Per-Survivor Timing Recovery for Uncoded Partial Response Channels

Piya Kovintavewat and John R. Barry
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332

M. Fatih Erden and Erozan Kurtas
Seagate Technology
1251 Waterfront Place, Pittsburgh, PA 15222

Abstract— **A conventional receiver performs timing recovery and equalization separately. Specifically, conventional timing recovery is based on a phase-locked loop that relies on the decision provided by its own symbol detector. We propose a new timing recovery scheme based on *per-survivor processing* (PSP) that jointly performs timing recovery and equalization for uncoded partial response channels. In the proposed scheme, each survivor of the Viterbi algorithm maintains its own estimate of the timing offset, and this estimate is updated according to the history data associated with the survivor path. As compared to conventional timing recovery at BER = 10^{-4}, the proposed scheme can provide a 0.5 dB gain in SNR.**

I. Introduction

Timing recovery is the process of synchronizing the sampler with the received analog signal. The quality of synchronization has a dominant impact on overall performance.

Theoretically, joint maximum-likelihood (ML) estimation of the timing offset and the data sequence is a preferred method of synchronization [1] but its complexity is prohibitive. A solution based on the extended Kalman filters [2] is also complex. In practice, a conventional receiver performs timing recovery and ML equalization separately, as shown in Fig. 1. Specifically, conventional timing recovery is based on a phase-locked loop (PLL) [3] that relies on the decision provided by a symbol detector, which can be either a Viterbi detector [4] with a short decision delay or a memoryless multi-level slicer. However, the former has a fundamental trade-off between the reliability and the decision delay, whereas the latter might yield an unreliable decision.

To overcome this drawback, a reliable decision with *zero* decision delay can be extracted by utilizing the already-given information inside the trellis structure [4]. Specifically, each state transition in the trellis uniquely specifies a corresponding symbol. Then, at least one state transition in each trellis stage will correspond to the correct decision. Utilizing that decision for the timing update operation will improve the performance of timing recovery. The idea of using the information available in the trellis to estimate other unknown parameters is known as *per-survivor processing* (PSP) [5]. PSP has been employed in many applications, including channel identification, adaptive ML sequence detector, and phase/carrier recovery [5]-[6].

The PSP-based timing recovery scheme was developed in [7] for achieving fast convergence in magnetic recording channels. Since the channel model used in [7] is complex, it is hard to fully explore its architecture. In this paper, we instead focus on a *simple* channel model, namely a perfectly equalized partial response (PR) channel model. Then, we investigate the PSP-based timing recovery scheme in detail and compare its performance with conventional schemes.

This paper is organized as follows. Section II describes our channel model and explains how conventional timing recovery works. The PSP-based timing recovery scheme is described in Section III, and its performance is compared with conventional timing recovery in Section IV. Finally, Section V concludes this paper.

II. Channel Description

We consider the perfectly equalized PR-IV channel model shown in Fig. 1, where the readback signal can be written as

$$s(t) = \sum_{k=0}^{L-1} a_k h(t - kT - \tau_k) + n(t), \tag{1}$$

where $a_k \in \{\pm 1\}$ is an input data sequence of length L with bit period T, $h(t) = p(t) - p(t-2T)$ is a PR-IV pulse, $p(t) = \sin(\pi t/T)/(\pi t/T)$ is an ideal zero-excess-bandwidth Nyquist pulse, and $n(t)$ is additive white Gaussian noise (AWGN) with two-sided power spectral density $N_0/2$. The timing offset, τ_k, is modeled as a random walk model according to $\tau_{k+1} = \tau_k + \mathcal{N}(0, \sigma_w^2)$, where σ_w determines the severity of the timing offset. The random walk model is chosen because of its simplicity to represent a variety of channels by changing only one parameter. We also assume perfect acquisition by setting $\tau_0 = 0$.

