Complexity Reduction of Modified Per-Survivor Iterative Timing Recovery for Partial Response Channels

Chanon Warisarn[1], **Piya Kovintavewat**[2], and **Pornchai Supnithi**[3], Non-members

ABSTRACT

The problem of timing recovery operating at low signal-to-noise ratio has been recently solved by a modified per-survivor iterative timing recovery (MPS-ITR) proposed in [1], which jointly performs timing recovery, equalization, and error-correction decoding. In practice, this scheme exploits a split-preamble strategy in conjunction with a per-survivor soft-output Viterbi algorithm equalizer to make it more robust against severe timing jitters or cycle slips. Although the MPS-ITR outperforms existing iterative timing recovery schemes [1], its complexity is extremely high. Therefore, this paper proposes a reduced-complexity MPS-ITR scheme (denoted as MPS-ITR-M) to make it more implementable in real-life applications. This is achieved by applying the M-algorithm [2] to the MPS-ITR. Numerical results show that at low-to-moderate complexity, the MPS-ITR-M performs better than other schemes.

Keywords: M-Algorithm, Per-Survivor Iterative Timing Recovery, Synchronization, Timing Acquisition

1. INTRODUCTION

Timing recovery is the process by which a receiver ensures that the received analog signal is sampled at the correct times. Sampling at the wrong times can have a devastating impact on overall system performance. The large coding gains of iterative error-correction codes (ECCs) enable reliable communication at very low signal-to-noise ratio (SNR). This means that timing recovery must be performed at an SNR lower than ever before. A conventional receiver performs timing recovery and turbo equalization [3] separately. Specifically, conventional timing recovery ignores the presence of ECCs and thus fails to work properly when the SNR is low enough.

To improve the performance of the conventional receiver, Kovintavewat *et al.* [4] proposed a per-survivor iterative timing recovery (PS-ITR) scheme, which jointly performs timing recovery, equalization, and error-correction decoding. It is realized by first applying the per-survivor processing (PSP) technique [5], a technique of jointly estimating a data sequence and unknown parameters, to the soft-output Viterbi algorithm (SOVA) [6], resulting in a per-survivor SOVA equalizer, denoted as "PSP-SOVA" [4]. Hence, PSP-SOVA iteratively exchanges soft information with a soft-in soft-out (SISO) decoder. As investigated in [4], the PS-ITR outperforms the conventional receiver because it can automatically correct a cycle slip [7] after a few number of turbo iterations.

To make the PS-ITR more robust against severe timing jitters, a modified per-survivor iterative timing recovery (MPS-ITR) scheme has been proposed in [1], which iteratively exchanges soft information between the *modified* PSP-SOVA and the decoder. This modified PSP-SOVA uses a new split-preamble strategy, which divides a preamble into two parts, each of which is used to adjust the branch metric calculation in PSP-SOVA so as to guarantee that the survivor path occurs in a correct direction. It has been shown [1] that the MPS-ITR performs better than the PS-ITR, and both outperforms the conventional receiver, especially when the timing jitter is severe.

The M-algorithm was first introduced by Simmons and Mohan [2], which has later been employed in many applications, including source coding [2] and channel decoding [8]. For example, Iki *et al.* [9] proposed the application of M-algorithm and stack algorithm to the trellis shaping with peak-to-average power ratio reduction for single-carrier signal. Additionally, the M-algorithm has also been used in data storage systems. For instance, a simplified noise-predictive partial response maximum likelihood system in conjunction with the M-algorithm was proposed in [10] for dual-layered perpendicular magnetic recording channels, and a low-complexity non-iterative detector for magnetic and optical multitrack high-density data storage was proposed in [11], whose detector is based on the M-algorithm.

Since the MPS-ITR scheme has very high complexity, we therefore apply the M-algorithm [2] to the MPS-ITR, resulting in a *reduced-complexity* MPS-

Manuscript received on January 13, 2012 ; revised on June 1, 2012.

