



# DST-CON Proceeding 2010

# Per-Survivor Iterative Interpolated Timing Recovery for Magnetic Recording System

S. Nutsatarn, P. Kovintavewat, and C. Tantibundhit

**Abstract**—Per-survivor iterative timing recovery was proposed by Kovintavewat *et al.* [1], which iteratively exchanges soft information between PSP-SOVA and an error-control decoder. Because an analog voltage-controlled oscillator (VCO) used in PSP-SOVA can be expensive to implement, this paper proposes to apply the interpolated timing recovery (ITR) concept to PSP-SOVA, resulting in “PSP-SOVA-ITR,” to make it even more implementable in today’s magnetic recording read-channel chip architectures and without significant performance loss. Results indicate that for low to moderate complexity, the proposed scheme can perform better than per-survivor iterative timing recovery, and both outperform the conventional receiver with separate timing recovery and turbo equalization.

**Index Terms**—Interpolated timing recovery, Magnetic recording, Perpendicular recording, Per-survivor iterative timing recovery, Turbo equalization.

## I. INTRODUCTION

Timing recovery is the process by which a receiver ensures that the received analog signal is sampled at the correct times. Timing recovery is a critical component in any digital communications system, since sampling at the wrong times can have a devastating impact on performance.

In general, the timing recovery problem might not be hard in isolation, but a *practical* receiver must perform not only timing recovery but also detection of the transmitted message bits, which involves other tasks, including equalization and error-control decoding. In doing so, it must contend with not only uncertainty in the timing of the pulses but also with additive noise and intersymbol interference (ISI).

The large coding gains [2, 3] of iterative error-control codes (ECCs) allow reliable operation at low signal-to-noise ratio (SNR). This means that timing recovery must also function at the SNR lower than ever before. The low SNR is a desirable property because it helps reduce the cost of operation, and in magnetic recording systems for example, allows for higher

storage capacity. Therefore, a conventional receiver, which performs timing recovery and error-control decoding *separately*, normally fails to work properly at low SNR. To solve this problem, Kovintavewat *et al.* [1] proposed per-survivor iterative timing recovery to *jointly* perform timing recovery, equalization, and error-control decoding. This scheme is realized by first developing a per-survivor soft-output Viterbi algorithm (SOVA) [4] equalizer, denoted as “PSP-SOVA,” by embedding the timing recovery step inside the SOVA equalizer using per-survivor processing (PSP) [5], a technique of jointly estimating data sequence and unknown parameters. Hence, PSP-SOVA iteratively exchanges soft information with a soft-output (SISO) decoder.

Practically, the timing recovery step used in PSP-SOVA is based on a 2nd-order phase-locked loop (PLL) [6], consisting of a timing error detector (TED), a loop filter (LF), and a voltage controlled oscillator (VCO), as illustrated in part in Fig. 2. In general, an analog VCO produces the output signal in analog domain to control the sampling clock, which can be expensive to implement. To get rid of this VCO, many methods have been proposed in the literature [7, 8, 9]. To replace the VCO with a fully digital circuit, we need i) a fixed sampling clock to sample the received signal *asynchronously*; ii) a digital accumulator; iii) an interpolation control unit to find the sampling location index; and iv) an interpolation filter to resample the data so as to obtain a synchronized sample. Thus, this scheme is known as *interpolated timing recovery* (ITR) [8].

This paper proposes to use this ITR architecture, instead of conventional timing recovery, inside PSP-SOVA [1], resulting in “PSP-SOVA-ITR,” to make it even more implementable in today’s magnetic recording read-channel chip architectures and without significant performance loss.

This paper is organized as follows. After describing the channel model in Section II, Section III explains how PSP-SOVA-ITR works. Complexity comparison is provided in Section IV, Simulation results and discussion are given in Section V. Eventually, Section VI concludes this paper.

## II. CHANNEL MODEL

Consider a rate-8/9 coded perpendicular recording channel in Fig. 1. A block of 3640 message bits,  $\{x_k\}$ , is encoded by a regular  $(j, k) = (3, 27)$  low-density parity-check (LDPC) code [10] and is mapped to a sequence  $a_k \in \{\pm 1\}$  with bit period  $T$ . The parity-check matrix has 3 ones in each column and 27 ones in each row. Then, the sequence  $a_k$  is filtered by an ideal

Manuscript received February 28, 2010.

