origin, and this slope corresponds to the TED gain A in (17). At steady state (i.e., $\Delta \approx 0$), (17) reduces to $\hat{\Delta}_k = p_k$. Further, assuming decisions are correct

(i.e., $\hat{y}_k = y_k$, $\hat{\varepsilon}_k = \varepsilon_k$), (19) can be written as $\hat{\Delta}_k = (c_k - y_k)\varepsilon_k = n_k\varepsilon_k$. Thus, we have

$$p_k = n_k\varepsilon_k. \quad (21)$$

For an uncorrelated sequence $\{x_k\}$, the autocorrelation function of p_k is easily shown to be

$$R_p(l) = \sigma_n^2 \sigma_x^2 \eta \delta(l). \quad (22)$$

and the PSD is

$$S_p(\Omega) = \sigma_n^2 \sigma_x^2 \eta \quad (23)$$

where $\Omega = fT$. For MMSE TED, $A = \sigma_x^2 \eta$ as seen in (20), and the TED gain to dc noise ratio (TGMR) we are looking for is

$$\text{TGMR}_{\text{MMSE}} = A^2 / S_p(0) = \sigma_x^2 \eta / \sigma_n^2. \quad (24)$$

For the proposed algorithm, focussing on the TED given by (8), we obtain

$$\begin{aligned} S(\Delta) &= E[\hat{\Delta}_k] \\ &= E\left[\frac{\sum_{i=k-M+1}^k \{(y_i + \Delta \cdot \varepsilon_i + n_i) - \hat{y}_i\} \varepsilon_i}{\sum_{i=k-M+1}^k \varepsilon_i^2}\right] \\ &= \Delta \cdot E\left[\frac{\sum_{i=k-M+1}^k \varepsilon_i \cdot \varepsilon_i}{\sum_{i=k-M+1}^k \varepsilon_i^2}\right] = \Delta \end{aligned} \quad (25)$$

indicating $A = 1$. Now to get $S_p(0)$ for the proposed TED, write

$$p_k = \gamma \sum_{i=k-M+1}^k n_i \varepsilon_i \quad (26)$$

assuming that the denominator of (8) is replaced by $1/\gamma$, with γ given by (14). By using the assumption that the symbols are uncorrelated, we get

$$\begin{cases} R_p(0) = \sigma_p^2 = \gamma^2 \sigma_n^2 \sigma_x^2 M \eta \\ R_p(1) = R_p(-1) = \gamma^2 \sigma_n^2 \sigma_x^2 (M-1) \eta \\ \vdots \\ R_p(M-1) = R_p(-M+1) = \gamma^2 \sigma_n^2 \sigma_x^2 \eta \end{cases} \quad (27)$$

the Fourier transformation of which leads to

$$S_p(0) = \gamma M \sigma_n^2 \quad (28)$$

and thus

$$\text{TGMR}_{\text{prop}} = 1 / (\gamma M \sigma_n^2) = \sigma_x^2 \eta / \sigma_n^2 \quad (29)$$

which is the same as the MMSE scheme. Again, it is clear that the TGMR-based analysis fails to show the positive impact of the path-dependent, windowed estimation of the phase on the timing loop performance.

For the sake of completeness, we also compute the TGMR for the popular MM TED. The application of the TED based on the MM principle to a partial response channel yields [5]

$$\hat{\Delta}_k = c_k \hat{y}_{k-1} - c_{k-1} \hat{y}_k. \quad (30)$$

The corresponding S-curve is

$$\begin{aligned} S(\Delta) &= \sigma_x^2 \sum_{k=-\infty}^{\infty} f(kT - \Delta) f\{(k-1)T\} \\ &\quad - \sigma_x^2 \sum_{k=-\infty}^{\infty} f\{(k-1)T - \Delta\} f(kT) \\ &\approx \Delta \cdot \sigma_x^2 \left[\sum_{k=-\infty}^{\infty} (f_k g_{k-1} - f_{k-1} g_k) \right] \end{aligned} \quad (31)$$

where the approximation is reasonable for a small Δ . The TED noise in steady state is

$$p_k = n_k y_{k-1} - n_{k-1} y_k \quad (32)$$

and its PSD can be shown to be

$$S_p(\Omega) = 2\eta_f \sigma_n^2 \sigma_x^2 [1 - \rho_f(2) \cos(2\pi\Omega)] \quad (33)$$

where $\eta_f = \sum_i f_i^2$ and $\rho_f(2) = \eta_f^{-1} \sum_i f_i f_{i-2}$. Thus, we have

$$\text{TGMR}_{\text{MM}} = \sigma_x^2 \eta_{\text{MM}} / \sigma_n^2 \quad (34)$$

where

$$\eta_{\text{MM}} = \frac{[\sum_i (f_i g_{i-1} - f_{i-1} g_i)]^2}{2\eta_f [1 - \rho_f(2)]}. \quad (35)$$

TGMRs of the MMSE and MM are fairly similar. For the minimum-bandwidth, extended partial response class-4 (EPR4), with the sampled channel impulse response given by $\{1 \ 1 \ -1 \ -1\}$, we get $\text{TGMR}_{\text{MM}} = 7.00 \sigma_x^2 / \sigma_n^2$ versus $\text{TGMR}_{\text{MMSE}} = \text{TGMR}_{\text{prop}} = 7.60 \sigma_x^2 / \sigma_n^2$. For the same PR channel but with an excess bandwidth of 25%, we obtain $\text{TGMR}_{\text{MM}} = 6.99 \sigma_x^2 / \sigma_n^2$ while $\text{TGMR}_{\text{MMSE}} = \text{TGMR}_{\text{prop}} = 7.55 \sigma_x^2 / \sigma_n^2$.

To understand the improved TED quality of the proposed joint detection/timing scheme, we resort to open-loop simulation, as discussed next.

