

Research Article

Raised Cosine Interpolator Filter for Digital Magnetic Recording Channel

Hui-Feng Tsai¹ and Zang-Hao Jiang²

¹Department of Computer Science and Information Engineering, Ching Yun University, Jhongli City 32097, Taiwan

²Optoelectronics and Systems Laboratories, Industrial Technology Research Institute, Hsinchu 31040, Taiwan

Correspondence should be addressed to Hui-Feng Tsai, hftsai@cyu.edu.tw

Received 29 September 2010; Accepted 6 February 2011

Academic Editor: Ricardo Merched

Copyright © 2011 H.-F. Tsai and Z.-H. Jiang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Interpolators have found widespread applications in communication systems such as multimedia. In this paper, the interpolated timing recovery employing raised cosine pulse for digital magnetic recording channel is investigated. This study indicates that the raised cosine interpolator with rolloff factor β between 0.4 and 0.6 is shown to have less aliasing effect and achieve better MSE performance than other interpolators such as the sinc, polynomial, and MMSE interpolators with similar computational complexity. The superiority of the raised cosine interpolator over other interpolators is also demonstrated on the ME2PRIV recording channel through computer simulations. The main advantage of the raised cosine interpolator is that it is potentially simpler and can be fully digitally implemented.

1. Introduction

The digital filter applications to continuous-time and discrete-time signals are possible because of the sampling theorem. The sampling frequency might change from one value to another employing a conversion referred to as interpolators and decimators. These subsystems are applied in communication systems applications such as multimedia. The sampling theorem states that a continuous signal can be perfectly recovered using an ideal lowpass filter provided that the sampling rate is above the Nyquist rate for a bandlimited channel; that is, the interpolation filter design can be normally based on a sinc pulse [1]. However, the infinite-length sinc interpolator is impossible to implement from the application perspective. A truncated sinc interpolator always results in severe errors.

Instead of the sinc interpolator, Erup et al. [2] suggested employing polynomial interpolators (such as linear, parabolic, and cubic polynomials) to obtain the synchronized samples from the A/D converter outputs. The polynomial interpolator is simple but is only suitable for high sampling rates. Kim et al. [3] suggested designing the interpolator by minimizing the mean square error (MMSE)

that takes into account the background noise. The MMSE interpolator is an optimal interpolator in the sense that it minimizes the timing error, but it always suffers from computational complexity. Based upon a finite-state Markov model, Zeng et al. [4] proposed a Baum-Welch algorithm for digital timing recovery.

In addition to the sinc pulse, there is an interpolation pulse, called the raised cosine pulse that also satisfies the first Nyquist criterion and can be applied to the design of the interpolation filter. The truncated raised cosine interpolator for timing recovery has been proposed in [5] and shown its superiority over other interpolators such as the sinc, polynomial, spline, and MMSE interpolators. In this paper, an interpolated timing recovery method that uses the raised cosine pulse for digital magnetic recording channel is investigated. Simulation results indicate that the raised cosine interpolator achieve the best performance in both MSE and error performance than other interpolators such as the sinc, polynomial, and MMSE interpolators with similar computational complexity.

The interpolated timing recovery scheme is depicted in Figure 1. As shown, in the partial response maximal likelihood (PRML) system [6], the digital magnetic recording

channel is shaped as a partial response channel using a PR equalizer. The maximum likelihood sequence detection (MLSD) or Viterbi detection is used to recover sampled data. The fully digital timing recovery scheme employs an interpolation filter to obtain the synchronized sampled data instead of the conventional PLL.

The rest of this paper is organized as follows. In Section 2, the truncated raised cosine interpolator and its frequency response are described. The aliasing effect due to truncation on several partial response recording channels is investigated. The mean square error (MSE) performance of the raised cosine interpolator and its computational complexity is studied and compared with other interpolators. Section 3 demonstrates the superiority of the proposed interpolated timing recovery over other interpolators through computer simulations on the ME2PRIV recording channel. Conclusions are provided in Section 4.

2. Raised Cosine Interpolator for PRML Channels

Conventional timing recovery is performed on a symbol-by-symbol basis with a phase locked loop (PLL) that employs the voltage control oscillator (VCO) to adjust the sampling phase at each sampling time. Gardner [7] first proposed a fully digital timing recovery algorithm in which an A/D converter is employed to sample the received signal at a fixed sampling rate, using an interpolation filter to recover the synchronized samples from the A/D converter outputs. A detector then operates on these interpolated samples as they would in a conventional PLL where the sampling rate is synchronized to the symbol rate of the received signal.

