Digital Detection with Asynchronous Sampling using Amplitude Error Prediction

Travis Oenning and Jaekyun Moon
Department of Electrical and Computer Engineering
University of Minnesota, Minneapolis, MN 55455

***Abstract*—This work concerns low complexity methods which allow sampling detectors to operate asynchronously at a rate only slightly higher than the synchronous sampling rate. There are several motivating factors for using a fixed rate sampling clock compared to conventional phase locked loop methods which use a variable rate sampling clock. We explain a few of these factors along with describing two different approaches for performing timing recovery with asynchronous samples. The first approach, which is usually used, is an interpolation based approach. The second approach is based on predicting amplitude error that is due to sampling phase error. Simulation results are presented along with discussion comparing the different methods.**

***Index Terms*—All-digital timing recovery, asynchronous sampling, interpolation, synchronization.**

I. Introduction

THE objective of this work is to find a low complexity method which allows sampling detectors to operate asynchronously at a rate only slightly higher than the synchronous sampling rate. There are several motivating factors for using asynchronous sampling methods over conventional techniques. One factor is that delay due to equalization does not contribute to the loop delay associated with conventional timing recovery techniques. Another factor is that timing recovery can be done one hundred percent digitally, thereby providing easier design and more reliable design verification.

High speed digital detection methods that operate using a free running sampling clock have only recently been appearing in the literature. The usual approach has been to first equalize asynchronous samples and then use these to interpolate to synchronous sample values. [1], [2], [3]. In this work we propose a different approach, which is based on predicting amplitude error that is due to sampling phase error.

In the following we will be describing interpolation and amplitude error predictor (AEP) methods applied to a partial response class IV (PR4) system operating with a fixed sampling clock. Keep in mind that these methods may equally well be applied to other detection schemes. The next section will start out by describing the corresponding synchronous system and then go on to describe the effects of asynchronous sampling. The end of the paper presents simulation results along with a discussion comparing the different methods.

Manuscript received October 17, 1997.

T. Oenning, 612-625-3474, toenning@ece.umn.edu; J. Moon, 612-625-7322, moon@ece.umn.edu; fax 612-626-1050, web address http://www-cdslab.ece.umn.edu/

This work was supported in part by IBM Partnership Award.

II. System Description

Fig. 1 shows the synchronous model of the PR4 system being considered. In modeling the readback waveform, additive white Gaussian noise and a Lorentzian transition response were assumed. As the figure shows, this readback waveform is low pass filtered, sampled synchronously, equalized, and then put through a maximum likelihood (ML) detector. In the figure d_k denotes the desired output value of the PR4 equalizer, $d_k = a_k - a_{k-2}$ for PR4 signaling, where $a_k \in \{\pm 1\}$ are the bits to be detected. Since there is noise due to finite-length constrained minimum mean-squared error (MMSE) equalization and white Gaussian noise added before the low pass filter, there is an added noise term denoted by n_k in the figure.

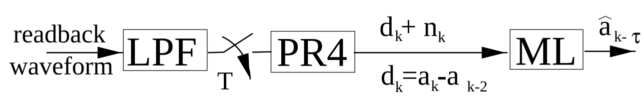

Fig. 1. Synchronous model of PR4ML system

Now we consider what happens when we sample asynchronously. When asynchronously sampling a low pass filtered read-back waveform from a magnetic recording system, each sample can be thought of as having a phase error associated with it. This phase error is the difference between the sampling time and the corresponding synchronous sampling time. With this view it is natural to think of using an interpolator that uses the phase error along with the asynchronous samples to interpolate to the synchronous sampling times. Another way of looking at the same situation is that each asynchronous sample has an amplitude error associated with it. The amplitude error is the difference between the asynchronous sample value and the value that would have been obtained for the corresponding synchronous sample. Thinking in these terms, the amplitude error will of course depend on phase, but it will also have a strong dependence on surrounding synchronous sample values. Using this perspective, it is natural to think of using an AEP filter that uses synchronous samples at the input. Of course, synchronous samples are not available; after all, that is what we are trying to obtain. However, the idea is that a less accurate estimate of the synchronous data can be used to drive the AEP to obtain a better estimate of it afterwards. An advantage of using the AEP approach is that in many cases the data driving the prediction filter is only two or three levels, allowing a much less complex implementation of the filter compared to an interpolator filter of the same size. Note that amplitude error also depends on the surrounding asynchronous sample values, and an AEP approach which uses asynchronous sample inputs reduces to the interpolation method presented next.