2) *Open-Loop TED Performance:* Fig. 3 shows the TED output, averaged over many data and noise sample sequences, and its standard deviation as a function of input phase error, at SNR = 6 dB for the EPR4 channel. The TED output and the low pass loop filter input are disconnected in Fig. 4 to simulate the open-loop performance. The input bit stream x_k is oversampled by four times and the oversampled channel output is obtained by passing the oversampled bit stream into the four-times oversampled EPR4 channel model. Finally, 30% oversampled waveform $r_{k'}$ is obtained by using a 4-tap cubic interpolator and by adding AWGN $n_{k'}$. The normalized phase error $\phi_{k'}$ is induced during the interpolation process according to the phase error generator output with $\tau_k = 0$. The normalized phase error output has changed from -0.5 to $+0.5$ by a step size of 0.02. For each phase error value, the TED output is averaged over many data and noise sequences to obtain the S-curve, and its standard deviation is also calculated. In each scheme, the phase error estimate in the TED is obtained using actual decisions. For the MMSE and MM schemes, the bit decisions are taken out

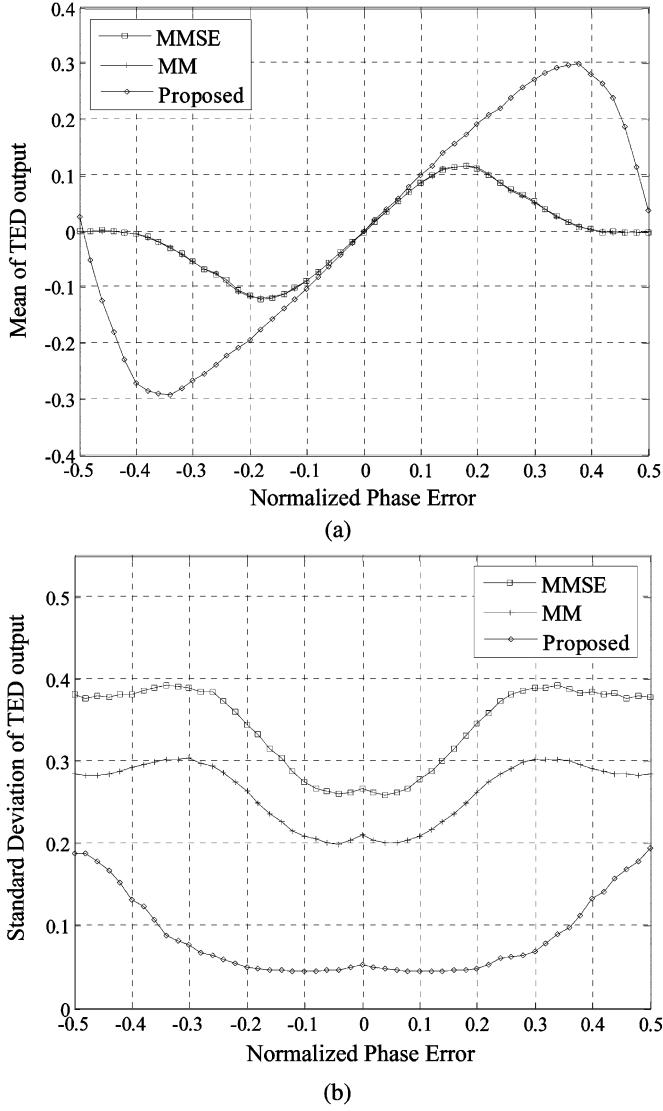


Fig. 3. Open-loop simulation results for the MM, MMSE and proposed timing error detectors (MMSE and MM curves fall on top of each other in (a) S-curve. (b) Standard deviation).

of the best Viterbi survivor to date with a delay of 11 symbol cycles, whereas the proposed TED output corresponds to the stored phase estimate associated with the best Viterbi survivor to date, taken with a delay of one cycle. The data window size M is set to 30 for the proposed scheme. These parameters are obtained empirically while trying to minimize the BER during simulation, as discussed in the next section.

Simulation results agree well with the computed S curves for small timing errors. As the input timing error increases, however, the small Δ assumption as well as the assumption $\hat{x}_k = x_k$ become less accurate, and the performance suffers in general. Nevertheless, it is clear that the proposed TED does have a larger linear range. It is also clear that the response range over which the mean TED output remains positive (negative) to a positive (negative) input phase error is also significantly larger with the proposed TED. This implies a better tracking behavior for the proposed loop. A smaller standard deviation is also noted with the proposed TED, although this is less important in closed-loop

operations where the jitter effect gets largely averaged out. The extended linear range and the wider response range are due to the enhanced decision and phase estimation quality that results from integrating the phase estimation process into branch metric computation, and will result in considerably lower probability of cycle slips, as will be discussed later.

B. Steady State Jitter and Loop Stability

In this section we analyze the overall loop characteristics of the proposed timing loop and the MMSE timing loop at steady state. Our goal here is to understand the impact of the window size parameter M on the loop behavior, including jitter and stability. We will not pursue the analysis of the MM-TED-based loop from this point on, as its performance is similar to that of MMSE timing loop. Comparison of (8) and (19) indicates that the proposed TED output can be viewed as a windowed and weighted accumulation of the MMSE TED output $e_k \varepsilon_k$. This observation suggests that the windowed accumulation inherent in (8) be viewed as an extra filter that operates on the basic TED output $e_k \varepsilon_k$ versus part of the TED function as assumed in the previous subsection. This alternative view also makes the comparison of the steady-state loop behavior more meaningful between the MMSE loop and the proposed loop, since the effective noise that enters the TED is the same (white) for both loops.

1) *Steady State Jitter Analysis:* Fig. 5(a) shows the steady-state model of the conventional second-order phase-locked loop, based on the linearized TED output model. Fig. 5(b) corresponds to the proposed loop base on the TED of (13), with the “averaging filter” placed after the basic MMSE TED. We assume A is simply $\sigma_x^2 \eta$ as given in (20), in both figures.

Letting $\Delta_k \triangleq \phi - \tau_k$, the transfer function from p_k to Δ_k is easily obtained as

$$H(z) = \frac{-\{\alpha(z-1) + \beta\}(1-z^{-M})}{A\{\alpha(z-1) + \beta\}(1-z^{-M}) + \gamma(z-1)^3 z^{\xi-1}}. \quad (36)$$

where α and β are the usual phase and frequency update gains of the first-order loop filter, respectively, ξ is the latency in the timing detector output, γ is a constant defined in (14) and M is the window size. It is clear that with $M = 1$ and $\gamma = 1$, the structure reduces to that of the conventional timing loop. The equivalent noise bandwidth is defined as [22]

$$\begin{aligned} B_{LT} &\triangleq \frac{T}{2|H(z=1)|^2} \cdot \int_{-1/(2T)}^{1/(2T)} |H(z=e^{j2\pi fT})|^2 df \\ &= A^2 \int_0^{1/2} |H(z=e^{j2\pi\Omega})|^2 d\Omega \end{aligned} \quad (37)$$

where $\Omega = fT$.