At the receiver, the readback signal $s(t)$ is filtered by a low-pass filter, whose impulse response is $p(t)/T$, to eliminate the out-of-band noise, and sampled at time $kT + \hat{\tau}_k$, creating

$$y_k = y(kT + \hat{\tau}_k) = \sum_i a_i h(kT + \hat{\tau}_k - iT - \tau_i) + n_k, \tag{2}$$

where $\hat{\tau}_k$ is the receiver's estimate of τ_k, and n_k is *i.i.d.* zero-mean Gaussian random variable with variance $\sigma_n^2 = N_0/(2T)$.

Conventional timing recovery is based on a PLL, which consists of a timing-error detector (TED), a loop filter, and a voltage-controlled oscillator (VCO), as depicted in Fig. 1. A decision-directed TED [3] computes the receiver's estimate of the timing error $\epsilon_k = \tau_k - \hat{\tau}_k$ using the well-known Mueller and Müller (M&M) TED algorithm [8] according to

$$\hat{\epsilon}_k = \frac{3T}{16} \left\{ y_k \hat{r}_{k-1} - y_{k-1} \hat{r}_k \right\}, \tag{3}$$

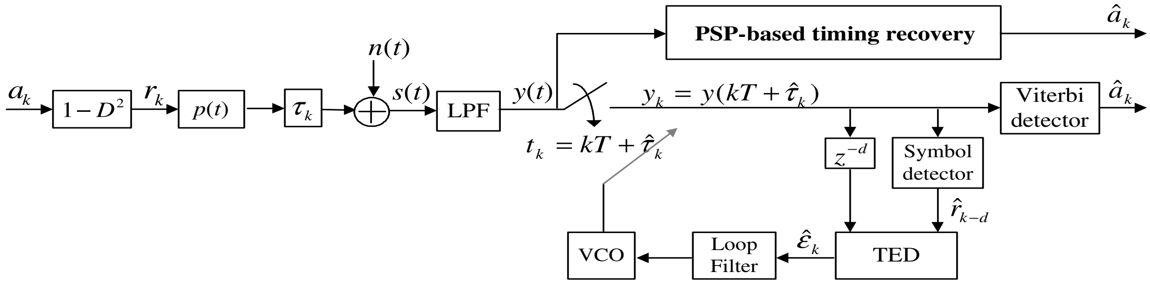

Fig. 1. The perfectly equalized PR-IV channel model with timing recovery.

where $\hat{r}_k$ is the k-th estimate of the noiseless *channel output* $r_k \in \{0, \pm 2\}$. The constant $3T/16$ assures that there is no bias at high signal-to-noise ratio (SNR) so that $E[\hat{\epsilon}_k] = \epsilon_k$. Because perfect acquisition is assumed and our model has no frequency offset component, the sampling phase offset is then updated by a first-order PLL according to

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \alpha\hat{\epsilon}_k, \tag{4}$$

where α is a PLL gain parameter [3]. Eventually, the Viterbi detector performs ML equalization to determine the most likely input data sequence.

It is apparent from Fig. 1 that the total delay in the timing loop results from the *decision delay*, d, introduced by the symbol detector. The instantaneous *hard decision* (i.e., $d = 0$) can be extracted by a simple ternary symbol-by-symbol decision with threshold at ± 1, i.e.,

$$\hat{r}_k = \begin{cases} 2 & \text{if } y_k > 1 \\ -2 & \text{if } y_k < -1 \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

but it might be very unreliable. An improved decision can be obtained from the Viterbi detector with a (short) decision delay of d. This is done by choosing the *best* survivor path at each time instant, and then the *tentative decision*, $\hat{r}_{k-d}$, is found by moving d steps backward along that survivor path. Clearly, there is a trade-off between the reliability and the decision delay since reliability can be improved by increasing the decision delay. However, a large delay is undesirable because it slows the PLL's response to time-varying timing offsets.

Another solution to obtain a good decision with zero decision delay is to utilize the PSP technique, which will be discussed in the next section.