III. Interpolation Method

The approach that is usually taken when using an asynchronous sampling system is an interpolation method such as the one depicted in Fig. 2. This method is sometimes

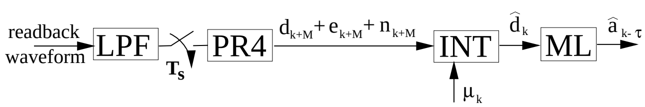

Fig. 2. Interpolation method applied to PR4ML system with fixed rate sampling clock (interpolator labeled as INT).

referred to as interpolating timing recovery. The only difference between the synchronous model shown in Fig. 1 and the interpolation model is that sampling is now done asynchronously and an interpolator has been added between the PR4 equalizer and the ML detector. Since sampling is done asynchronously the output of the equalizer is no longer $d_k + n_k$, but now has an added component, e_k, which is the amplitude error.

The interpolator block is realized as an FIR filter whose coefficients depend on the current sampling phase or fraction variable which is denoted by μ in the figure. In practice μ can be obtained in a way similar to that used in phase locked loops. The interpolator output would be input to a timing error detector, whose output in turn would be input to a loop filter. The loop filter output would then be input to a control mechanism which would provide the interpolator with the estimates of μ. This control mechanism is also responsible for providing a clocking signal, since there is oversampling and therefore more asynchronous samples than corresponding synchronous samples. It is not the focus of this paper to investigate the characteristics of this portion of the timing loop any further; for this see [5] or [6]. In our comparisons we assume all systems are provided with accurate estimates of μ.

Feasibility of implementation is always an issue, so it is of interest to look into how the interpolator can be implemented. A direct approach is to discretize μ and load new sets of filter coefficients for each new value of μ. In this case, the filter coefficients can be found ahead of time by using the least-mean square (LMS) algorithm. An alternative approach is to approximate each coefficient's variation with phase by a polynomial in μ. This can then lead to an efficient implementation called the Farrow structure [4]. One nice feature of using a Farrow structure is that as the number of interpolator coefficients increase, the number of multiplies involving μ do not. Multiplies involving μ only increase as the approximating polynomial order increases. From a formulation based on [7] a six tap interpolator with fairly good characteristics can be implemented using only 5 delay elements, 5 scaling operations, 2 multiplies, and 10 adders/subtracters.

Note that Fig. 2 shows interpolation being done after equalization; it is also possible to do interpolation before equalization. There are tradeoffs, however, involving timing loop delay and possible misequalization of the PR4 equalizer due to equalizing asynchronous samples.

IV. AEP Method

Fig. 3 shows how the AEP method can be applied to the PR4 system being considered. Comparing to Fig. 2

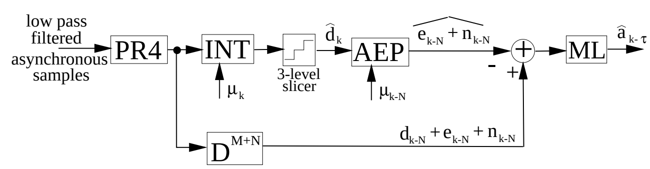

Fig. 3. AEP method applied to PR4ML system with fixed rate sampling clock.

a three level slicer, an AEP filter, and a subtracter have been added between the interpolator and the ML detector. A delay element has also been added to provide an appropriately delayed version of the PR4 equalizer output. Describing the operation of the proposed structure is simple. After PR4 equalization, samples are interpolated and sliced to provide an estimate of the synchronous data to the AEP filter. The AEP filter output is then subtracted from an appropriately delayed output from the PR4 equalizer before being input to the ML detector. Here, both the interpolator and AEP filters are provided with the phase variable μ. As in the case of the interpolating filter, the AEP filter is realized as an FIR filter whose coefficients depend μ and can be found using the LMS algorithm. Note that the AEP filter can also be implemented adaptively.

It may provide some insight to note that the AEP filter can be interpreted as shown in Fig. 4. The interpolator shown in this figure interpolates from synchronous sample

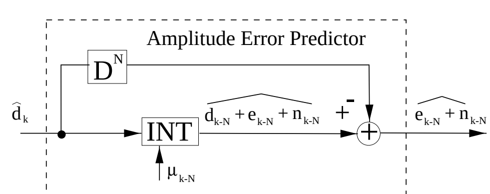

Fig. 4. Block diagram showing how amplitude error predictor can be viewed as consisting of an interpolator.

values to asynchronous sample values. Such an interpolator may be needed even in the interpolation approach of Fig. 2 in order to adapt the PR4 equalizer [3]. It would be advantageous if its presence could be made useful in the forward direction as well.

V. Simulation Results

Simulations were done at a user density of 2.0 using a rate 8/9 (0,4/4) code. A 6th order Butterworth low pass filter and a 10 tap PR4 equalizer was used in all simulations. SNR is defined using the peak signal level before low pass filtering and rms noise after lowpass filtering. Fig. 5 shows results for the case of 5 percent oversampling. Results are shown for the interpolation method using a 6 coefficient interpolator and the AEP method using a 4 coefficient interpolator with 8 AEP coefficients. The results indicate that under the assumed model the interpolator method outperforms the AEP method.