S. Nutsatarn is with the Electrical and Computer Engineering Department, Thammasat University, Pathum Thani 12121, Thailand. (corresponding author to provide e-mail: 5110030185@student.tu.ac.th).

P. Kovintavewat is with Data Storage Technology Research Unit, Faculty of Science and Technology, Nakhon Pathom Rajabhat University, Nakhon Pathom 73000, Thailand. (e-mail: piya@npru.ac.th).

C. Tantibundhit is with the Electrical and Computer Engineering Department, Thammasat University, Pathum Thani 12121, Thailand. (e-mail: tchartur@engr.tu.ac.th).

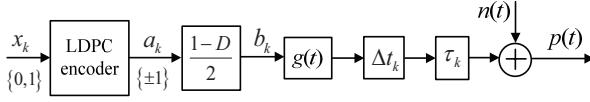


Fig. 1 Data encoding with a perpendicular recording model.

differentiator  $(1 - D)/2$ , to obtain a transition sequence  $b_k \in \{-1, 0, 1\}$ , where  $D$  is a delay operator,  $b_k = \pm 1$  corresponds to a positive or a negative transition, and  $b_k = 0$  corresponds to the absence of a transition. The transition sequence  $b_k$  passes through the magnetic recording channel represented by  $g(t)$ . The transition response  $g(t)$  for perpendicular recording is [11]

$$g(t) = \operatorname{erf}\left(\frac{2t\sqrt{\ln 2}}{\text{PW}_{50}}\right) \quad (1)$$

where  $\operatorname{erf}(x) = \left(2/\sqrt{\pi}\right) \int_0^x e^{-z^2} dz$  is an error function, and  $\text{PW}_{50}$  determines the width of the derivative of  $g(t)$  at half its maximum. In the context of magnetic recording, a normalized recording density is defined as  $\text{ND} = \text{PW}_{50}/T$ , which determines how many data bits can be packed within  $\text{PW}_{50}$ .

The readback signal,  $p(t)$ , can then be expressed as [12]

$$p(t) = \sum_k a_k \{g(t - kT - \Delta t_k - \tau_k) + g(t - (k+1)T - \Delta t_{k+1} - \tau_{k+1}) + n(t)\} \quad (2)$$

where  $n(t)$  is an additive white Gaussian noise (AWGN) with two-sided power spectral density  $N_0/2$ . The media jitter noise,  $\Delta t_k$ , is modeled as a random shift in the *transition position* with a Gaussian probability distribution function with zero mean and variance  $|b_k|\sigma_j^2$  truncated to  $T/2$ , where  $|x|$  takes the absolute value of  $x$ . The clock jitter noise,  $\tau_k$ , defined as the difference between the actual and expected arrival time of the  $k$ -th pulse, is modeled as a random walk [13] according to  $\tau_{k+1} = \tau_k + \mathcal{N}(0, \sigma_w^2)$ , where  $\sigma_w$  determines 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 signal  $p(t)$  is filtered by a seventh-order Butterworth low-pass filter (LPF), whose cutoff frequency is at  $1/(2T)$ , and is sampled at time  $kT + \hat{\tau}_k$ , where  $\hat{\tau}_k$  is the receiver's estimate of  $\tau_k$ . The sampler output  $s_k$  is equalized by an equalizer  $F_1(D)$ , such that an equalizer output  $y_k$  closely resembles a desired sample. Note that the design of a target  $H(D) = \sum_{i=0}^{\nu} h_i D^i$ , where  $\nu$  is the target memory, and its corresponding equalizer can be found in [14].