In an all-digital interpolated timing recovery scheme, as depicted in Figure 1, an A/D converter samples the received signal $r(t)$ using a fixed clock with period T_s . Neither its frequency nor its phase is synchronized with the received signal. An interpolation filter is then used to obtain the desired samples for detection from unsynchronized input samples. As shown, the output of the interpolation filter $y(t)$ can be expressed as

$$y(t) = \sum_m r(mT_s)h_I(t - mT_s), \quad (1)$$

where $h_I(t)$ represents the impulse response of the interpolator. The synchronized sample $y(kT)$ is obtained by sampling $y(t)$ at time $t = kT$, where T is the channel bit period and $y(kT)$ is given by

$$y(kT) = \sum_m r(mT_s)h_I(kT - mT_s). \quad (2)$$

Timing phase error was measured by the timing phase error detector and filtered in the loop filter with output that drives the calculator. As shown in Figure 2 [7], assume that the time

$t = kT$ is located between $[m_k T_s, (m_k + 1)T_s]$, $y(kT)$ can be written as

$$\begin{aligned} y(kT) &= y((m_k + \mu_k)T_s) \\ &= \sum_{n=-N_1}^{N_2} r((m_k - n)T_s)h_I((n + \mu_k)T_s), \end{aligned} \quad (3)$$

where the basepoint index m_k and the fractional interval μ_k are, respectively, given by

$$m_k = \text{int} \left\lfloor \frac{kT}{T_s} \right\rfloor, \quad \mu_k = \frac{kT}{T_s} - m_k, \quad n = m_k - m. \quad (4)$$

Theoretically, the received signal $r(t)$ can be perfectly recovered by the interpolation filter using an infinite-length sinc pulse (i.e., $h_I(t) = \sin(\pi t/T_s)/(\pi t/T_s) = \text{sinc}(t/T_s)$) if the sampling rate is above the Nyquist rate. However, it is impossible to implement to use an infinite-length sinc pulse in actual applications. The interpolation filter design has been investigated in many papers [1–4, 7–15], including using truncated sinc pulse [1] and polynomial functions (linear, parabolic, and cubic functions) [2]. Although the polynomial filters are simple, they are only suitable for high sampling rates. In addition, Kim et al. [3] proposed using MMSE (minimum mean square error) criterion to design an interpolation filter in which the background noise has been taken into account. The MMSE interpolation filter can outperform other filters, but it suffers from computational complexity.

Instead of the sinc pulse, a raised cosine pulse is proposed in previous work [5] for use as an interpolation filter. The raised cosine pulse also satisfies the first Nyquist criterion for zero intersymbol interference (ISI). The impulse response of the raised cosine filter is given by

$$h_{RC}(t) = \frac{\cos(\beta\pi t/T_s) \sin(\pi t/T_s)}{1 - 4\beta^2 t^2/T_s^2} \frac{\sin(\pi t/T_s)}{\pi t/T_s}, \quad (5)$$

with $0 \leq \beta \leq 1$. Note that for $\beta = 0$ the raised cosine pulse is exactly the sinc pulse. The frequency response $H_{RC}(w)$ of the raised cosine filter $h_{RC}(t)$ is given by

$$H_{RC}(w) = \begin{cases} T_s, & 0 \leq |wT_s| \leq \pi(1 - \beta), \\ \frac{T_s}{2} \left\{ 1 - \sin\left(\frac{|wT_s| - \pi}{2\beta}\right) \right\}, & \pi(1 - \beta) \leq |wT_s| \leq \pi(1 + \beta) \\ 0, & |wT_s| \geq \pi(1 + \beta), \end{cases} \quad (6)$$

where β is called the rolloff factor.

2.1. Frequency Response of Truncated Raised Cosine Filters. There are some commonly used windows to truncate the raised cosine interpolator such as rectangular, triangular, Blackman, Hamming, and Hanning windows. An intensive study indicates that the symmetrical rectangular window is

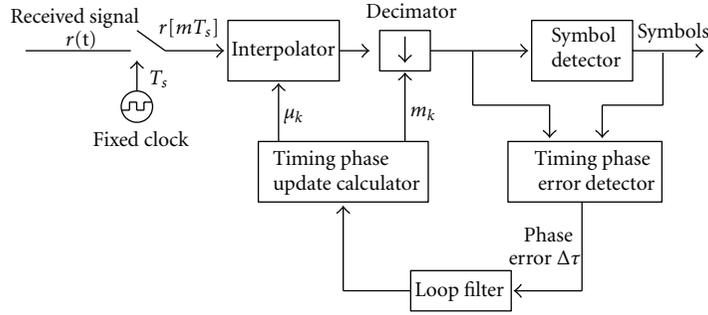
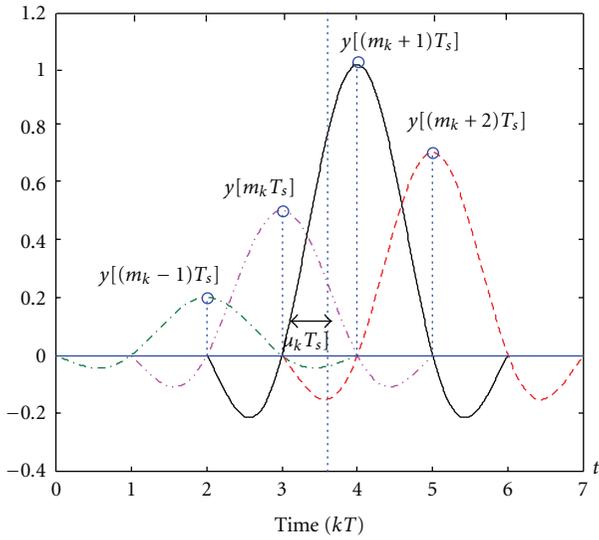


FIGURE 1: Interpolated timing recovery scheme.