III. PSP-Based Timing Recovery

PSP was first applied to the application of reduced-state sequence estimation [9]. The general PSP concept and its various applications were later introduced in [5]. PSP is a technique for jointly estimating the data sequence and unknown parameters, such as the channel coefficients, the carrier phase, and so forth. Note that the PSP concept is quite general because it results in different solutions for a given application. In this paper, we apply PSP to develop a new timing recovery scheme called *PSP-based timing recovery*, which jointly performs timing recovery and ML equalization, as shown in Fig. 1.

(A-1) Initialize $\Phi_0(p) = 0$ for $\forall p$
*(A-2) Initialize $\hat{\tau}_0(p) = 0$ for $\forall p$
(A-3) For $k = 0, 1, \ldots, L+\nu-1$
(A-4) For $q = 0, 1, \ldots, Q-1$
*(A-5) $y_k(p) = y(kT + \hat{\tau}_k(p))$ for $\forall p$
(A-6) $\rho_k(p,q) = |y_k(p) - \hat{r}(p,q)|^2$ for $\forall p$
(A-7) $\pi_{k+1}(q) = \arg\min_p\{\Phi_k(p) + \rho_k(p,q)\}$
(A-8) $\Phi_{k+1}(q) = \Phi_k(\pi_{k+1}(q)) + \rho_k(\pi_{k+1}(q), q)$
(A-9) $\mathbf{S}_{k+1}(q) = [\,\mathbf{S}_k(\pi_{k+1}(q)) \mid \pi_{k+1}(q)\,]$
*(A-10) $\hat{\tau}_{k+1}(q) = \hat{\tau}_k(\pi_{k+1}(q)) + \alpha\frac{3T}{16}\{y_k(\pi_{k+1}(q))\hat{r}(\pi_k(\pi_{k+1}(q)), \pi_{k+1}(q)) - y_{k-1}(\pi_k(\pi_{k+1}(q)))\hat{r}(\pi_{k+1}(q), q)\}$
(A-11) End
(A-12) End
(A-13) Extract $\hat{\mathbf{a}}$ from the survivor path that minimizes $\Phi_{L+\nu}$

Fig. 2. PSP-MM algorithm, where the lines beginning with * are the additional steps beyond the conventional Viterbi algorithm.

PSP-based timing recovery works in a similar fashion as the Viterbi algorithm does, except with an additional timing update operation. The key idea of PSP-based timing recovery is to sample the received analog signal using different sampling phase offsets associated with each state transition. Additionally, each survivor path has its own PLL to update the sampling phase offset. For simplicity, we first restrict ourselves to the M&M TED algorithm when performing the timing update operation. As a result, we shall refer to the PSP-based timing recovery scheme with the M&M TED as "PSP-MM."

A. PSP-MM Algorithm

Fig. 2 shows the PSP-MM algorithm, where the lines beginning with * are the additional steps beyond the conventional Viterbi algorithm. The PSP-MM algorithm is explained below.

Consider the PR-IV trellis structure in Fig. 3. Let $\Psi_k = \{a_{k-1}\, a_{k-2}\}$ denote the *state* at time k (or the k-th *stage*). There are $Q = 2^\nu = 4$ states in this trellis labeled as state 0 to state 3, where ν is the PR-IV channel memory. Let (p,q) be

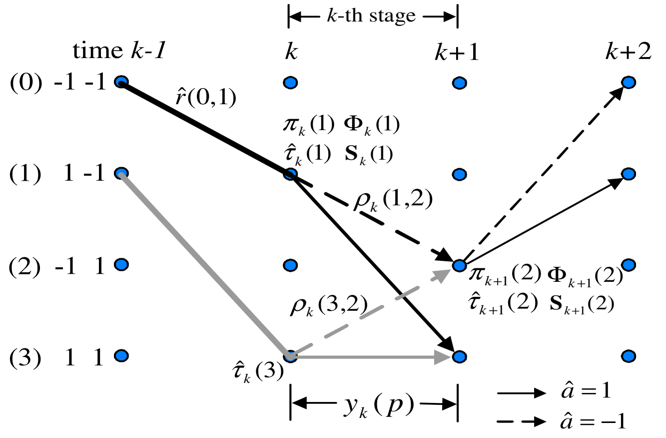

Fig. 3. The PR-IV trellis structure explaining how PSP-MM performs.

the *state transition* from state p to state q, and $\pi_k(p)$ denote a *predecessor* for state p at time k, defined as the starting state associated with the *best* state transition. We define $\hat{\tau}_k(p)$ as the k-th sampling phase offset for state p at time k, which is used to sample $y(t)$ at time k for the state transitions emanating from state p at time k, e.g., $y_k(p) = y(kT + \hat{\tau}_k(p))$, where $y_k(p)$ is the k-th sampler output for state p at time k.