Conventional timing recovery uses a decision-directed TED [6] is used to compute the receiver's estimate of the timing error  $\varepsilon_k = \tau_k - \hat{\tau}_k$  based on the Mueller and Müller (M&M) TED algorithm [15] according to  $\hat{\varepsilon}_k = \{y_k \hat{r}_{k-1} - y_{k-1} \hat{r}_k\}$ , where  $\hat{r}_k$  is an estimate of  $r_k$  produced by the Viterbi detector (VD)

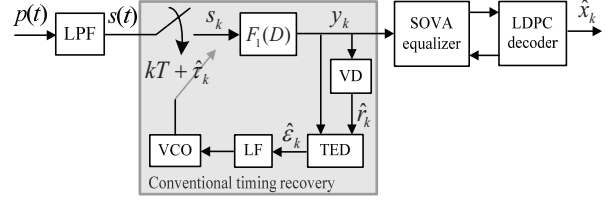


Fig. 2. A conventional receiver with separate timing recovery and turbo equalization.

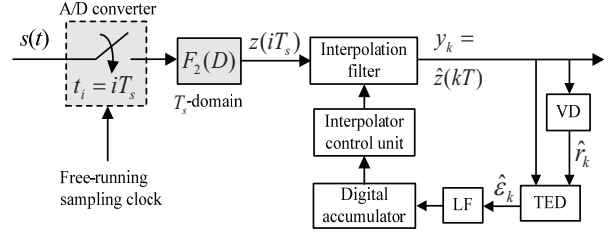


Fig. 3. Interpolated timing recovery architecture.

[16] with a decision delay  $d = 4T$ . The next sampling phase offset is updated by a second-order PLL by [6]

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \beta \hat{\varepsilon}_k \quad (3)$$

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \alpha \hat{\varepsilon}_k + \hat{\theta}_{k+1} \quad (4)$$

where  $\hat{\theta}_k$  represents an estimate of frequency error, and  $\alpha$  and  $\beta$  are the PLL gain parameters.

In the conventional receiver (see Fig. 2), conventional timing recovery is followed by a turbo equalizer [17], which iteratively exchanges soft information between the SOVA equalizer and the LDPC decoder implemented based on the message passing algorithm [10] with  $N_{\text{in}} = 5$  internal iterations.

### III. PSP-SOVA-ITR ALGORITHM

Per-survivor iterative timing recovery based on SOVA has been proposed in [1]. It is realized by first applying the PSP concept to SOVA, resulting in ‘‘PSP-SOVA.’’ In other words, PSP-SOVA combines the conventional timing recovery block and the SOVA equalizer (see Fig. 2) into one block called PSP-SOVA so as to perform timing recovery and maximum-likelihood equalization *jointly*. Therefore, per-survivor iterative timing recovery iteratively exchanges soft information between PSP-SOVA and the LDPC decoder.

PSP-SOVA-ITR works in a same manner as PSP-SOVA does, except that the conventional timing recovery block in Fig. 2 is replaced by the ITR block shown in Fig. 3, where the  $T_s$ -spaced equalizer,  $F_2(D)$ , can be placed outside the timing loop, where  $T_s$  is a sampling period. The design of  $F_2(D)$  corresponding to the target  $H(D)$  can be found in [18]. This ITR samples the signal  $s(t)$  asynchronously by an A/D converter operating at a fixed sampling frequency  $1/T_s$  to obtain a  $T_s$ -spaced sequence,  $s(iT_s)$ , which will then be equalized by an equalizer  $F_2(D)$ . Therefore, the interpolation filter uses a set of

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

Module	Number of operations (per bit)	
	Addition	Multiplication
Ideal sinc interpolation filter	$4N_{\text{sinc}} - 1$	$N_{\text{sinc}}$
Equalizer	$N_{\text{eq}} - 1$	$N_{\text{eq}}$
2nd-order PLL	4	4
Viterbi detector	$5Q + d + 2$	$2Q$
SOVA	$7Q + \frac{\delta^2 + 9\delta + 9}{2} + 1$	$6Q + 1$
PSP-SOVA	$(9 + 4N_{\text{sinc}} + N_{\text{eq}})Q + \frac{\delta^2 + 9\delta + 9}{2} + 1$	$(10 + N_{\text{sinc}} + N_{\text{eq}})Q + 1$
PSP-SOVA-ITR	$(13 + 4N_{\text{int}})Q + \frac{\delta^2 + 9\delta + 9}{2} + 1$	$(12 + N_{\text{int}})Q + 1$
LDPC decoder	$(1 + (k-1)(1-R))N_{\text{in}} + 1$	$(1-R)N_{\text{in}}$

asynchronous samples  $\{z(iT_s)\}$  and the timing information obtained from the interpolator control unit [8] to output the synchronized samples,  $y_k$ s (in  $T$ -domain).