 FIGURE 2: Resample $y(kT)$.

the best way to truncate the raised cosine pulse. The impulse response of the filter is given by

$$h_I(t) = h_{RC}(t)w_r(t) = \begin{cases} h_{RC}(t), & -\frac{M}{2}T_s \leq t \leq \frac{M}{2}T_s, \\ 0, & \text{otherwise,} \end{cases} \quad (7)$$

where M is an even integer. The rectangular window $w_r(t)$ is given by

$$w_r(t) = \begin{cases} 1, & -\frac{M}{2}T_s \leq t \leq \frac{M}{2}T_s, \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

It follows from the modulation or windowing property that the frequency response $H_I(w)$ of the interpolation filter using a truncated raised cosine pulse can be expressed as

$$H_I(w) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H_{RC}(\theta)W_r(w - \theta)d\theta, \quad (9)$$

with the Fourier transform of the window $W_r(w)$ given by

$$W_r(w) = \frac{2 \sin(wMT_s/2)}{w}. \quad (10)$$

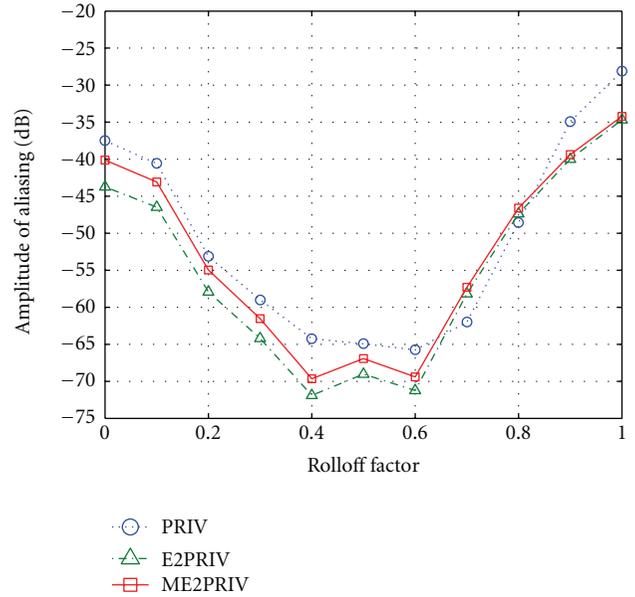


FIGURE 3: Peak amplitude of aliasing versus rolloff factor for PRML channel.

2.2. Aliasing Effect on PRML Channels. Consider that the bandlimited channel is a PRML recording channel [6] and that the received signal $r(t)$ is sampled at a rate of 2 samples/symbol (i.e., $T/T_s = 2$). Figure 3 shows the peak amplitude of the aliasing introduced in the truncated raised cosine (with truncation length $M = 12T_s$) interpolator versus the rolloff factor β for various PRML recording channels (PRIV, E2PRIV, and ME2PRIV, with response given by $1 - D^2$, $(1 - D)(1 + D)^3$ and $5 + 4D - 3D^2 - 4D^3 - 2D^4$, resp., [16]). The result indicates that the truncated raised cosine interpolator with rolloff factor β between 0.4 and 0.6 achieves good aliasing performance. The truncated raised cosine pulse with rolloff factor $\beta = 0.5$ is employed for further study.

Figure 4 displays the peak amplitude of the aliasing introduced in sinc ($\beta = 0$) and raised cosine ($\beta = 0.5$) interpolators versus the truncation length for these PRML recording channels. As shown, the raised cosine pulse outperforms the sinc pulse and the aliasing effect can be significantly reduced when the truncation length of both pulses increases. The case for the cubic pulse is also shown

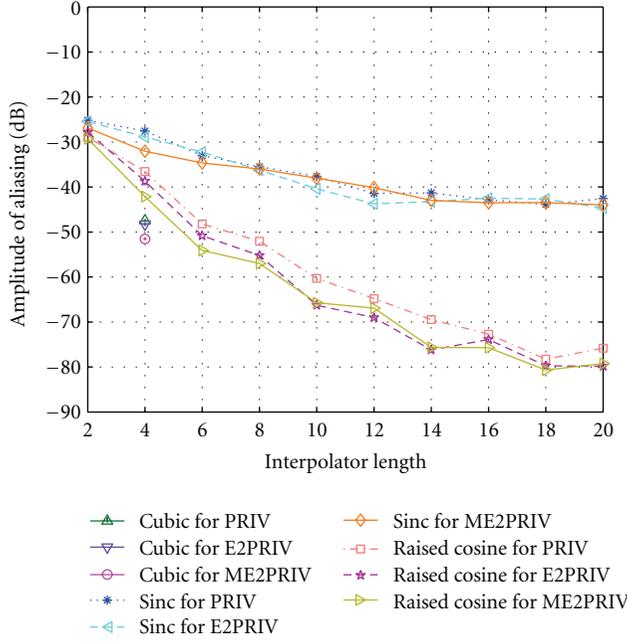


FIGURE 4: Peak amplitude of aliasing versus truncation length for PRML channel.