The same results as those shown in Fig. 5 are also shown in Fig. 6 but for the case of 1 percent oversampling. Again the AEP method is not as good as the interpolation method. There is an additional result shown in Fig. 6, however, which corresponds to the results obtained when using correct inputs to the AEP filter instead of estimates from the 3-level slicer. In this case, performance is better than the interpolation method. Some of this performance increase can be attributed to the correlation of d_k with n_k since the PR4 equalizer is finite-length constrained and trained with the MMSE criterion. This result suggests that if an improved detector were to replace the 3-level slicer, some of the gain shown could be realized. Alternatively, AEP could also be done within the viterbi trellis, eliminating the use of an interpolator and the performance degradation due to preliminary incorrect decisions, at the cost of a more complex detector.

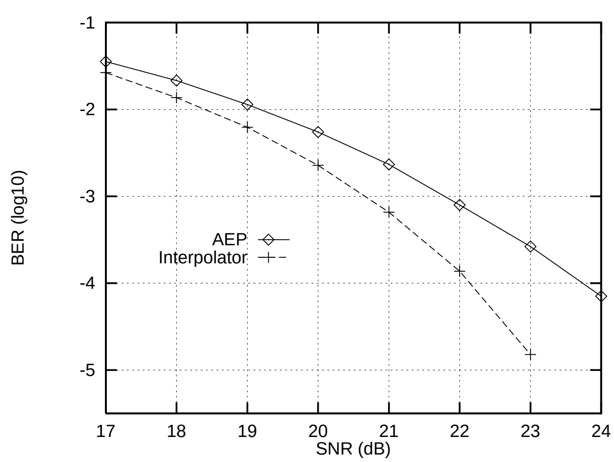

Fig. 5. Simulation results comparing AEP and interpolation methods at a user density of two using 5% oversampling.

Another encouraging observation was made. When detecting the same data, the AEP method and the interpolation method make a majority of their errors in different places. This suggests the possibility of multiplexing the schemes for an additional performance gain. To control such a multiplexing operation an indicator signal would need to be derived from some other signal in the system (e.g. phase, amplitude error, etc.).

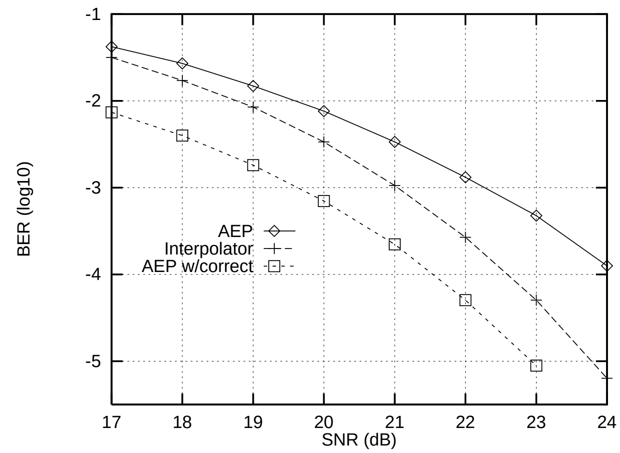

Fig. 6. Simulation results comparing AEP and interpolation methods at a user density of two using 1% oversampling

VI. Conclusions

A new way of viewing asynchronously sampling systems has been presented, along with a method for applying these ideas. Simulation results indicate that the new method does not outperform the traditional interpolation method. Characteristics pointed out in the simulation results section indicate that there still may be a potential for an AEP approach to outperform the interpolation method.

References

[1] C. M. Melas, and P. Sutardja, “An asynchronous fully digital channel for magnetic recording,” *IEEE Globecom Conference Record*, Piscataway, NJ, vol. 2, pp.1144-1147, 1994.

[2] M. Spurbeck and R. T. Behrens, “Interpolated timing recovery for hard disk drive read channels,” *ICC '97*, pp.1618-1624, 1997.

[3] Z-N. Wu, J. M. Cioffi, K. D. Fisher, “A MMSE interpolated timing recovery scheme for the magnetic recording channel,” *ICC '97*, pp.1625-1629, 1997.

[4] C. W. Farrow, “A continuously variable digital delay element,” *Proc. IEEE Int. Symp. Circuits & Syst.*, pp.2641-2645, June 1988.

[5] F. M. Gardner, “Interpolation in digital modems–part I: fundamentals,” *IEEE Trans. on Commun.*, vol. 41, no. 3, pp.501-507, March 1993.

[6] D. Kim, M. J. Narasimha, and D. C. Cox, “Unbiased timing-error estimation in the presence of nonideal interpolation,” *IEEE Trans. on Commun.*, vol. 45, no. 6, pp.647-650, June 1997.

[7] L. Erup, F. M. Gardner, and R. A. Harris, “Interpolation in digital modems–part II: implementation and performance,” *IEEE Trans. on Commun.*, vol. 41, no. 6, pp.998-1008, June 1993.