[1] The author is with College of Data Storage Innovation, King Mongkut's Institute of Technology Ladkrabang Bangkok, Thailand., E-mail: kwchanon@kmitl.ac.th

[2] The author is with Data Storage Technology Research Center, Nakhon Pathom Rajabhat University Nakhon Pathom, Thailand., E-mail: piya@npru.ac.th

[3] The author is with Telecommunications Engineering Department, King Mongkut's Institute of Technology Ladkrabang Bangkok, Thailand., E-mail: ksupornc@kmitl.ac.th

ITR scheme (denoted as MPS-ITR-M), so as to reduce its complexity and to make it more implementable in real-life applications.

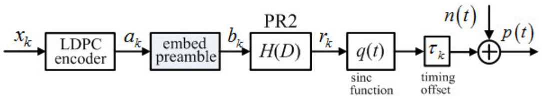

Fig.1: *Data encoding with a PR2 channel model.*

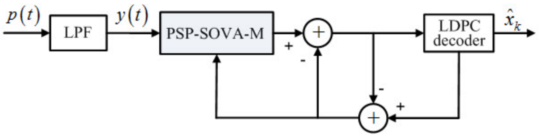

Fig.2: *Proposed iterative timing recovery.*

Additionally, we consider only a coded partial response channel in this paper because this channel is widely used in magnetic recording systems [12, 13]. Thus, it will be shown later that at low-to-moderate complexity, the MPS-ITR-M performs better than the MPS-ITR and the PS-ITR.

This paper is organized as follows. After explaining the channel model in Section 2, Section 3 briefly describes how the MPS-ITR-M works. Complexity comparison is provided in Section 4. Simulation results and discussion are given in Section 5. Finally, Section 6 concludes this paper.

2. CHANNEL MODEL

Consider the coded partial-response (PR) channel [14] in Fig. 1, where $H(D) = \sum_{k=0}^{v-1} h_k D^k = 1 + 2D + D^2$ is a PR2 channel [1, 15], h_k is the k-th channel coefficient, D is the delay operator, and v is channel memory. The message sequence $x_k \in \{0, 1\}$ is encoded by an error-correction encoded and is mapped to a binary sequence $a_k \in \{\pm 1\}$ of length L. Next, a preamble is inserted in a sequence a_k to obtain a sequence $b_k \in \{\pm 1\}$. The readback signal can then be written as

$$p(t) = \sum_k r_k q(t - kT - \tau_k) + n(t) \tag{1}$$

where $r_k = b_k * h_k \in \{0, \pm 2, \pm 4$ is the noiseless channel output, $*$ is the convolution operator, $q(t) = \sin(\pi t/T)/(\pi t/T)$ is an ideal zero-excess-bandwidth Nyquist pulse, T is a bit period, and n(t) is an additive white Gaussian noise with two-sided power spectral density N0/2. The uncertainty in the timing is captured by the timing offset τ_k, which is modeled as a random walk [16] according to $\tau_{k+1} = \tau_k + \mathcal{N}(0, \sigma_w^2)$, where σ_w controls the severity of the timing jitter. The random walk model is chosen because of its simplicity and its ability to represent a variety of channels by changing only one parameter.

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

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

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
$\sigma_n^2 = N_0/(2T)$, i.e.,$n_k \sim \mathcal{N}(0, \sigma_n^2)$.

Conventional timing recovery is based on a PLL [7], which consists of a timing error detector (TED), a loop filter, and a voltage-controlled oscillator (VCO). A decision-directed TED computes the receiver's estimate of the timing error $\varepsilon_k = \tau_k - \hat{\tau}_k$ using the well-known Mueller and Müller (M&M) TED algorithm [17] according to

$$\hat{\varepsilon}_k = \frac{6T}{40}\{y_k \hat{r}_{k-1} - y_{k-1}\hat{r}_k\}, \tag{3}$$

where $\hat{r}_k$ is an estimate of r_k, and the constant $6T/40$ [18] is used to normalize the timing function of the M&M TED in (3) to have unit slope at origin [7]. For simplicity, we assume no frequency offset in the system. Thus, the next sampling phase offset can be updated by a first-order PLL according to

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

where α is a PLL gain parameter [7].

In a conventional setting, conventional timing recovery is followed by a turbo equalizer [3], which iteratively exchanges soft information between the SISO equalizer for the PR2 channel and the SISO decoder for the outer code.