in the figure for comparison. The results demonstrated the superiority of the raised cosine interpolator over other interpolators of the limited number of interpolators tested.

2.3. Mean Square Error (MSE) on PRML Channels. Assume that the received signal before the A/D converter is given by

$$r(t) = \sum_{j=-\infty}^{\infty} a_j g(t - jT) + N(t), \quad (11)$$

where $\{a_j\} \in \{\pm 1\}$ represents the binary transmitted sequence and $N(t)$ is the background noise. For an ideal PRIV channel, the isolated transition response has a nonzero amplitude at sampling instants $t = 0$ and $t = T$. The NRZ bit response $g(t)$ is given by

$$g(t) = \frac{\sin(\pi t/T)}{\pi t/T} - \frac{\sin[\pi(t-2)/T]}{\pi(t-2)/T}. \quad (12)$$

The amplitude at sampling instants is $+2$, 0 , or -2 . The receiver requires obtaining $r(kT)$, but it is sampled at a period T_s (probably with a time offset μ normalized to T_s). Therefore, the resample output of the interpolation filter is given by

$$\begin{aligned} \tilde{y}(kT) &= y((m_k + \mu_k + \mu)T_s) \\ &= \sum_{n=-N_1}^{N_2} r((m_k - n)T_s) h_I((n + \mu_k + \mu)T_s) \\ &= \sum_{i=-\infty}^{\infty} a_{k-i} \tilde{G}_i + \tilde{N}_k, \end{aligned} \quad (13)$$

where

$$\begin{aligned} \tilde{G}_i &= \sum_{n=-N_1}^{N_2} h_I((n + \mu_k + \mu)T_s) \\ &\quad \times g(iT - (n + \mu_k + \mu)T_s) = G_i^T H_I, \\ \tilde{N}_k &= \sum_{n=-N_1}^{N_2} h_I((n + \mu_k + \mu)T_s) N((m_k - n)T_s) = N^T H_I. \end{aligned} \quad (14)$$

To compare the interpolation filter performance, the mean square error $\text{MSE}(\mu)$ between the ideal (synchronized) sample and the asynchronous resample is defined as

$$\begin{aligned} \text{MSE}(\mu) &= E\{[a_k - \tilde{y}(kT)]^2\} \\ &= E\left\{ \left[a_k - \sum_{i=-\infty}^{\infty} a_{k-i} G_i^T H_I + N^T H_I \right]^2 \right\} \\ &= 1 - 2G_0^T H_I + H_I^T \left(R_{NN} + \sum_{i=-\infty}^{\infty} G_i G_i^T \right) H_I, \end{aligned} \quad (15)$$

where $R_{NN} = E\{NN^T\}$ is the autocorrelation matrix of the background noise and $p\{a_k = 1\} = p\{a_k = -1\} = 1/2$.

For the MMSE interpolator [3] (i.e., $\partial \text{MSE}(\mu)/\partial H_I = 0$), the optimal impulse response H_I of the interpolation filter is given by

$$\begin{aligned} H_{I,\text{opt}} &= [h_I((-N_1 + \mu_k + \mu)T_s) h_I((-N_1 + 1 + \mu_k + \mu)T_s) \\ &\quad \cdots h_I((N_2 + \mu_k + \mu)T_s)]^T \\ &= \left(R_{NN} + \sum_{j=-\infty}^{\infty} G_j G_j^T \right)^{-1} G_0. \end{aligned} \quad (16)$$

When the sinc and raised cosine interpolators are used, the impulse responses H_I are, respectively, given by

$$\begin{aligned} H_I &= [\sin c(-N_1 + \mu_k + \mu) \sin c(-N_1 + 1 + \mu_k + \mu) \\ &\quad \cdots \sin c(N_2 + \mu_k + \mu)]^T, \\ H_I &= \left[\frac{\cos \beta(-N_1 + \mu_k + \mu)\pi}{1 - 4\beta^2(-N_1 + \mu_k + \mu)^2} \sin c(-N_1 + \mu_k + \mu) \right. \\ &\quad \left. \cdots \frac{\cos \beta(N_2 + \mu_k + \mu)\pi}{1 - 4\beta^2(N_2 + \mu_k + \mu)^2} \sin c(N_2 + \mu_k + \mu) \right]^T. \end{aligned} \quad (17)$$