#### IV. COMPLEXITY COMPARISON

To measure the complexity of iterative timing recovery schemes, we consider the total number of additions and multiplications as a criterion. For other mathematical functions, such as  $\log(x)$ ,  $\exp(x)$ , and etc., we assume they are 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  $N_{\text{sinc}}$  is the total number of ideal sinc interpolation filter taps used to sample the analog signal and to refine the samples at each iteration [12] based on a set of the previous samples and their corresponding sampling phase offsets;  $N_{\text{eq}}$  is the total number of equalizer taps;  $Q = 2^v$  is the number of trellis states [16],  $d = 4T$  is a decision delay;  $\delta$  is the decoding depth used to output the soft decision in SOVA [4];  $k$  is a parameter of an LDPC code [10];  $N_{\text{in}}$  is the internal iterations used in the LDPC decoder; and  $R$  is a code rate.

Based on Table 1, we can then summarize the complexity of each iterative timing recovery schemes, as given in Table 2, where we employ  $N_{\text{sinc}} = 21$ ,  $N_{\text{eq}} = 21$ ,  $v = 2$ ,  $\delta = 5(v+1)$  [12], and  $N_{\text{in}} = 5$ , and  $N$  is the number of turbo iterations. The interpolation filter employed in ITR is an 8-tap (i.e.,  $N_{\text{int}} = 8$ ) MMSE interpolation filter [8]. It should be pointed that multiplication has much more complexity than addition in terms of circuit implementation. Accordingly, this will make PSP-SOVA-ITR even more interesting than PSP-SOVA in terms of implementation cost.

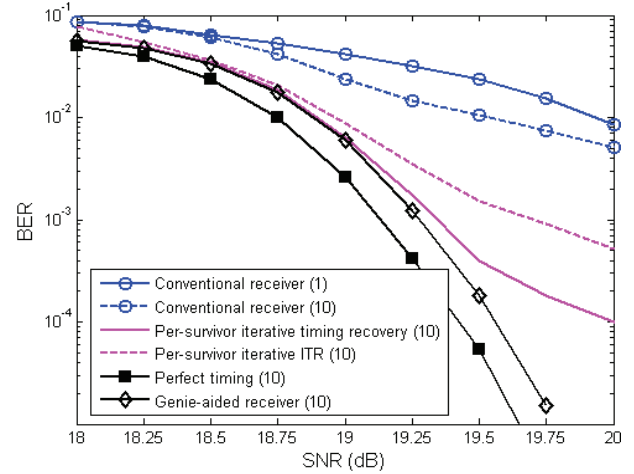
#### V. SIMULATION RESULT

Per-survivor iterative ITR is easily obtained by discarding the front-end PLL in Fig. 2 and replacing the SOVA equalizer with PSP-SOVA-ITR.

Consider a perpendicular recording channel with  $ND = 2$ ,  $\sigma_j/T = 3\%$  media jitter noise,  $\sigma_v/T = 0.5\%$  clock jitter noise, and

**Table 2:** Complexity (per bit) of different iterative timing recovery schemes, where  $N$  is the number of turbo iterations.

System	Number of operations (per bit)	
	Addition	Multiplication
Conventional receiver	$50 + 243.94N$	$33 + 25.56N$
System with perfect timing	$20 + 243.94N$	$21 + 25.56N$
Per-survivor iterative:		
- with PSP-SOVA	$671.94N$	$209.56N$
- with PSP-SOVA-ITR	$83 + 395.94N$	$21 + 81.56N$