As we have shown earlier, the PSD of p_k is given by

$$S_p(\Omega) = \sigma_n^2 \sigma_x^2 \eta \quad (38)$$

so the timing jitter variance σ_Δ^2 can be calculated as [8]

$$\sigma_\Delta^2 = 2 \int_0^{0.5} S_p(\Omega) |H(\Omega)|^2 d\Omega = 2\sigma_n^2 B_{LT}/A. \quad (39)$$

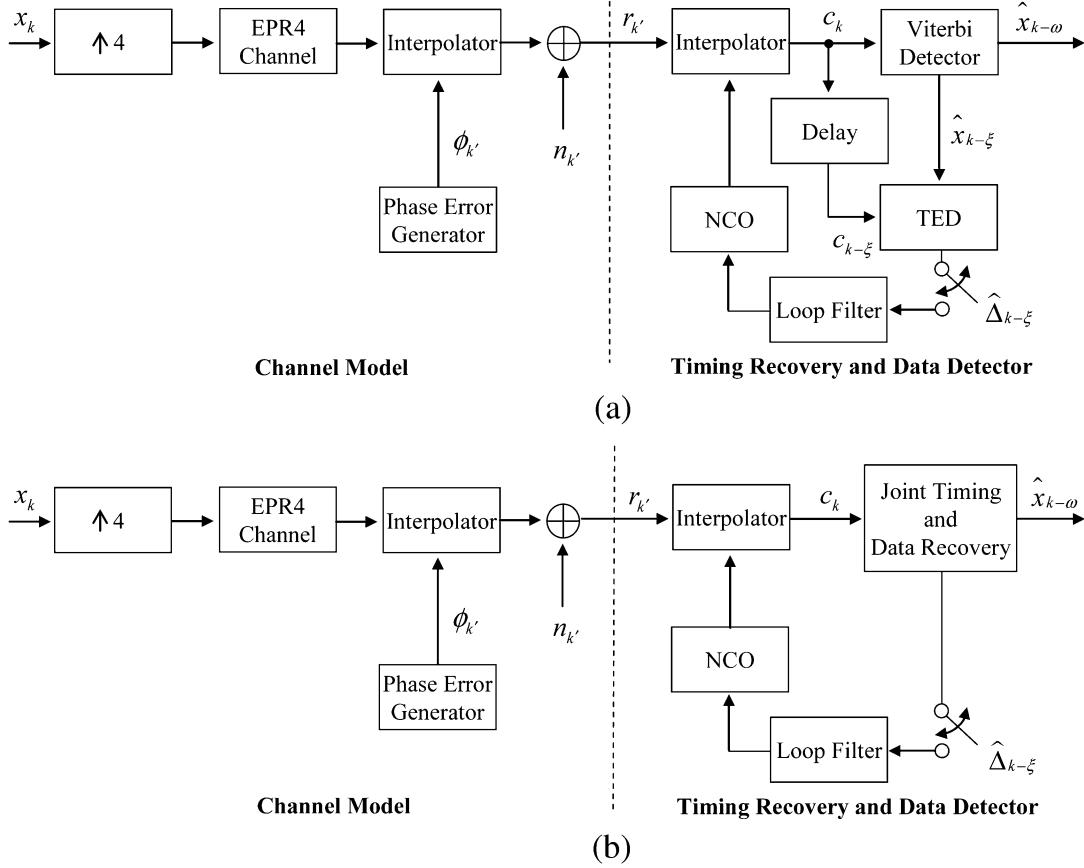


Fig. 4. Block diagrams for open-loop/closed-loop simulations. (a) Simulation model for the conventional scheme (b) Simulation model for the proposed joint scheme.

From (36), (37), and (39), it is clear that the jitter performance of the proposed scheme depends on the window size M . While the window is introduced to improve the phase estimation quality of the proposed loop, it turns out that increasing M also increases B_{LT} .

We now attempt to understand the impact of M on the steady-state jitter performance as well as stability. In the conventional timing loop, it is also possible to improve the timing phase estimation quality by driving the TED with high-latency, high-quality bit decisions. Therefore, it is insightful to start the analysis by comparing the effects of the data window size M and the loop latency ξ on the characteristics of the timing recovery loop. The loop latency ξ is caused by the delay with which the TED releases its phase estimate to the loop filter.

To proceed, the magnitude of the loop transfer function $|H(\Omega)|$ is considered. In Fig. 6(a), the window size M is set to 1 and the TED latency ξ is varied from 0 to 30 in steps of 10. As ξ increases, the peak of the frequency response increases considerably, but there is little change at high frequencies. From this observation, it is clear that the timing jitter σ_{Δ}^2 would increase as the loop latency ξ increases. In Fig. 6(b), ξ is set to zero and the data window size M is varied from 1 to 60 in steps of 20. As the window size M increases, the peak also increases much like the responses in Fig. 6(a), but there also exists an increasingly fast roll-off at high frequencies. It turns out that the jitter of the timing loop increases as M increases, but at a much slower rate than the jitter increase as a function of ξ . Other loop

parameters are set to: $A = 1$, $\sigma_x^2 = 1$, $\eta = 1$, $\alpha = 0.02$ and $\beta = 0.0002$ for Fig. 6(a) and (b).

Fig. 7 shows a contour plot of B_{LT} and σ_{Δ}^2 for various combinations of M and ξ , assuming $A = 1$, $\sigma_x^2 = 1$, $\sigma_n^2 = 0.1$, $\eta = 1$, $\alpha = 0.02$, and $\beta = 0.0002$. As M and/or ξ increase, B_{LT} and σ_{Δ}^2 increase. It is also clear that many different combinations of M and ξ can lead to the same B_{LT} and thus the same steady state timing jitter. As an example, for all the cases of $M = 1$ and $\xi = 11$, $M = 10$ and $\xi = 8$, $M = 20$ and $\xi = 4$, and $M = 30$ and $\xi = 1$, the steady state jitter remains at 0.002 (corresponding to $B_{LT} \approx 0.01$).