As shown above, the MSE is a function of the time offset μ . The MSE performance comparison of these different interpolation filters is made under the assumptions of no noise corruption (i.e., $R_{NN} = 0$) and the PRIV recording channel. Figure 5 depicts the MSE performance of the interpolators as a function of the time offset μ at a rate

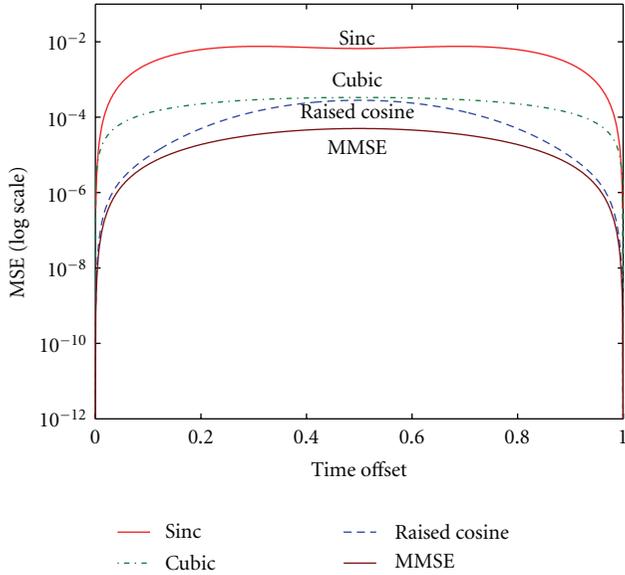


FIGURE 5: MSE versus time offset μ for $M = 4T_s$.

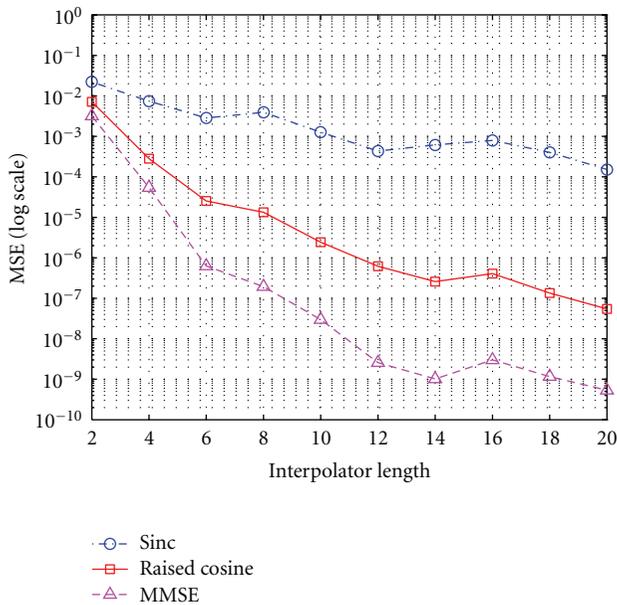


FIGURE 6: MSE versus truncation length for time offset $\mu = 0.5$.

of 2 samples/symbol for truncation length $M = 4T_s$. The MSE performance of the cubic interpolator [7] is also shown in Figure 5 for comparison. As shown in Figure 5, with the same truncation length, the interpolator using the raised cosine pulse (with $\beta = 0.5$) outperforms the cubic and sinc functions. The MMSE interpolator achieves the best performance but with much more computational complexity. Figure 6 shows the MSE performance of these interpolators as a function of the truncation length for the time offset $\mu = 0.5$. A similar MSE performance can be obtained for other partial response channels and omitted here.

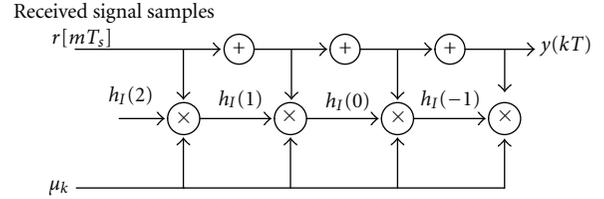


FIGURE 7: Preliminary structure for raised cosine interpolator with $M = 4T_s$.

2.4. Computational Complexity. In the interpolated timing recovery, the synchronized output signal $y(kT)$ is obtained by computing the interpolant at time $kT = (m_k + \mu_k)T_s$ in which the fractional interval μ_k is evaluated using a time phase estimator. The interpolant can be calculated using two types of FIR filters. The first FIR filter stores the impulse response h_I from the interpolation filter in a finite memory. In this type of implementation, the fractional interval μ_k is quantized into P uniform intervals, and each sample value from the fractional interval μ_k is stored in a memory that requires MP words. For each interpolation, each sample from the memory is loaded into a transversal filter as the filter coefficient. Both the sinc and raised cosine interpolators can be implemented in this kind of transversal filter. A preliminary structure for raised cosine interpolator with $M = 4T_s$ is shown in Figure 7.