Consider the k-stage of the trellis. There are two state transitions arriving at state 2 at time $k+1$, i.e., $(1,2)$ and $(3,2)$. We first sample $y(t)$ using $\hat{\tau}_k(1)$ and $\hat{\tau}_k(3)$ to obtain $y_k(1)$ and $y_k(3)$, respectively. Next, we compute two branch metrics $\rho_k(1,2)$ and $\rho_k(3,2)$ according to (A-6), where $\hat{r}(p,q)$ is the channel output associated with (p,q). Then, the starting state associated with the best state transition leading to state 2 at time $k+1$ is chosen according to (A-7).

Suppose $(1,2)$ is the best state transition leading to state 2 at time $k+1$ so that $\pi_{k+1}(2) = 1$. The path metric for state 2 at time $k+1$, $\Phi_{k+1}(2)$, is updated by (A-8), and the survivor path for state 2 at time $k+1$, $\mathbf{S}_{k+1}(2)$, is extended according to (A-9). Then, the next sampling phase offset, $\hat{\tau}_{k+1}(2)$, is updated based on (A-10) using the information from $\mathbf{S}_{k+1}(2)$. This $\hat{\tau}_{k+1}(2)$ will be used to sample $y(t)$ at time $k+1$ for the state transitions emanating from state 2 at time $k+1$. We follow these steps according to the Viterbi algorithm for an entire received signal. Finally, the decision is made by choosing the survivor path that has the minimum path metric.

Beyond the conventional Viterbi algorithm, PSP-MM needs new storage requirement for: 1) the sampling phase offsets and 2) the sampler outputs. However, only sampling phase offsets and sampler outputs of the *current* and *previous* stages are needed to be stored, thus minimizing extra memory. Furthermore, it is clear that PSP-MM requires one PLL for each survivor path. Thus, for a PR-IV channel, the complexity of timing recovery is four times the complexity of conventional timing recovery.

B. New Timing Error Detector

The PSP-MM described above does not exploit the *future* information available in the trellis, i.e., the channel output at the next time instant. Consider the case where we are at state 2 at time $k+1$, we would know exactly that there will be two state transitions emanating from this state, i.e., $(2,0)$ and $(2,1)$. Since these two future channel outputs, $\hat{r}(2,0)$ and $\hat{r}(2,1)$, are available at time k, it might be a good idea to incorporate them for the timing update operation at time k.

To do so, we need to develop a TED algorithm that is able to use future information. One such TED algorithm can be found by minimizing the log-likelihood function of the samples $\{y_k\}$ [1] according to

$$\begin{aligned} L(\boldsymbol{y}|\hat{\boldsymbol{r}},\epsilon) &= \sum_m \left| y_m - \sum_n \hat{r}_n p((m-n)T - \tau + \hat{\tau}) \right|^2 \\ &= K - \sum_m y_m \sum_n \hat{r}_n p((m-n)T - \tau + \hat{\tau}), \end{aligned} \tag{6}$$

where K is a constant independent of $\hat{\tau}$, $y_m = y(mT + \hat{\tau})$, τ is the actual timing offset, and $\hat{\tau}$ is an estimate of τ.