2) Stability Issues: An important issue that should be considered during timing recovery loop design is the loop stability. It is well known that increasing loop latency ξ results in a rapid reduction of the stable region for the loop filter coefficients α and β [23], [24]. However, the proposed loop has the additional parameter M . Here we examine the impact of M on stability region of the loop. We plotted the regions of stability in Fig. 8 by identifying the α and β values for which all the poles of $H(z)$ are still within the unit circle in the z -domain, for different combinations of ξ and M . Other loop parameters are again set to: $A = 1$, $\sigma_x^2 = 1$, and $\eta = 1$. The stable region is on the left side of a given curve, so any combinations of α and β corresponding to the outside of the stable region will result in an improper timing recovery loop operation. As can be seen, the stable region shrinks rapidly as we increase the loop latency ξ , but the region seems to be less sensitive to increasing M . Fig. 9

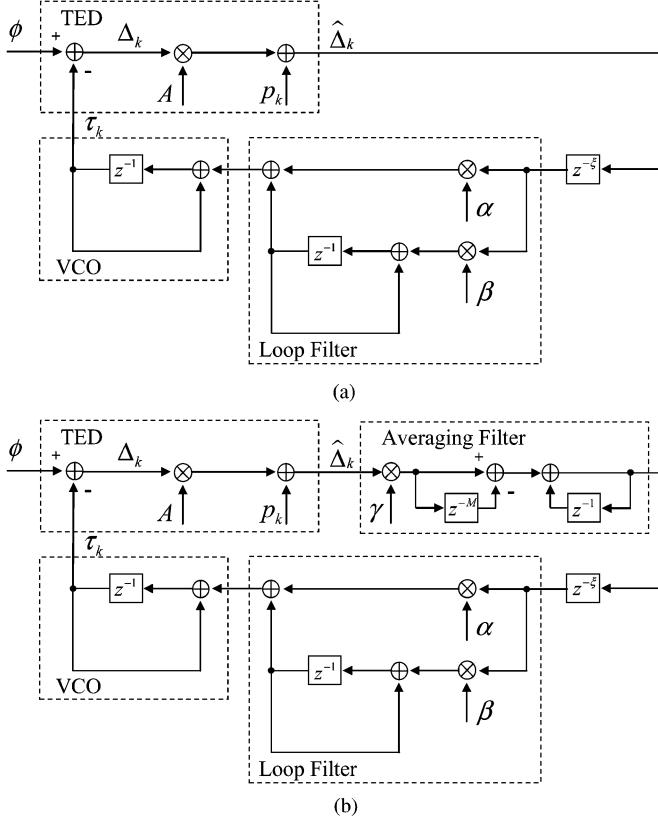


Fig. 5. Steady-state PLL models. (a) Conventional scheme. (b) Proposed scheme.

shows the stability regions of the four combinations of M and ξ that gave the same B_LT of 0.01 (and $\sigma_\Delta^2 = 0.002$), as shown in Fig. 7. It is ensured that any of these combinations provide an ample stability region for α and β .

Note that the above analysis still assumes correct symbol decisions for all timing loops. However, recall that one advantage of the proposed algorithm is in the Viterbi algorithm with phase estimation incorporated into its branch metric computation to provide better symbol estimates without adding latency in the loop. Although the phase estimate stream can be released to the loop filter and the VCO with some delays, the phase error compensation occurring in branch metric computation is effectively without latency. Also, the TED latency ξ in the proposed algorithm can be made small without sacrificing the quality of the TED output. The same cannot be said for the other timing loops. The more realistic (non-steady-state) performance advantage of the proposed scheme is investigated using simulation in the next section.

V. TRACKING SIMULATION AND BER PERFORMANCE

In this section we compare the tracking time and BER performance of the timing recovery loops by simulation. The TED output and the low pass loop filter input are connected in Fig. 4 to simulate the closed-loop performance. The numerically controlled oscillator (NCO) plays the role of a VCO. Four parameter settings of $M = 1$ and $\xi = 11$, $M = 10$ and $\xi = 8$, $M = 20$ and $\xi = 4$, and $M = 30$ and $\xi = 1$ are considered. For $M = 1$ and $\xi = 11$, the traditional Viterbi detector is used as the data detector to model the conventional MMSE timing recovery loop.

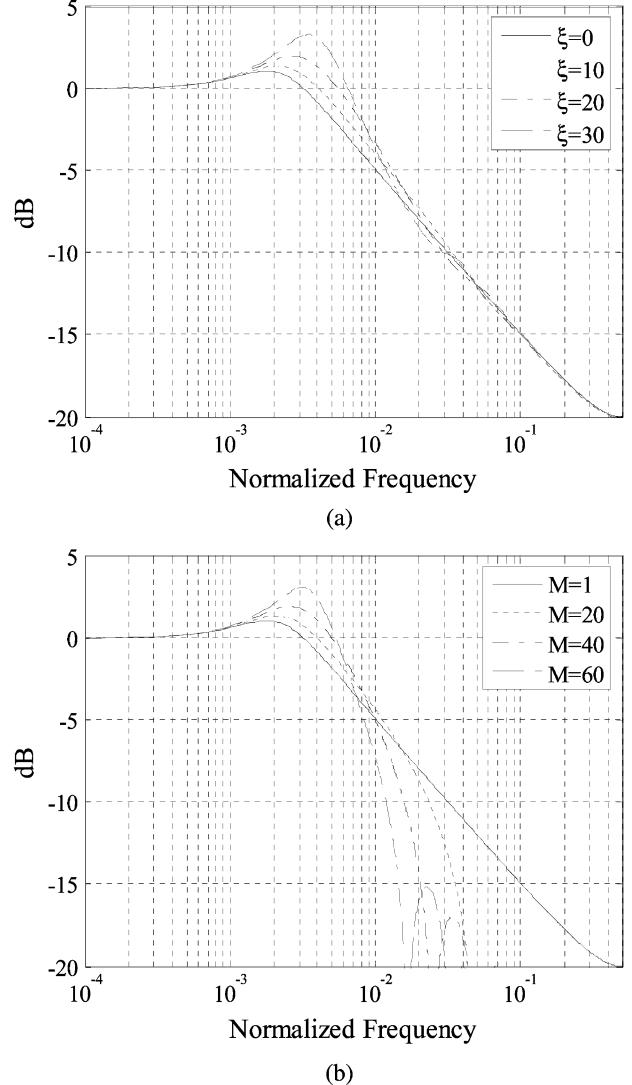


Fig. 6. Effect of ξ and M on the equivalent loop $|H(f)|$. (a) Effect of loop latency ξ on $|H(f)|$. (b) Effect of window size M on $|H(f)|$.