For the MMSE interpolator, it is impossible to store the impulse response of the filter because its impulse response is dependent upon the noise or the fractional interval μ_k . Therefore the interpolant must be computed directly online. In this type of implementation, all computations are performed online, and no memory for the filter coefficient or quantization is required. The computational complexity is much higher than that of the sinc or raised cosine filter. For polynomial interpolators such as linear, parabolic, or cubic interpolators, the interpolation can be accomplished by direct computation with a Farrow structure [2], and the computational complexity is greatly reduced.

Tables 1(a) and 1(b) shows the computational complexities of interpolation filters that require computing an interpolant. Note that since the sinc interpolator has the same computational complexity as the raised cosine interpolator, and its complexity is not shown in the table. As displayed in this table, only $(M - 1)$ multipliers are required for the sinc or raised cosine interpolators with truncation length M . The computational complexity is much less than that of the MMSE interpolator. As can be seen from Tables 1(a) and 1(b), the raised cosine interpolator with truncation length $M = 16T_s$ still has much less computational complexity than the MMSE interpolator with $M = 4T_s$. Although the computational complexity of the MMSE interpolator can be improved by using a lookup table to store precalculated coefficients, its performance might be degraded.

TABLE 1: Computational complexities with require computing an interpolant.

Operation	Interpolator					
	Cubic	Raised cosine $4T_s$	Raised cosine $8T_s$	Raised cosine $12T_s$	Raised cosine $16T_s$	Raised cosine $20T_s$
Add/subtract	11	3	7	11	15	19
Multiply/divide	3	4	8	12	16	20

Operation	Interpolator				
	MMSE $4T_s$	MMSE $6T_s$	MMSE $8T_s$	MMSE $10T_s$	MMSE $12T_s$
Add/subtract	102	370	910	1818	3190
Multiply/divide	236	1542	5560	14690	32100

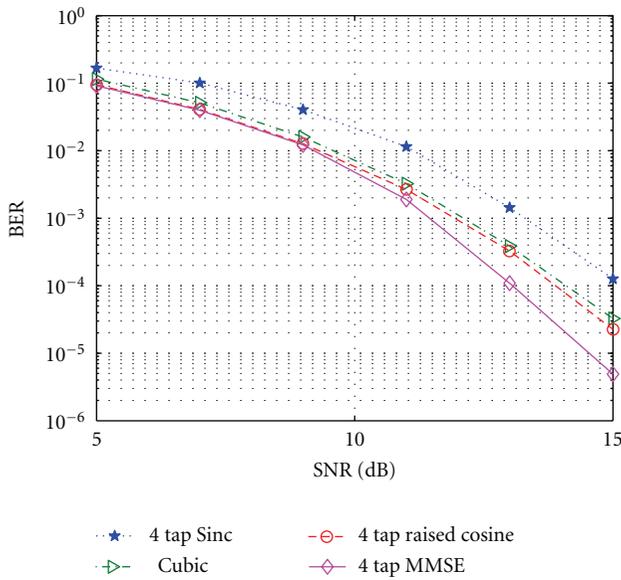


FIGURE 8: BER versus SNR for ME2PRIV channel with various interpolation filters.

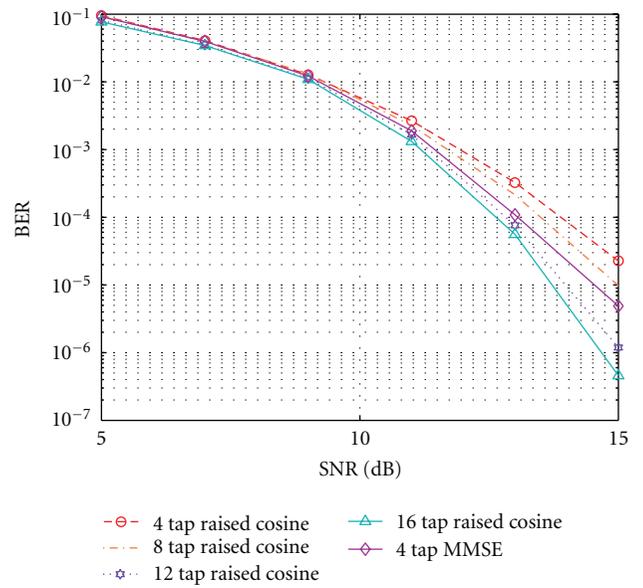


FIGURE 9: BER versus SNR for ME2PRIV channel with various truncation lengths.

3. Performance on ME2PRML Recording Channel

In this section, the error performance of the all-digital interpolated timing recovery is investigated through computer simulations on a PRML recording channel [6].