Since we are concerned with an error feedback algorithm, only $\hat{\tau}$ close to τ is of interest. Thus, the timing error signal can be obtained by differentiating (6) with respect to $\hat{\tau}$, i.e.,

$$\frac{\partial L(\boldsymbol{y}|\hat{\boldsymbol{r}},\epsilon)}{\partial \hat{\tau}} = -\sum_m y_m \sum_n \hat{r}_n \dot{p}((m-n)T), \tag{7}$$

where $\dot{p}(kT)$ is the derivative of $p(t)$ evaluated at time kT, which can be expressed as

$$\dot{p}(kT) = \begin{cases} 0 & k = 0 \\ \frac{1}{kT}(-1)^k & \text{otherwise} \end{cases}. \tag{8}$$

With a symmetric property, the estimated timing error at time k for four observations can be written as

$$\begin{aligned} \hat{\epsilon}_k &= -\frac{180T}{3086} \sum_{m=k-2}^{k+1} y_m \sum_{n=k-2}^{k+1} \hat{r}_n \dot{p}((m-n)T) \\ &= \frac{180T}{3086} \{ y_{k+1}(\hat{r}_k - 0.5\hat{r}_{k-1} + \hat{r}_{k-2}/3) \\ &\quad + y_k(-\hat{r}_{k+1} + \hat{r}_{k-1} - 0.5\hat{r}_{k-2}) \\ &\quad + y_{k-1}(0.5\hat{r}_{k+1} - \hat{r}_k + \hat{r}_{k-2}) \\ &\quad + y_{k-2}(-\hat{r}_{k+1}/3 + 0.5\hat{r}_k - \hat{r}_{k-1}) \}, \end{aligned} \tag{9}$$

where we use $y_{k+1} = y(kT + T + \hat{\tau}_k)$ assuming that the timing offset is slowly varying. The constant $180T/3086$ is introduced to ensure that there is no bias at high SNR so that $E[\hat{\epsilon}_k] = \epsilon_k$. We shall refer to this TED as "4S-TED," where "4S" stands for the samples taken from time $k-2$ to $k+1$ that are used to compute $\hat{\epsilon}_k$. Note that when using the samples only at time $k-1$ and k, (9) reduces to the M&M TED (by ignoring the constant term of both TEDs).

It is worth exploring the characteristics of both TEDs, which can be determined by the *timing function* [3], defined as the mean of $\hat{\epsilon}_k$ assuming that all decisions are correct and the input data symbols are uncorrelated with unit energy. For a PR-IV channel, the timing function of the M&M TED is given by

$$\begin{aligned} &E[\hat{\epsilon}_k|\hat{r}_{k-1} = r_{k-1}, \hat{r}_k = r_k] \\ &= \frac{3T}{16}\{-h(-T-\epsilon) + 2h(T-\epsilon) - h(3T-\epsilon)\}, \end{aligned} \tag{10}$$

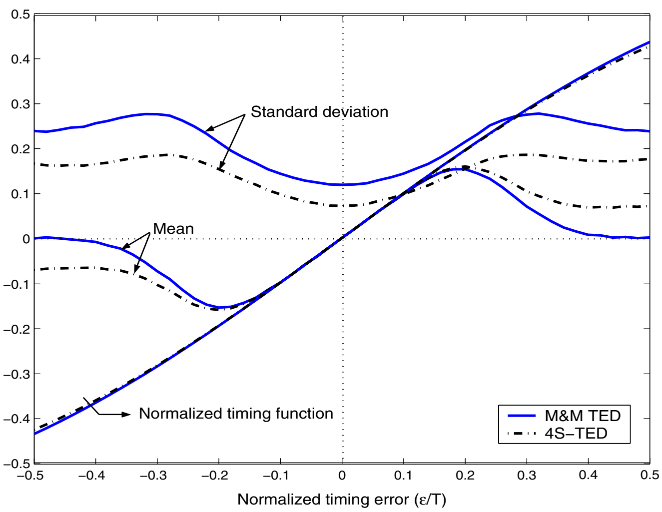

Fig. 4. The mean and the standard deviation of different TEDs for a PR-IV channel at SNR = 10 dB.