For other cases, proposed joint Viterbi detector which incorporates phase estimation in the data detection is used to provide better estimation results.

A. Tracking Performance Comparison

Fig. 10 shows the tracking curves of the four cases mentioned. In the simulation, the EPR4 signal is used as the channel impulse response, and loop parameters are set to: the normalized frequency offset $f_e = 1/100$, SNR = 10 dB with AWGN, $\alpha = 0.02$, and $\beta = 0.0002$. For the proposed TED, (13) is used in the tracking performance comparison for simplicity. The depth of the Viterbi data path is fixed at 80 for reliable data decision. The conventional MMSE timing recovery loop ($M = 1$ and $\xi = 11$) which has a traditional Viterbi detector converges to the steady-state after about 8 000 samples. In contrast, the proposed timing recovery loops ($M = 10$ and $\xi = 8$, $M = 20$ and $\xi = 4$, and $M = 30$ and $\xi = 1$) equipped with the “phase-tracking” Viterbi detector converge to the steady-state after about 3000 samples or less. The proposed joint timing recovery loops converge almost three times faster than the conventional timing recovery loop. This improvement in tracking

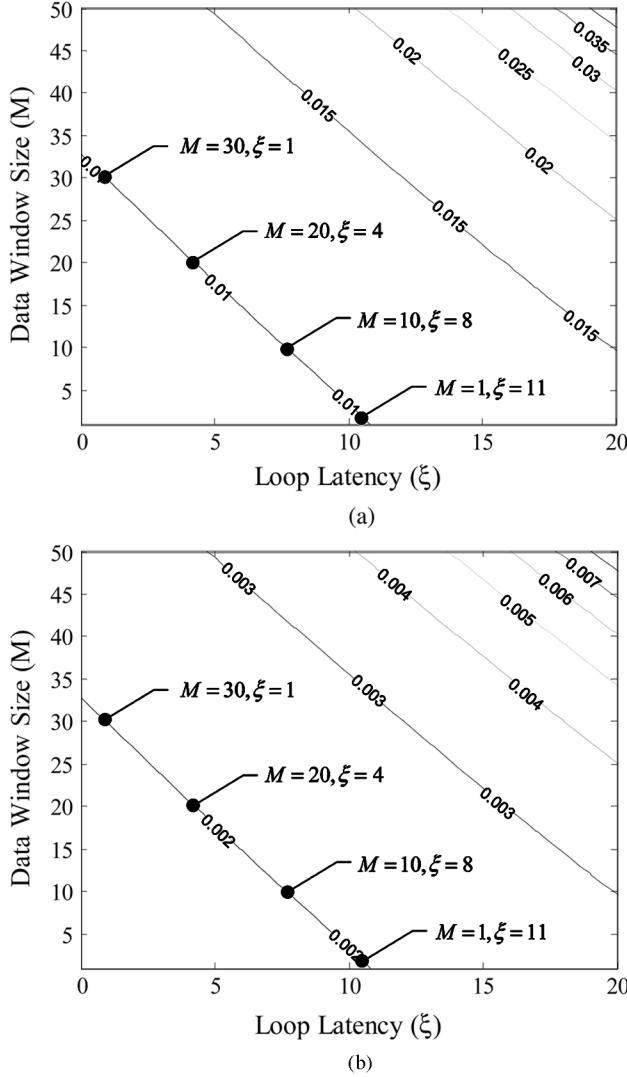


Fig. 7. Effect of loop latency ξ and window size M on B_{LT} and σ_{Δ}^2 . (a) Effect of loop latency ξ and window size M on B_{LT} . (b) Effect of loop latency ξ and window size M on σ_{Δ}^2 .

time is caused by the improved estimation quality during the frequency acquisition. While the plots are not shown here, the BERs of the four timing loops are almost the same at all SNRs once the acquisition is done. This is consistent with the jitter analysis result in the previous section. At the steady-state of an AWGN environment, the proposed joint timing recovery loop does not exhibit a BER improvement because the timing error Δ is around zero. However, if the frequency offset f_e fluctuates rapidly or the noise is not AWGN, we expect that the BER improvement of the proposed timing loop would be significant in addition to the tracking time improvement.

B. BER and Cycle Slip Probabilities

We also ran BER simulations and obtained the estimates of the cycle slip probabilities by simulation in the presence of time-varying phase errors. The normalized phase error Δ_k/T is modelled as

$$\frac{\Delta_k}{T} = f_e k + p_e \sin\left(\frac{2\pi k}{T_p}\right) \quad (40)$$

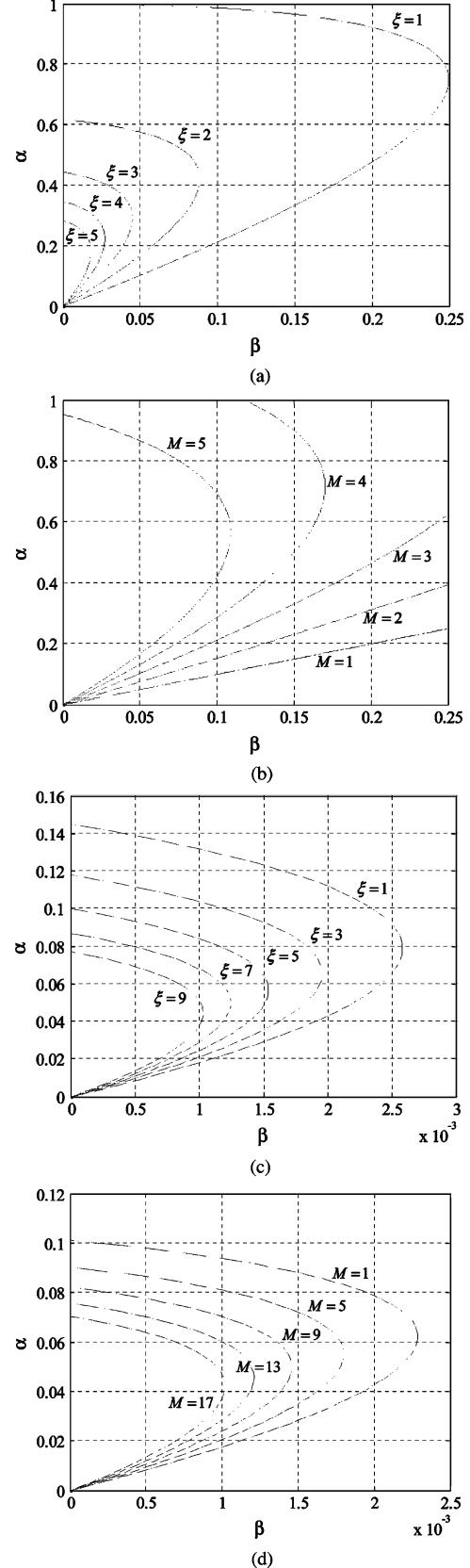


Fig. 8. Stable regions of loop filter parameters with respect to various combinations of M and ξ . (a) Stable regions with respect to various ξ when $M = 1$. (b) Stable regions with respect to various M when $\xi = 0$. (c) Stable regions with respect to various ξ when $M = 30$. (d) Stable regions with respect to various M when $\xi = 15$.