In the PRML system, the digital recording channel is shaped as a ME2PRIV partial response channel (i.e., $5 + 4D - 3D^2 - 4D^3 - 2D^4$ channel) using a ME2PRIV equalizer, and the maximum likelihood sequence detection (MLSD) or Viterbi detection is used to recover sampled data. The fully digital timing recovery scheme employs an interpolation filter to obtain the synchronized sample instead of the conventional PLL. In addition, a decision-directed phase error detector described in [17] is employed to adjust the time phase with the timing gradient

$$\Delta\tau_k = g_k^T \vec{y}_m, \quad (18)$$

where $\vec{y}_m = (y_{k-m+1}, y_{k-m+2}, \dots, y_k)$ is the signal vector at $t = \tau + kT$ containing the last m input samples, and g_k is weighting vector function of the transmitted symbols

$$g_k = \begin{pmatrix} g_1(a_{k-m+1}, \dots, a_k) \\ g_2(a_{k-m+1}, \dots, a_k) \\ \vdots \\ g_m(a_{k-m+1}, \dots, a_k) \end{pmatrix}, \quad (19)$$

where $\{a_k\} \in \{\pm 1\}$. The timing gradient $\Delta\tau_k$ is passed to a second-order loop filter to eliminate time phase jitters. The filtered timing gradient $\Delta\tau'_k$ is then used to calculate the next fractional interval μ_{k+1} by

$$\begin{aligned} \tau_{k+1} &= \tau_k + \alpha \Delta\tau'_k + \Delta T_k, \\ \mu_{k+1} &= \frac{(k+1)T}{T_s} - m_{k+1} + \frac{\tau_{k+1}}{T_s}, \end{aligned} \quad (20)$$

where ΔT_k is used to compensate for variations of the A/D converter. The next fractional interval μ_{k+1} is fed into the interpolation filter to compute the next resample data $y((k+1)T)$.

For an ideal ME2PRIV channel, the isolated transition response has a nonzero amplitude at sampling instants $t = 0$ and $t = T$, and the NRZ bit response $g(t)$ is given by

$$g(t) = \frac{5 \sin(\pi t/T)}{\pi t/T} + \frac{4 \sin[\pi(t-T)/T]}{\pi(t-T)/T} - \frac{3 \sin[\pi(t-2T)/T]}{\pi(t-2T)/T} - \frac{4 \sin[\pi(t-3T)/T]}{\pi(t-3T)/T} - \frac{2 \sin[\pi(t-4T)/T]}{\pi(t-4T)/T}. \quad (21)$$

The amplitude at sampling instants is $-2, +14, +12, -6$, or -18 . The receiver must obtain (kT) , but it is sampled at a period T_s probably with a time offset μ (normalized to T_s). The resample output of the interpolation filter is given by (13). The timing recovery control loop operates in two separate modes: the acquisition mode, and the tracking mode. In the acquisition mode the timing loop locks onto a preamble data pattern that is given by the sequence $\{\dots, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, \dots\}$. The recorded or transmitted sequence becomes $\{\dots, +1, -1, -1, -1, +1, -1, +1, +1, +1, -1, \dots\}$ after the NRZI modulation, and the ideal output amplitudes at sampling instants are $\{\dots, -2, +14, +12, -6, -18, -2, +14, +12, -6, -18, \dots\}$. To eliminate the hang-up effect, the estimate of transmitted symbol a_k in the acquisition mode is given by

$$\hat{a}_k = \begin{cases} +|\eta_k| & \text{for } y(kT) \geq \eta_k, \\ 0 & \text{for } \eta_k = 0, \\ -|\eta_k| & \text{for } y(kT) < \eta_k, \end{cases} \quad (22)$$

with the variable decision threshold η_k given by

$$\eta_k = \hat{a}_{k-1} + \hat{a}_{k-2} + \hat{a}_{k-3} + \hat{a}_{k-4}. \quad (23)$$

In the tracking mode, the transmitted symbol estimate a_k is obtained directly using a symbol-by-symbol decision

$$\hat{a}_k = \begin{cases} 18 & y(kT) > 16, \\ 14 & 13 < y(kT) < 16, \\ \pm m & \pm m - 1 < y(kT) < \pm m + 1, \\ -14 & -16 < y(kT) < -13, \\ -18 & y(kT) < -16. \end{cases} \quad (24)$$

The performance of the interpolated timing recovery using interpolators mentioned previously is evaluated on the PRML channel through computer simulations. The sampling rate is 2 samples/symbol ($T/T_s = 2$). The input noise is assumed to be the AWGN noise that is filtered by an ideal ME2PRIV equalizer for a Lorentzian channel with recording density $S = 3$ (S is defined as pw_{50}/T , where pw_{50} is

the duration of the half amplitude of the isolated transition response). During the simulations, the initial time phase was assumed to be $0.8T$, and a 140-bit preamble is used to lock the time phase in the acquisition mode.

Figure 8 compares the performance of different interpolators for the time offset $\mu = 0.5$. The truncation length for all interpolators is $4T_s$. As shown, the 4-tap MMSE interpolator outperforms other interpolators and has a $0.8\sim 2$ dB gain over the others. The raised cosine interpolator is superior in error performance to both cubic and sinc interpolators. The error performance was also simulated for raised cosine interpolators with various truncation lengths for the time offset $\mu = 0.5$, and the result is shown in Figure 9. The 16-tap raised cosine interpolator has an improvement of 1.6 dB over the 4-tap raised cosine interpolator, and it also outperforms the 4-tap MMSE interpolator, as can be seen from Figure 9. To better serve the system performance and computational complexity requirement, we proposed a 12-tap raised cosine interpolator for a PRML digital recording channel. This is shown in both Figures 6 and 9.