3. PROPOSED SCHEMES

The proposed scheme (MPS-ITR-M) will iteratively exchange soft information between a PSP-SOVA-M module and the SISO decoder as shown in Fig. 2. This PSP-SOVA-M is obtained by applying the M-algorithm [2] to the modified PSP-SOVA proposed in [1], which can be explained how it performs as follows.

The modified PSP-SOVA is developed based on the PSP-SOVA [4] with an aid of the new split-preamble strategy [1]. Specifically, the PSP-SOVA uses the conventional preamble arrangement, which places all C known bits at the beginning of the data sector. On the other hand, the modified PSP-SOVA uses a split a C-bit preamble into two parts. The first part of $C/2$ bits is placed at the beginning of the data sector, and the second part of $C/2$ bits is divided into $C/(2m)$ clusters (e.g., m = 1, 2, or 4 bits), each of which is then embedded uniformly within the user data stream. This split preamble is utilized to adjust the branch metric calculation in PSP-SOVA to ensure that the survivor path occurs in a correct direction. Based on extensive simulation search, we found that the modified PSP-SOVA with the one-bit split-preamble arrangement (i.e., $m = 1$) provides the best performance.

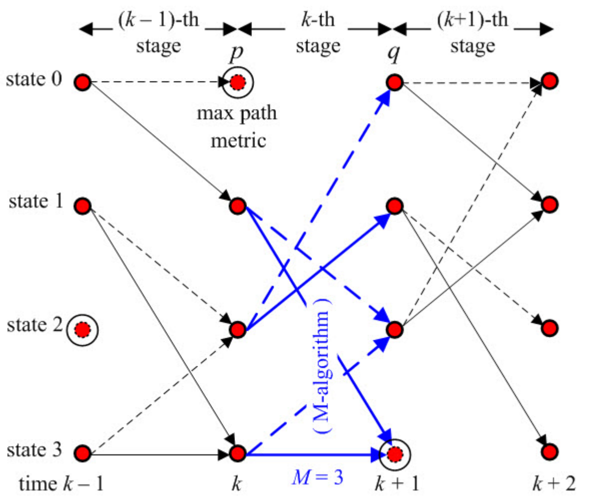

Fig.3: *The PR2 trellis structure illustrating how M-algorithm performs.*

The M-algorithm [2] was originally proposed to reduce complexity of the Viterbi algorithm, which can be described how it works as follows. Fig. 3 shows the trellis diagram of the PR2 channel, which has 4 states. At each time instant, the M-algorithm first finds the minimum path metric leading to each trellis state. Hence, it retains only the M paths (M must be less than the total number of states in one stage of the trellis) with the lowest path metrics among all survivor paths. For example, in Fig. 3, if we assume that $M = 3$ and the state 0 has maximum path metric at time k. As a result, only three states (i.e., state 1, 2, and 3) will be used in branch metric calculation at the k-th stage.

Because the modified PSP-SOVA is developed based on the Viterbi algorithm, its complexity grows exponentially with channel memory [19]. Therefore, to reduce its complexity, we apply the M-algorithm [2] to the modified PSP-SOVA, creating the PSP-SOVA-M. Fig. 4 illustrates the PSP-SOVA-M algorithm, where a constant 6T/40 in (A-11) is only for the PR2 channel, which can be included in the PLL gain parameter. It should be noted that the PSP-SOVA-M works in a same manner as the modified PSP-SOVA [1] does, except that the PSP-SOVA-M has an extra step according to the M-algorithm, which can be briefly explained as follows.

Consider the trellis diagram in Fig. 3 at the k-th stage, where we denote $\mathbf{M}$ as a set of all states (e.g., $\mathbf{M} = \{1, 2, 3\}$) that still remain at time k according to the M-algorithm. To reduce the number of states, a state p will be chosen from $\mathbf{M}$, i.e., $p \in$ M. In other words, only the branches emanating from $\mathbf{M}$ will be