**Fig. 4.** Performance comparison of different schemes (at the 10-th iteration).

0.2% frequency offset. The 3-tap target and a 21-tap equalizer were designed at the SNR required to achieve a bit-error rate (BER) of  $10^{-5}$ . The 3-tap target is  $H(D) = 1 + 1.144D + 0.462D^2$ . The SNR is defined as  $\text{SNR} = 10\log_{10}(E_i/N_0)$  in decibel (dB), where  $E_i$  is the energy of the channel impulse response (the derivative of the transition response scaled by 2). The PLL gain parameters were designed to recover phase/ frequency changes within 100 bit periods during acquisition mode and 256 bit periods during tracking mode, based on a linearized model of PLL [6]. Note that a 100-bit preamble will be inserted to a sequence  $a_k$  before passing it through the channel. At the receiver, after performing timing recovery, the preamble will be discarded at the equalizer output before feeding the resulting sequence to the turbo equalizer.

To account for a coded system, we define a *user density*,  $D_u$ , as  $D_u = ND/R$ , where  $R = 8/9$  is an LDPC code rate. In addition, we assume that there is no frequency offset left in the system after the first iteration. This means that a 1st-order PLL (obtained by setting  $\beta = 0$  in (3)) will be used after the first iteration. Each BER point was computed many data packets as needed until at least 100 packets in error were collected at the 5-th iteration.

Fig. 4 compares the performance of iterative timing recovery schemes for a perpendicular recording channel at  $D_u = 2$  and 10-th iteration, where the number inside the parenthesis indicates the number of iterations used to generate each curve. The curve labeled ‘‘Perfect timing’’ represents the conventional receiver that uses  $\hat{\tau}_k = \tau_k$  to sample  $s(t)$ , whereas the curve

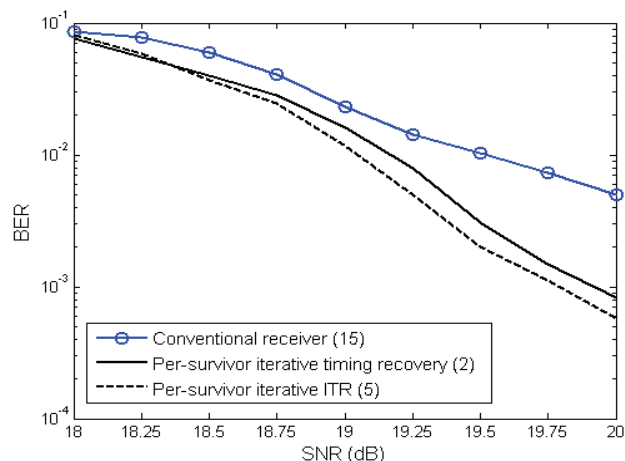


Fig. 5. Performance comparison of different iterative timing recovery schemes with same complexity.

labeled “Genie-aided receiver” represents the conventional receiver whose PLL has access to all correct decisions, thus serving as a lower bound for a timing recovery scheme that is based on PLL. Clearly, per-survivor iterative timing recovery performs better than per-survivor iterative ITR, and both outperforms the conventional receiver. Note that there is a big performance gap between per-survivor iterative timing recovery and the system with perfect timing (and the genie-aided receiver) at SNR larger than 19.50 dB. We observed that this problem is caused by a *severe cycle slip* [1, 12].

The plot in Fig. 4 shows the performance of each iterative timing recovery scheme at the 10-th iteration, regardless of complexity. However, it is worth comparing their performances when they have same complexity. To do so, we first ignore the complexity caused by the number of additions. Then, we assume that current technology can support the total number of multiplications equal to 2 iterations of per-survivor iterative timing recover. As illustrated in Table 2, it can be shown that 2 iterations of per-survivor iterative timing recovery are equal to 5 iterations of per-survivor iterative ITR, and 15 iterations of the conventional receiver.

Fig. 5 compares the performance of different schemes with same complexity. It is now apparent that per-survivor iterative ITR performs better than per-survivor iterative timing recovery and the conventional receiver. Moreover, it should be pointed out that we did not count the complexity of an analog VCO used in PSP-SOVA, which might be hard to implement. As a result, per-survivor iterative ITR will be even more attractive for applications with strict complexity constraints.