4. Conclusion

In this paper, the interpolated timing recovery employing raised cosine pulse for digital magnetic recording channel is presented. The raised cosine interpolator was shown to be superior to the other interpolators. The raised cosine pulse with rolloff factor β between 0.4 and 0.6 introduces less aliasing effect and achieves good MSE performance. From the simulation results presented in this paper, when the recording density is 3.0 with a ME2PRIV target, the error performance for the raised cosine pulse interpolator outperforms the other interpolators. Although the MMSE interpolator can achieve very good performance, it always suffers from excessive computational complexity. Based on the analysis in this paper, a 12-tap raised cosine interpolator is a superior choice compared to the other interpolators. Full digital implementation is possible for a raised cosine pulse used in a digital magnetic recording channel.

References

- [1] J. Armstrong and D. Strickland, "Symbol synchronization using signal samples and interpolation," *IEEE Transactions on Communications*, vol. 41, no. 2, pp. 318–321, 1993.
- [2] L. Erup, F. M. Gardner, and R. A. Harris, "Interpolation in digital modems Part II: implementation and performance," *IEEE Transactions on Communications*, vol. 41, no. 6, pp. 998–1008, 1993.
- [3] D. Kim, M. J. Narasimha, and D. C. Cox, "Design of optimal interpolation filter for symbol timing recovery," *IEEE Transactions on Communications*, vol. 45, no. 7, pp. 877–884, 1997.
- [4] W. Zeng, A. Kavcic, and R. Motwani, "Extraction of timing error parameters from readback waveforms," *IEEE Transactions on Magnetics*, vol. 42, no. 2, pp. 194–199, 2006.
- [5] H. F. Tsai, Z. H. Jiang, and Y. Y. Lin, "Interpolation filter using raised cosine pulse for timing recovery," *International Journal of Electrical Engineering*, vol. 12, no. 4, pp. 391–398, 2005.

- [6] R. D. Cideciyan, F. Dolivo, R. Hermann, W. Hirt, and W. Schott, "A PRML system for digital magnetic recording," *IEEE Journal on Selected Areas in Communications*, vol. 10, no. 1, pp. 38–56, 1992.
- [7] F. M. Gardner, "Interpolation in digital modems—part I: fundamentals," *IEEE Transactions on Communications*, vol. 41, no. 3, pp. 501–507, 1993.
- [8] Z. N. Wu, J. M. Cioffi, and K. D. Fisher, "MMSE interpolated timing recovery scheme for the magnetic recording channel," in *Proceedings of the IEEE International Conference on Communications (ICC '97)*, pp. 1625–1629, June 1997.
- [9] X. Liu and A. N. Willson Jr., "A new interpolated symbol timing recovery method," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 569–572, Los Angeles, Calif, USA, May 2004.
- [10] J. T. Kim, "Efficient implementation of polynomial interpolation filters for full digital receivers," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 1, pp. 175–178, 2005.
- [11] M. Kiviranta, "Novel interpolator structure for digital symbol synchronisation," in *Proceedings of the International Conference on Wireless Communications and Applied Computational Electromagnetics*, pp. 1014–1017, Oulu, Finland, April 2005.
- [12] K. K. Hong, S. H. Yoon, and J. W. Chong, "Design and implementation of the high speed VP (variable parameter)-cubic interpolator for the system of MB-OFDM UWB," in *Proceedings of the International Conference on Digital Telecommunications (ICDT '06)*, pp. 44–48, Cote d'Azur, France, August 2006.
- [13] X. Dong, W. S. Lu, and A. C. K. Soong, "Linear interpolation in pilot symbol assisted channel estimation for OFDM," *IEEE Transactions on Wireless Communications*, vol. 6, no. 5, pp. 1910–1920, 2007.
- [14] R. Pulikkoonattu, H. K. Subramanian, and S. Laxman, "Least square based piecewise parabolic interpolation for timing synchronization," in *Proceedings of the IEEE Radio and Wireless Symposium (RWS '08)*, pp. 155–158, Orlando, Fla, USA, January 2008.
- [15] F. J. Harris and M. Rice, "Multirate digital filters for symbol timing synchronization in software defined radios," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 12, pp. 2346–2357, 2001.
- [16] H. Sawaguchi, M. Knodou, N. Kobayahi, and S. Mita, "channels," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '98)*, pp. 2694–2699, Sydney, Australia, November 1998.
- [17] K. H. Mueller and M. Müller, "Timing recovery in digital synchronous data receivers," *IEEE Transactions on Communications*, vol. 24, no. 5, pp. 516–531, 1976.