(A-1) Initialize $\mathbf{\Phi}_0(p) = 0$ for $\forall p$
(A-2) Initialize $\hat{\tau}_0(p) = 0$ for $\forall p$
(A-3) For $k = 0, 1, \ldots, L + \upsilon - 1$
(A-4) For $q = 0, 1, \ldots, Q - 1$
(A-5) $y_k(p) = y_k(kT + \hat{\tau}_k(p))$ for $p \in \mathbf{M}$
(A-6) $\rho_k(p,q) = |y_k(p) - \hat{r}(p,q)|^2$ for $p \in \mathbf{M}$
(A-7) If $k = i$ then [i is the preamble position]
For $p \in \mathbf{M}$
If $\hat{b}(p,q) \neq f_i$ then
$\rho_k(p,q) = \Delta$ [Δ is a large number]
End
(A-8) $\pi_{k+1}(q) = \arg\min_{p\in\mathrm{M}}\{\mathbf{\Phi}_k(p) + \rho_k(p,q)\}$
(A-9) $\mathbf{\Phi}_{k+1}(q) = \mathbf{\Phi}_k(\pi_{k+1}(q)) + \rho_k(\pi_{k+1}(q), q)$
(A-10) $\mathbf{S}_{k+1}(q) = [\mathbf{S}_k(\pi_{k+1}(q)) | \pi_{k+1}(q)]$
(A-11) $\hat{\varepsilon} = \frac{6T}{40}\{y_k(\pi_{k+1}(q))\hat{r}(\pi_k(\pi_{k+1}(q)), \pi_{k+1}(q))$
$- y_{k-1}(\pi_k(\pi_{k+1}(q)))\hat{r}(\pi_{k+1}(q), q)\}$
(A-12) $\hat{\tau}_{k+1}(q) = \hat{\tau}_k(\pi_{k+1}(q)) + \alpha\hat{\varepsilon}$
(A-13) $\Delta_{k+1}(q) = \max_{p\in\mathrm{M}}\{\mathbf{\Phi}_k(p) + \rho_k(p,q)\} - \mathbf{\Phi}_{k+1}(p)$
(A-14) Initialize $\hat{L}_k(q) = +\infty$ [Soft decision update]
(A-15) For $j = k - \nu, \ldots, k - \delta$
Compare the two paths merging
in state q (i.e., $\mathbf{\Psi}_{k+1} = q$)
If $\hat{a}_j^{(1)}(\mathbf{\Psi}_{j+1}) \neq \hat{a}_j^{(2)}(\mathbf{\Psi}_{j+1})$ then
Update $\hat{L}_j(\mathbf{\Psi}_{j+1}) = \min(\hat{L}_j(\mathbf{\Psi}_{j+1}), \Delta_{k+1}(q))$
End
(A-16) If $k \geq \delta$ then
Output the soft decision according to $\lambda_k^p = \hat{a}_{k-\delta}\hat{L}_{k-\delta}$,
which can be extracted from the survivor path
that minimizes $\mathbf{\Phi}_{k+1}$
End
End

Fig.4: *The PSP-SOVA-M algorithm.*

used in branch metric calculation at the k-th stage.

As for the new split-preamble strategy, denote $f_k \in \{\pm 1\}$ as a $C/2$-bit preamble that are embedded uniformly within the user data stream at the i-th position. In other words, each m-bit preamble is inserted at every i-th data bit, where $i = \lfloor 8190m/C \rfloor$ is the lowest integer close to $8190m/C$. Following the notations in [1], during the branch metric calculation at each k-th stage, we check the condition in (A-7) to adjust the branch metrics $\rho_k(p, q)$.

Specifically, at the k-th stage, if $\hat{b}(p, q) \neq f_i$, where $\hat{b}(p, q) \in \{\pm 1\}$ is the data bit corresponds to the state transition from state p to state q, and f_i is the preamble bit at the i-th position, we then set $\rho_k(p,q) = \Delta$, where Δ is a large number to guarantee that the PSP-SOVA-M will not choose this branch as part of a survivor path.

It should be noted that the M parameter must be carefully chosen according to the channel model used. For the PR2 channel with 4 trellis states, we found that $M = 3$ is a suitable value for employing in the MPS-ITR. Additionally, the results reported in this paper are still valid for any PR channel, given that the M parameter is chosen suitably.

4. COMPLEXITY COMPARISONS

To measure the complexity of iterative timing recovery schemes, we consider the total number of additions and multiplications as a criterion used in each scheme. For other mathematical functions, such as $\log(x)$, $\exp(x)$, and etc., we assume they can be implemented as lookup tables, and that we ignore their complexity. Note that we attempt to fairly count the number of operations (both addition and multiplication) for each scheme such that the memory requirement is minimized.