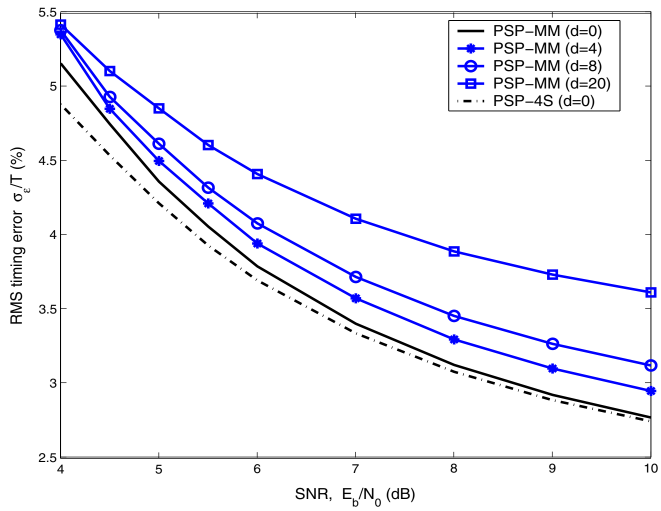

Fig. 5. Performance comparison of PSP-based timing recovery as a function of SNRs.

whereas that of the 4S-TED can be expressed as

$$\begin{aligned}&E[\hat{\epsilon}_k|\hat{r}_{k-2} = r_{k-2}, \hat{r}_{k-1} = r_{k-1}, \hat{r}_k = r_k, \hat{r}_{k+1} = r_{k+1}] \\ &= \frac{180T}{3086}\{-h(-\epsilon) + 6h(T-\epsilon) - h(2T-\epsilon) \\ &\quad -h(-3T-\epsilon)/3 + h(-2T-\epsilon) - 8h(-T-\epsilon)/3 \\ &\quad -8h(3T-\epsilon)/3 + h(4T-\epsilon) - h(5T-\epsilon)/3\}. \quad (11)\end{aligned}$$

The mean and the standard deviation of $\{\hat{\epsilon}_k\}$ given in (3) and (9) based on instantaneous decisions as a function of the normalized timing error ϵ/T at a per-bit SNR, E_b/N_0, of 10 dB are plotted in Fig. 4, assuming that we have access to the correct future information. Clearly, both timing functions are odd symmetric with respect to $\epsilon = 0$. Thus, regardless of the TED used, the sampling phase offset updated according to (4) will settle down in the steady state at $\epsilon = 0$. Observe that the mean of both TEDs is approximately proportional to ϵ/T over a range of $\pm 20\%$ about the origin. As expected, the standard deviation of the 4S-TED is lower than that of the M&M TED because more information is used in evaluating the estimated timing error. Therefore, the 4S-TED is more robust to the noise in the timing error signal than the M&M TED.

From this point on, we shall denote the PSP-based timing recovery scheme with the 4S-TED as "PSP-4S." Unlike PSP-MM, for a PR-IV channel, the complexity of timing recovery is eight times that of conventional timing recovery because PSP-4S requires one PLL for each state transition in one stage of the trellis.

C. Note on Conventional Timing Recovery

We can also explain how conventional timing recovery works in the context of the trellis structure. This will show that it is in fact a special case of PSP-MM.

Practically, conventional timing recovery employs the *same* sampling phase offset $\hat{\tau}_k$ to sample $y(t)$ for all state transitions at time k. Then, the *same* decision (either the hard decision $\hat{r}_k$ or the tentative decision $\hat{r}_{k-d}$ found by tracing back d steps along the *best* survivor path chosen at time k) is used to compute the estimated timing error $\hat{\epsilon}_k$ for all states, which finally results in the *same* $\hat{\tau}_{k+1}$ for all states after updating it.

Therefore, PSP-based timing recovery differs from conventional timing recovery in the sense that: 1) it uses different sampling phase offsets associated with each state transition to sample $y(t)$; and 2) it employs the instantaneous decision with zero decision delay associated with each state transition to compute the estimated timing error.

IV. Numerical Results

In simulation, unless otherwise specified, we consider $\sigma_w/T = 0.5\%$ and employ the PLL gain parameter, α, designed to recover phase change within 100 symbols based on a linearized model of PLL [3], assuming that the slope of the timing function is one at origin and there is no noise in the system. The α's designed for the delays of $0, 4T, 8T$ and $20T$ are 0.030, 0.027, 0.025 and 0.019, respectively.