## VI. CONCLUSION

This paper proposed a reduced-complexity version of per-survivor iterative timing recovery by replacing PSP-SOVA with PSP-SOVA-ITR, resulting in per-survivor iterative ITR, so as to make it more implementable in today’s read-channel chip. Simulation results indicate that when the complexity is limited to a low-to-moderate amount, per-survivor iterative

ITR performs better than both per-survivor iterative timing recovery and the conventional receiver. Therefore, per-survivor iterative ITR is more attractive for applications with strict complexity constraints.

## ACKNOWLEDGMENT

This work was supported by I/UCRC in Data Storage Technology and Application Research Center (D\*STAR), King Mongkut’s Institute of Technology Ladkrabang, and National Electronics and Computer Technology Center (NECTEC), NSTDA, Thailand, under grant HDD-08-51-014M.

## REFERENCES

- [1] P. Kovintavewat, J. R. Barry, F. M. Erden, and E. M. Kurtas, “Reduced-complexity per-survivor iterative timing recovery for coded partial response channels,” in *Proc. of ICASSP 2005*, Philadelphia, USA, vol. 3, pp. iii/841 – iii/844, Mar 19 – 23, 2005.
- [2] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo codes,” in *Proc. of ICC’93*, vol. 2, pp. 1064 – 1070, May 1993.
- [3] S. B. Wicker, *Error control systems for digital communication and storage*. New Jersey: Prentice Hall International, 1995.
- [4] J. Hagenauer and P. Hoehner, “A viterbi algorithm with soft-decision outputs and its applications,” in *Proc. of Globecom’89*, pp. 1680 – 1686, Nov 1989.
- [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. W. M. Bergmans, *Digital baseband transmission and recording*. Boston, Massachusetts: Kluwer Academic Publishers, 1996.
- [7] F. M. Gardner, “Interpolation in digital modems – Part I: Fundamentals,” *IEEE Trans. Comm.*, vol. 41, no. 3, pp. 501 – 507, Mar 1993.
- [8] Z. N. Wu, J. M. Cioffi, and K. D. Fisher, “A MMSE interpolated timing recovery scheme for the magnetic recording channel,” in *Proc. of ICC’97*, vol. 3, pp. 1625 – 1629, 1997.
- [9] M. Spurbeck and R. T. Behrens, “Interpolated timing recovery for hard disk drive read channels,” in *Proc. of ICC’97*, vol. 3, pp. 1618 – 1624, 1997.
- [10] R. Gallager, “Low-density parity-check codes,” *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21 – 28, January 1962.
- [11] T. A. Roscamp, E. D. Boerner and G. J. Parker, “Three-dimensional modeling of perpendicular recording with soft underlayer,” *J. of Applied Physics*, vol. 91, no. 10, May 2002.
- [12] P. Kovintavewat and J. R. Barry, *Iterative timing recovery: a per-survivor approach*. VDM Verlag Publisher, Germany, Nov 2009.
- [13] A. N. Andrea, U. Mengali, G. M. Vitetta, “Approximate ML decoding of coded (PSK) with no explicit carrier phase reference,” *IEEE Trans. Commun.*, vol. 42, pp. 1033 – 1039, Feb/Mar/Apr 1994.
- [14] J. Moon and W. Zeng, “Equalization for maximum likelihood detector,” *IEEE Trans. Magn.*, vol. 31, no. 2, pp. 1083 – 1088, Mar 1995.
- [15] K. Mueller and M. Müller, “Timing recovery in digital synchronous data receivers,” *IEEE Trans. Comm.*, 24:516–531, May 1976.
- [16] 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.
- [17] T. Souvignier, A. Friedmann, M. Oberg, P. Siegel, R. Swanson, and J. K. Wolf, “Turbo decoding for PR4: parallel vs. serial concatenation,” in *Proc. of ICC’99*, vol. 3, pp. 1638 – 1642, 1999.
- [18] P. Kovintavewat, F. M. Erden, E. M. Kurtas, and J. R. Barry, “Employing Fractionally-Spaced Equalizers for Magnetic Recording Channels,” *IEEE Joint North American Perpendicular Magnetic Recording Conference (NAPMRC2003)*, Monterey, California, USA, pp. 43, January 6-8, 2003.