It can be shown that the complexity of each component is given in Table 1, where Nsinc is the total number of ideal sinc interpolation filter taps used to sample the analog signal and to refine the samples at each iteration [20] based on a set of the previous samples and their corresponding sampling phase offsets; $Q = 2^v$ is the number of trellis states [19]; δ is the decoding depth used to output the soft decision

Table 1: *The total number of operations (per bit) of each function.*

MODULE	Number of operations (**per bit**)	
	Addition	Multiplication
Ideal sinc interpolation filter	$(4N_{\text{sinc}} - 1)Q$	$(N_{\text{sinc}})Q$
1st-order PLL	Q	Q
SOVA	$7Q + \frac{\delta^2 + 9\delta + 9}{2} + 1$	$6Q + 1$
Modified PSP-SOVA-M	$(7 + 4N_{\text{sinc}})Q + \frac{\delta^2 + 9\delta + 9}{2} + 1$	$(7 + N_{\text{sinc}})Q + 1$
LDPC decoder	$(1 + (k-1)(1-R))N_{\text{in}} + 1$	$(1-R)N_{\text{in}}$

Table 2: *Complexity (per bit) of different iterative timing recovery schemes.*

SYSTEM	Number of operation (**per bit**)	
	Addition	Multiplication
Convention receiver	$27 + 223.94N$	$9 + 25.56N$
Per-survivor iterative:		
-with PSP-SOVA	569.94	$113.56N$
Modified per-survivor iterative:		
-with modified PSP-SOVA	$569.94N$	$113.56N$
-with modified PSP-SOVA-M1	$478.94N$	$85.56N$
-with modified PSP-SOVA-M2	$387.94N$	$57.56N$

in SOVA [6]; k is a parameter of a low-density parity-check (LDPC) code [21]; N_{in} is the internal iterations used in the LDPC decoder; and R is a code rate.

In this paper, we consider the proposed scheme with $M = 3$ (referred to as MPS-ITR-M1) and $M = 2$ (referred to as MPS-ITR-M2). Based on Table 1, we can summarize the complexity of each iterative timing recovery schemes as given in Table 2, where we employ N_{sinc}=21, v=2, δ=5(v+1) [20], and N_{in}=5, and N is the number of turbo iterations. It should be pointed out that multiplication has much more complexity than addition in terms of circuit implementation. Thus, we consider only the number of multiplications when comparing the performance of different iterative timing recovery schemes.

5. SIMMULATION RESULTS

Consider a rate-8/9 system in which a block of 3640 message bits is encoded by a regular (3, 27) LDPC code [21], resulting in a coded block length of 4095 bits. The parity-check matrix has 3 ones in each column and 27 ones in each row. The SISO equalizer is implemented based on SOVA, whereas the SISO decoder is implemented based on the message-passing algorithm with 5 internal iterations ($N_{in} = 5$). Note that one data sector consists of 256-bit preamble and 4095 coded bits. Each bit-error rate (BER) was computed by using as many data sectors as needed to collect 1000 error bits at the 5-th turbo iteration.

During an acquisition mode, the PLL gain parameters (α's) for the conventional receiver and PS-ITR were designed to recover the phase change within 256 symbols (according to its preamble), whereas those for MPS-ITR, MPS-ITR-M1, and MPS-ITR-M2 were designed to recover the phase change within 128 symbols because the preamble was divided into two parts (according to the PSP-SOVA-M algorithm as explained in Section 3). Note that the α's for all schemes were designed based on a linearized model of PLL [7], assuming that the S-curve slope is one at the origin, and there is no noise in the system. Furthermore, we consider the case where the α designed to recover the phase change within 256 symbols is used for all schemes during a tracking mode.