We first explore how the decision delay affects the performance of timing recovery. In doing so, we consider the PSP-MM scheme where we have access to all decisions $\{\hat{r}_{k-d}\}$ (at any d steps earlier) associated with each survivor path. Fig. 5 compares the performance of different PSP-MMs, where the RMS timing error $\sigma_\epsilon = \sqrt{E[(\tau_k - \hat{\tau}_k)^2]}$ is plotted as a function of SNRs. Apparently, PSP-MM with $d = 0$ yields the best performance. This can be confirmed by plotting σ_ϵ/T performance as a function of α's at SNR = 8 dB in Fig. 6. Again, PSP-MM with $d = 0$ is better than that with $d \neq 0$ for all α's. Results imply that the decision delay has a tremendous impact on overall performance. Therefore, it is desirable to use the decision with zero decision delay whenever possible.

Figs. 5 and 6 also show the performance of the PSP-4S with $d = 0$. In Fig. 5, PSP-4S performs better than PSP-MM at low SNRs. This is because the future information used in PSP-4S helps improve the performance of timing recovery, especially when the uncertainty is high. Fig. 6 indicates that

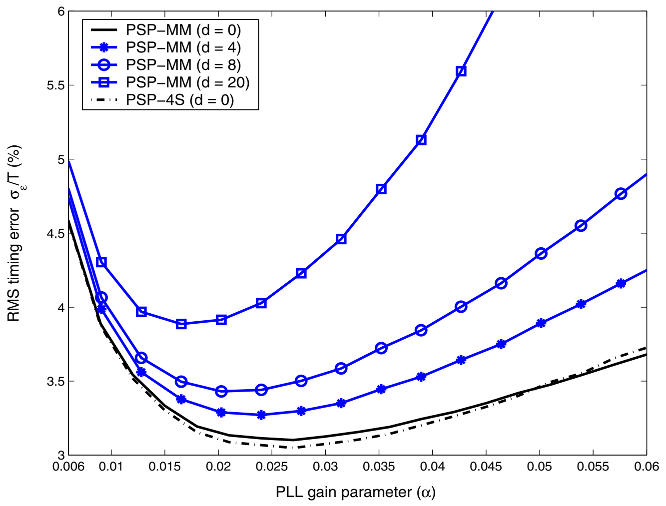

Fig. 6. Performance comparison of PSP-based timing recovery as a function of α's at SNR = 8 dB.

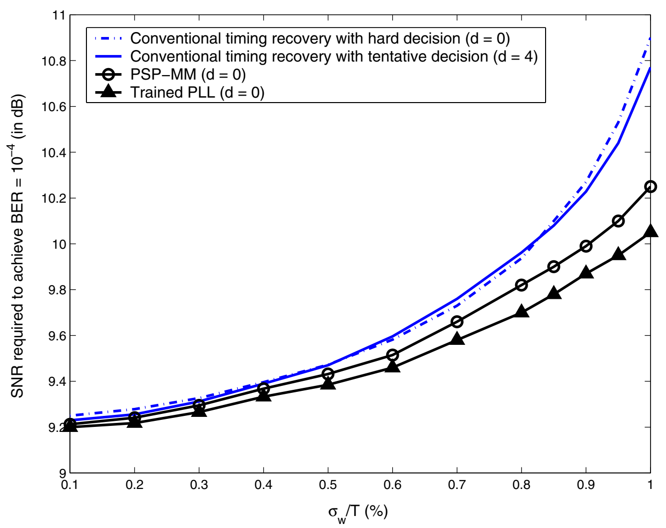

Fig. 7. Performance comparison of different timing recovery schemes.

PSP-4S yields slightly lower σ_ϵ/T performance than PSP-MM for small α. However, it starts performing worse than PSP-MM when α is large. Since PSP-4S provides only small gain over PSP-MM and the complexity of PSP-4S is much higher than that of PSP-MM, PSP-MM is then preferred. From this point on, we shall consider only the PSP-MM with $d = 0$ when comparing with conventional timing recovery.