where f_e is the normalized constant frequency offset, p_e is the peak phase fluctuation, and T_p is the period of phase fluctu-

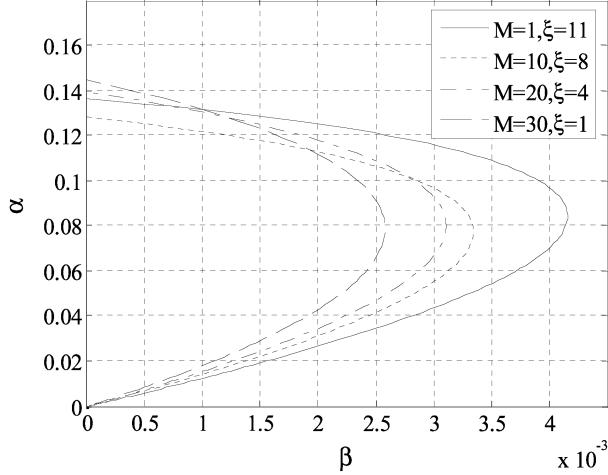


Fig. 9. Stable regions for different combinations of M and ξ which produce $B_{LT} = 0.01$ and $\sigma_{\Delta}^2 = 0.002$.

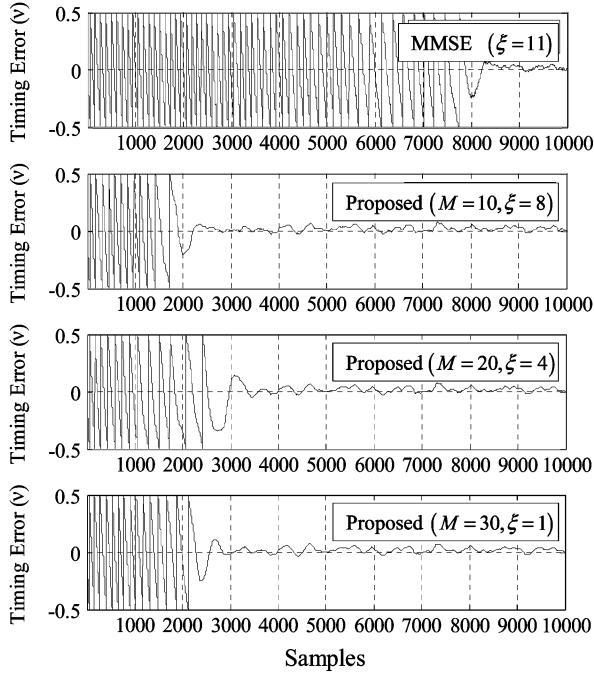
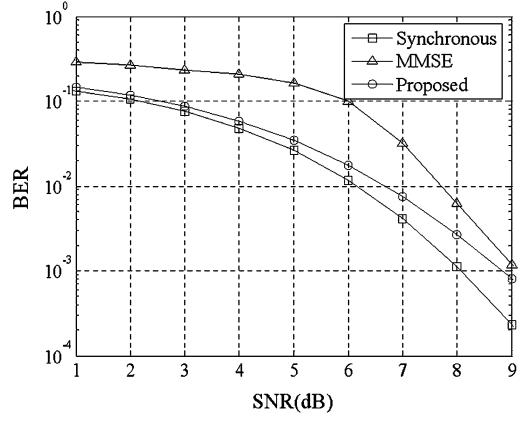


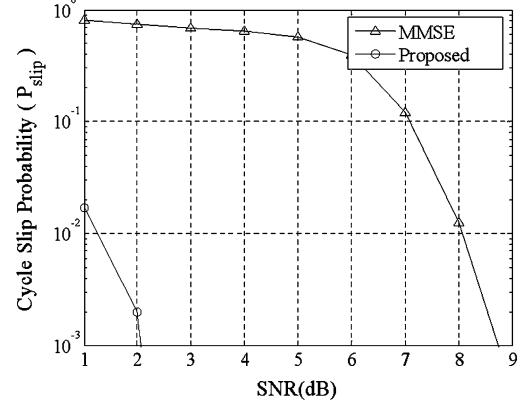
Fig. 10. Tracking speed comparison for different settings of M and ξ .

ations. Equation (40) simulates the effect of the time-varying phase error and a constant frequency offset. The BER performance and the cycle slip probability of the proposed algorithm are compared with those of the MMSE timing recovery algorithm operating in conjunction with the conventional Viterbi algorithm. The MMSE TED is driven by preliminary symbol decisions made from a conventional 8-state Viterbi detector, but the proposed timing recovery loop operates with a 32-state Viterbi which generates timing estimate and data estimate jointly. The 32-state trellis is necessary as the response g is longer than f by approximately two samples (there exist one significant non-causal sample as well as one more extra causal sample in g).