Fig. 5 compares the BER performance of different iterative timing recovery schemes at the 5-th iteration for the system with a moderate random walk parameter $\sigma_w/T = 0.6\%$ (which implies a low probability of occurrence of cycle slips) and $\sigma_w/T = 1.2\%$ (which implies a high probability of occurrence of cycle slips) as a function of per-bit SNRs, E_b/N_0's. Note that the number inside the parenthesis in Fig. 5 indicates the total number of iterations used to generate each curve. Apparently, the MPS-ITR-M1 performs better than the MPS-ITR-M2, the PS-ITR, and the conventional receiver, especially when σ_w/T is large. Furthermore, it is evident that for given the number of iterations, the MPS-ITR provides better performance than the other schemes because the MPS-ITR can reduce the occurrence of cycle slips and can also automatically correct a cycle slip much more efficiently than the PS-ITR [1]. Nevertheless, we will show later that the MPS-ITR-M1 scheme can perform better than the MPS-ITR scheme when operating at low-to-moderate complexity.

Fig. 6 compares the total number of multiplications for each iterative timing recovery scheme. Clearly, the MPS-ITR scheme has very high complexity if compared with other schemes. Additionally, we assume that the current technology can support the total number of multiplications equal to 3 iterations of the MPS-ITR scheme, which is approximately equal to 4, 6, and 13 iterations of the MPS-ITR-M1, the MPS-ITR-M2, and the conventional receiver, respectively.

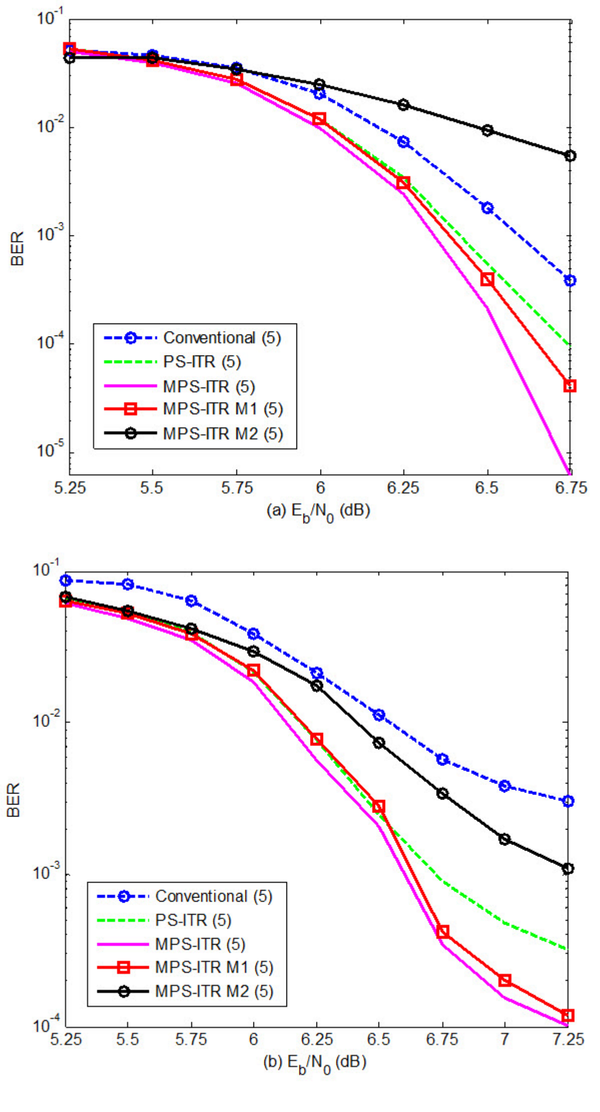

Fig.5: *Performance comparison at the 5-th iteration when (a) σ_w/T = 0.6% and (b) σ_w/T = 1.2%.*

Therefore, it is worth comparing their performance when they all have same complexity.

Fig. 7 compares the BER performance of different iterative timing recovery schemes when they have same complexity at $\sigma_w/T = 1.2\%$. It is apparent that the MPS-ITR-M1 performs better than other schemes. As a result, it is worth employing the MPS-ITR-M1 in the system when the complexity is limited to a low-to-moderate amount.