Finally, we compare PSP-MM with conventional timing recovery by plotting SNR requirement for bit-error rate (BER) of 10^{-4} as a function of σ_w/T's in Fig. 7. The curve labeled "Trained PLL" is conventional timing recovery whose PLL has access to all correct decisions, thus serving as a lower bound for all timing recovery schemes that are based on PLL. Obviously, PSP-MM performs better than conventional timing recovery, especially when σ_w/T is large. As shown in Fig. 7, PSP-MM is 0.5 dB better than conventional timing recovery when operating in the system with $\sigma_w/T = 1\%$. Although conventional timing recovery with hard decision seems to perform comparably to that with tentative decision, this is not true when the channel is complex, e.g., the channel with large channel memories or with arbitrary coefficients. That is why conventional timing recovery practically uses the tentative decision provided by the Viterbi detector in most applications.

The reason that PSP-MM performs better than conventional timing recovery can be intuitively explained as follows. At each time instant, at least one state transition in each trellis stage will correspond to the correct decision. Using that decision to perform the timing update operation will then improve the performance of timing recovery. In other words, PLL is *fully trained* if a correct path is chosen. Following this idea for an entire received signal, the overall system performance will be improved.

V. Conclusion

We proposed the PSP-based timing recovery scheme to jointly perform timing recovery and ML equalization for uncoded partial response channels.

It is apparent that the delay in the timing loop affects overall performance. That is why PSP-MM with $d = 0$ performs better than that with $d \neq 0$. Therefore, PSP-based timing recovery has the advantage of reducing the delay in the timing loop. Since PSP-4S provides only a small gain over PSP-MM, PSP-MM is then more desirable than PSP-4S because it has less complexity. Finally, we have shown that PSP-MM yields better performance than conventional timing recovery, especially when the timing error is large. Specifically, PSP-MM provides a 0.5 dB gain over conventional timing recovery when $\sigma_w/T = 1\%$.

As the complexity of PSP-based timing recovery is high, all advantages obtained from PSP-based timing recovery must be balanced against the increased implementation cost.

References

[1] H. Meyr, M. Moeneclaey, and S. A. Fechtel, *Digital communication receivers: synchronization, channel estimation, and signal processing*, New York: John Wiley & Sons, Inc., 1998.

[2] R. A. Iltis, "A Bayesian maximum-likelihood sequence estimation algorithm for a *priori* unknown channels and symbol timing," *IEEE J. Selected Areas Commun.*, vol. 10, pp. 579-588, April 1992.

[3] J. W. M. Bergmans, *Digital baseband transmission and recording*, Boston/London/Dordrecht: Kluwer Academic Publishers, 1996.

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

[5] R. Raheli, A. Polydoros, and C. K. Tzou, "Per-survivor processing: a general approach to MLSE in uncertain environments," *IEEE Trans. Commun.*, vol. 43, no. 2, pp. 354-364, Feb./Mar./Apr. 1995.

[6] A. V. Coralli, P. Salmi, S. Cioni, G. E. Corazza, and A. Polydoros, "A performance review of PSP for joint phase/frequency and data estimation in future broadband satellite networks," *IEEE J. Selected Areas Commun.*, vol. 19, no. 12, pp. 2298-2309, December 2001.

[7] P. Kovintavewat, M. F. Erden, E. Kurtas, and J. R. Barry, "A new timing recovery architecture for fast convergence," *IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 13-16, May 2003.

[8] K. H. Mueller and M. Müller, "Timing recovery in digital synchronous data receivers," *IEEE Trans. Commun.*, vol. 24, no. 5, pp. 516-531, May 1976.

[9] M. V. Eyuboğlu and S. U. H. Qureshi, "Reduced-state sequence estimation for coded modulation on intersymbol interference channels," *IEEE J. Selected Areas Commun.*, vol. 7, pp. 989-995, August 1989.