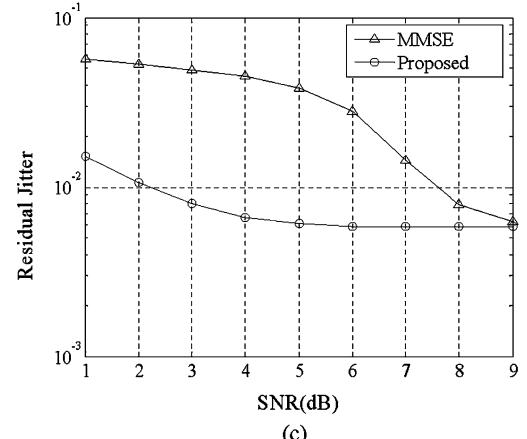
The closed-loop simulation results are shown in Fig. 11. For the proposed TED, (12) is used in this case to observe the full performance potential. The frequency offset f_e is set to 2/100, peak phase p_e is to 0.1, and period T_p is to 1 000. All the



(a)



(b)



(c)

Fig. 11. Closed-loop simulation results with different B_{LT} . (a) BER. (b) Cycle slip probability (P_{slip}). (c) Residual jitter (σ_{Δ}^2).

loop parameters are adjusted to minimize the BER at $\text{SNR} = 6$ dB. The SNR here is defined as $\sum_i f_i^2 / (2\sigma_n^2)$. For the MMSE timing recovery loop, the corresponding loop parameters are: $\xi = 11$, $\alpha = 0.01$, and $\beta = 0.0001$. For the proposed loop, they are: $M = 30$, $\xi = 1$, $\alpha = 0.005$, and $\beta = 0.00005$. It is clear that the proposed algorithm outperforms the MMSE approach especially at low SNRs. The MMSE approach breaks down at low SNRs, whereas the proposed algorithm exhibits graceful degradation.

The main cause of the BER break down is the cycle slip of the timing recovery loop. The cycle slip is the bit-shift in the reproduced binary data caused by an insertion of redundant bits or

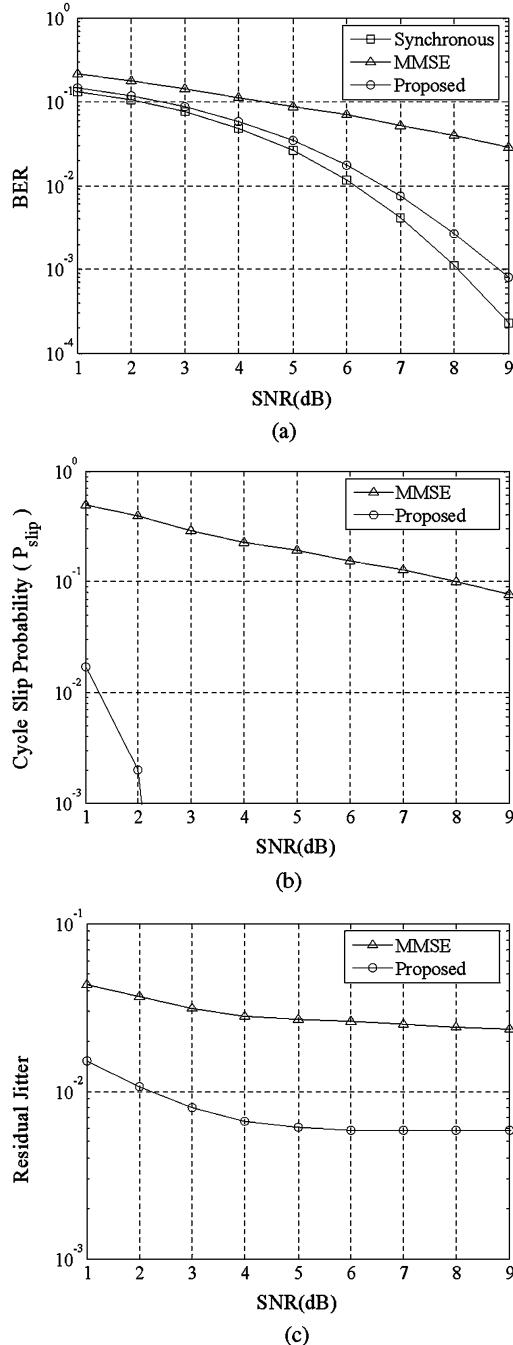


Fig. 12. Closed-loop simulation results with same $B_L T$. (a) BER. (b) Cycle slip probability (P_{slip}). (c) Residual jitter (σ_{Δ}^2).

missing of data bits in the reproduced data sequence [23]. If the cycle slip occurs at some point in the reproduced data sequence, an error burst will continue until the sequence is synchronized by using known patterns like preambles.

Cycle slip probabilities of the conventional MMSE timing recovery loop and those of the proposed joint timing recovery loop are compared in Fig. 11(b). Each frame contains 4096 uncoded bits. The cycle slip probability is measured as the number of the captured frames contaminated by cycle slips divided by the number of the frames free of cycle slips. Loop parameters are optimized to have the lowest BER at SNR = 6 dB when

the cycle-slip-induced errors were not counted. Consequently, the conventional loop runs with large loop filter coefficients to follow the fast-varying phase fluctuations, but the proposed scheme operates with smaller loop filter coefficients because the fast-varying phase fluctuations can be compensated to a significant extent within the Viterbi processing cycle. At SNRs above 9 dB, there is little performance difference between the conventional loop and the proposed loop. However, at below SNR = 8 dB the difference is significant. While the cycle slip probability of the proposed loop remains very small, that of the conventional MMSE loop goes to 1. The relatively strong performance at a low SNR is of great importance as this allows powerful iterative coding schemes to operate at low channel SNRs. Residual jitter performances of the two loops are also shown in Fig. 11(c). As can be seen, the jitter curves are consistent with the BER results.

Another BER simulation results are shown in Fig. 12 with the competing timing loops set up to have the same $B_L T$ this time. Simulation parameters and loop parameters are the same as in Fig. 11 for the proposed scheme, whereas for the MMSE timing recovery loop, loop parameters are set to: $\xi = 11$, $\alpha = 0.005$, and $\beta = 0.00005$. It is clear that while the cycle slip probability is somewhat reduced for the MMSE loop at low SNRs, the high SNR performance now suffers greatly, reflecting the inability of the MMSE loop to follow the time-varying phase fluctuations, given a small $B_L T$ (the same as the proposed loop). The jitter curves also reflect this trend.

VI. CONCLUSION

A combined timing recovery and sequence detection scheme has been described. The scheme works relatively well even at very low SNRs, where conventional timing recovery schemes fail. This advantage over conventional schemes makes the proposed timing recovery scheme very attractive for deployment in conjunction with powerful codes which enable the communication system to operate in the very low SNR regime. The tracking and jitter performance of the proposed algorithm has also been analyzed and compared with conventional timing recovery algorithms. BER and cycle slip probability of the proposed scheme are also compared with those of the MMSE technique.

REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. ICC'93*, Geneve, Switzerland, May 1993, pp. 1064–1070.
- [2] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.
- [3] W. Zhang and R. R. Spencer, "Timing recovery for backplane ethernet," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 8, pp. 1711–1723, Aug. 2007.
- [4] F. A. Musa and A. C. Carusone, "Modeling and design of multilevel bang-Bang CDRs in the presence of ISI and noise," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 10, pp. 2137–2147, Oct. 2007.
- [5] 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.
- [6] K. Mueller and M. Muller, "Timing recovery in digital synchronous data receivers," *IEEE Trans. Commun.*, vol. COM-24, no. 5, pp. 516–531, May 1976.
- [7] H. Shafiee, "Timing recovery for sampling detectors in digital magnetic recording," in *Proc. IEEE Conf. Commun.*, 1996, pp. 577–581.
- [8] 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.

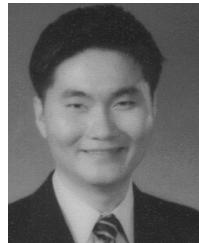
- [9] J.-Y. Lin and C.-H. Wei, "Adaptive nonlinear decision feedback equalization with channel estimation and timing recovery in digital magnetic recording systems," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 42, no. 3, pp. 196–206, Mar. 1995.
- [10] S. Qureshi, "Timing recovery for equalized partial-response systems," *IEEE Trans. Commun.*, vol. COM-24, no. 12, pp. 1326–1331, Dec. 1976.
- [11] J. W. M. Bergmans and H. W. Wong-Lam, "A class of data-aided timing-recovery schemes," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 1819–1827, Feb./Mar./Apr. 1995.
- [12] P. Kovintavewat, J. R. Barry, M. F. Erden, and E. Kurtas, "A new timing recovery architecture for fast convergence," in *Proc. IEEE ISCAS*, May 2003, vol. 2, no. 1, pp. II-13–II-16.
- [13] P. Kovintavewat, J. R. Barry, M. F. Erden, and E. Kurtas, "Per-survivor timing recovery for uncoded partial response channels," in *Proc. IEEE ICC*, Jun. 2004, vol. 27, no. 1, pp. 2715–2719.
- [14] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo-equalization," *Eur. Trans. Telecommun.*, vol. 6, pp. 507–511, Sep. 1995.
- [15] J. R. Barry, A. Kavcic, S. W. McLaughlin, A. Nayak, and W. Zeng, "Iterative timing recovery," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 89–102, Jan. 2004.
- [16] J. Moon, "Joint data and phase detection at low SNRs," Univ. Minnesota Invention Disclosure No. Z01024, 2000.
- [17] J. Riani, J. W. M. Bergmans, S. van Beneden, and A. Immink, "Data-aided timing recovery for recording channels with data-dependent noise," *IEEE Trans. Magn.*, vol. 42, no. 11, pp. 3752–3759, Nov. 2006.
- [18] W. Zeng, M. F. Erden, A. Kavcic, E. M. Kurtas, and R. C. Venkataramani, "Trellis-based optimal baud-rate timing recovery loops for magnetic recording systems," *IEEE Trans. Magn.*, vol. 43, no. 7, pp. 3324–3332, Jul. 2007.
- [19] C. N. Georghiades and D. L. Snyder, "The expectation-maximization algorithm for symbol unsynchronized sequence detection," *IEEE Trans. Commun.*, vol. 39, no. 1, pp. 54–61, Jan. 1991.
- [20] 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.
- [21] J. Moon and J. Lee, "Joint timing recovery and data detection with applications to magnetic recording," *IEEE Trans. Magn.*, vol. 42, no. 10, pp. 2576–2578, Oct. 2006.
- [22] U. Mengali and A. N. D'Andrea, *Synchronization Techniques for Digital Receivers*. New York: Kluwer Academic/Plenum Publishers, 1997.
- [23] J. W. M. Bergmans, *Digital Baseband Transmission and Recording*. Boston, MA: Kluwer Academic, 1996.
- [24] A. De Gloria, D. Grosso, M. Olivieri, and G. Restani, "A novel stability analysis of a PLL for timing recovery in hard disk drives," *IEEE Trans. Circuits Syst. I, Fundam. Theory Applicat.*, vol. 46, no. 8, pp. 1026–1031, Aug. 1999.



Jaekyun Moon (S'89–M'90–SM'97–F'05) received the B.S.E.E. degree from the State University of New York at Stony Brook and the M.S. and Ph.D. degrees in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 1987 and 1990, respectively.

Since 1990, he has been with the faculty of the Department of Electrical and Computer Engineering at the University of Minnesota. In 2001, he co-founded Bermai, Inc., a fabless semiconductor start-up, and served as founding President and CTO. His research interests are in the area of channel characterization, signal processing, and coding for data storage and digital communication.

Dr. Moon received the 1994–1996 McKnight Land-Grant Professorship from the University of Minnesota, Minneapolis. He also received the IBM Faculty Development Awards as well as the IBM Partnership Awards. He was awarded the National Storage Industry Consortium (NSIC) Technical Achievement Award for the invention of the maximum transition run (MTR) code, a widely used error-control/modulation code in commercial storage systems. He served as Program Chair for the 1997 IEEE Magnetic Recording Conference. He is also Past Chair of the Signal Processing for Storage Technical Committee of the IEEE Communications Society. He served as a Guest Editor for the 2001 IEEE JOURNAL ON SELECTED AREAS OF COMMUNICATIONS issue on Signal Processing for High Density Recording. He also served as an Editor for IEEE TRANSACTIONS ON MAGNETICS in the area of signal processing and coding for 2001–2006.



Jaewook Lee (S'07) received the B.S. and M.S. degrees in electrical engineering from Yonsei University, Seoul, Korea, in 1996 and 1998, respectively. He is currently working toward the Ph.D. degree in electrical engineering at the University of Minnesota, Minneapolis.

From 1998 to 2006, he was a Senior Engineer at Samsung Electronics Co. Ltd., Suwon, Korea, where he worked on the implementation of partial response maximum likelihood (PRML) IC for optical disc drive (ODD) and hard disc drive (HDD). During the summer of 2007, he was an Intern at Quantum Corporation, Irvine, CA, where he worked on the simulation of MMSE timing interpolator and the adaptation of fractionally spaced equalizer (FSE) for magnetic tape drive. His current research interests include timing recovery, automatic gain control (AGC), equalization, iterative decoding, error control code, and digital signal processing techniques applied to digital data storage.