6. CONCLUSION

We proposed a reduced-complexity modified per-survivor iterative timing recovery scheme to jointly perform timing recovery, equalization, and error-correction decoding. This scheme is obtained by applying the M-algorithm to the modified PSP-SOVA to make it more implementable in real-life applications. In addition, we found that the choice of M's is crucial to the overall

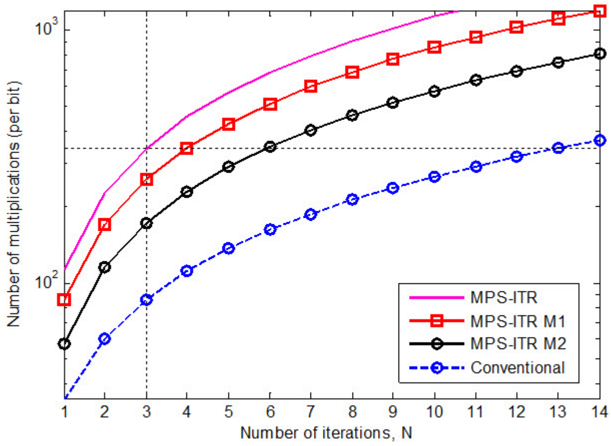

Fig.6: *Complexity comparison (based on the PR2 channel).*

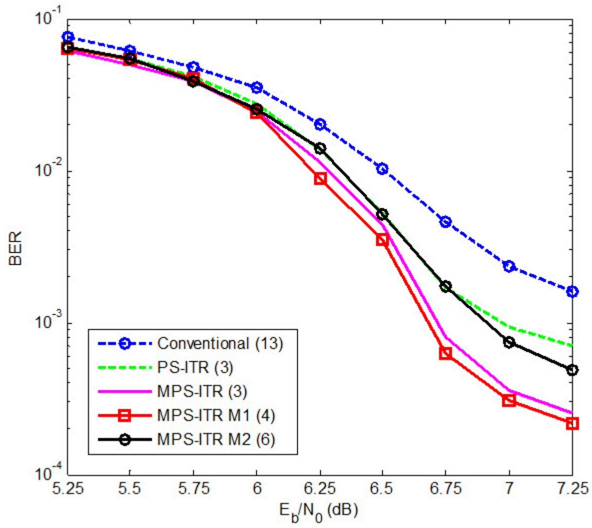

Fig.7: *Performance comparison with same complexity at σ_w/T = 1.2%.*

system performance. Specifically, the M parameter mainly depends on the channel used. For the PR2 channel, $M = 3$ is a good choice for our proposed scheme. Simulation results show that at low-to-moderate complexity, the reduced-complexity modified per-survivor iterative timing recovery scheme (with $M = 3$) performs better than other iterative timing recovery schemes and the conventional receiver.

7. ACKNOWLEDGEMENT

This work was supported by College of Data Storage Innovation, King Mongkut's Institute of Technology Ladkrabang, Thailand.

References

[1] C. Warisarn, P. Kovintavewat, and P. Supnithi, "Iterative Timing Recovery with the Split-Preamble Strategy for Coded Partial Response Channels," *IEICE Trans. Elec.*, vol. E94-C, no. 3, Mar. 2011. pp. 368–374.

[2] J. B. Simmons and S. Mohan, "Sequential coding algorithms: A survey and cost analysis," *IEEE Trans. Commun.*, vol. COM-32, pp. 169–176, Feb. 1984.

[3] T. Souvignier, A. Friedmann, M. Oberg, P. Siegel, R. Swanson, and J. Wolf, "Turbo decoding for PR4: parallel vs. serial concatenation," in *Proc. of ICC'99*, vol. 3, pp. 1638–1642, 1999.

[4] P. Kovintavewat, J. R. Barry, M. F. Eeden, and E. Kurtas, "Reduced-complexity per-survivor iterative timing recovery for coded partial response channels," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005)*, Philadelphia, USA, vol. 3, pp. iii/841–iii/844, Mar. 2005.

[5] R. Raheli, A. Polydoros, and C. K. Tzou, "The principle of per-survivor processing: a general approach to approximate and adaptive MLSE," in *Proc. of Globecom'91*, vol. 2, pp. 1170–1175, Dec. 1991.

[6] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. of Globecom'89*, pp. 1680–1686, Nov. 1989.

[7] J. W. M. Bergmans, *Digital baseband transmission and recording.* Boston, Massachusetts: Kluwer Academic Publishers, 1996.

[8] N. Seshadri and J. B. Anderson, "Decoding of severely filtered modulation codes using the (M,L) algorithm," *IEEE J. Selected Areas Commun.*, vol. 7, pp. 1006–1016, Aug. 1989.

[9] M. Iki, M. Takanashi, and H.Ochiai, "Peak Power Reduction of Single-Carrier Signals Using Trellis Shaping with M and Stack Algorithms," *Proc. of the 70th IEEE Vehicular Techology Conference*, pp. 1–5, 2009.

[10] L. Joohyun and L. Jaejin, "A Simplified Noise-Predictive Partial Response Maximum Likelihood Detection Using M-Algorithm for Perpendicular Magnetic Recording Channels," *IEEE Trans. Magn.*, vol.41, no.2, pp.1064–1066, Feb. 2006.

[11] S. Tosi, M. Power, and T. Conway, "Multitrack Digital Data Storage: A Reduced Complexity Architecture," *IEEE Trans. Magn.*, vol. 43, no. 3, pp. 1101–1111, Mar. 2007.

[12] A. R. Nayak, J. R. Barry, G. Feyh, and S. W. McLaughlin, "Timing RecoveryWith Frequency Offset and Random Walk: Cram?r-Rao Bound and a Phase-Locked Loop Postprocessor," IEEE Trans. Magn., vol. 54, no. 11, pp. 2004-2013, Nov. 2006.

[13] Z. Wei, M. F. Erden, A. Kavcic, E. Kurtas, and R. 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.

[14] H. K. Thapar and A. M. Patel, "A Class of Partial Response Systems for Increasing Storage Density in Magnetic Recording," *IEEE Trans. on Magnetics*, vol. 23, no. 5, pp. 3666–3668, Sep. 1987.

[15] B. E. Moision and P. H. Siegel, "Distance-Enhancing Codes for Gigital Recording," *IEEE Trans. on Magnetics*, vol. 34, no. 1, pp. 69–74, Jan. 1998.

[16] A. N. Andrea, U. Mengali, and G. M. Vitetta, "Approximate ML decoding of coded PSK with no explicit carrier phase reference," *IEEE Trans. Commun.*, vol. 42, no. 2/3/4, pp. 1033–1039, Feb/Mar/Apr 1994.

[17] 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.

[18] C. Warisarn, P. Kovintavewat, and P. Supnithi, "Bi-directional timing recovery for perpendicular magnetic recording channels," in *Proc. of DST-CON 2009*, Bangkok, May 13–15, 2009.

[19] 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.

[20] P. Kovintavewat and J. R. Barry, *Iterative timing recovery: a per-survivor approach.* VDM Verlag Publisher, Germany, Nov. 2009.

[21] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

Chanon Warisarn received the B.Eng. (Hon.) in Electronics Engineering Technology from King Mongkut's Institute of Technology North Bangkok (KMITNB), Thailand in 2006, the Ph.D. degree in Electrical Engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Bangkok, Thailand in 2011. He currently works at the College of Data Storage Innovation, KMITL. His current research interests are in the areas of communications and signal processing for data storage systems.

Piya Kovintavewat received the B.Eng. summa cum laude from Thammasat University, Thailand (1994), the M.S. degree from Chalmers University of Technology, Sweden (1998), and the Ph.D. degree from Georgia Institute of Technology (2004), all in Electrical Engineering. He is currently at Nakhon Pathom Rajabhat University. His research interests include coding and signal processing as applied to digital data storage systems. Prior to working at NPRU, he worked as an engineer at Thai Telephone and Telecommunication company (1994-1997), and as a research assistant at National Electronics and Computer Technology Center (1999), both in Thailand. He also had work experiences with Seagate Technology, Pennsylvania, USA (summers 2001, 2002, and 2004).

Pornchai Supnithi received the B.S. degree in Electrical Engineering from University of Rochester, Rochester, New York, USA, in 1995. M.S. degree in Electrical Engineering from University of Southern California, Los Angeles, California, USA in 1997 and Ph.D. in Electrical Engineering from Georgia Institute of Technology, Atlanta, Georgia, USA in 2002. Since then, he has been with the Telecommunications Engineering Department, King Mongkut's Institute of Technology Ladkrabang (KMITL), Bangkok, Thailand. His current research interests are in the area of data storage and communication systems, atmospheric study and GPS technology.