

A new combined equalization/synchronization technique for partial response systems (Invited Paper)

CAROLINE HUI EN CHIN, YUQIN MONG, XIAOSONG TANG, IAN LI- JIN THNG and HAILONG LI

Department of Electrical and Computer Engineering

National University of Singapore

10 Kent Ridge Crescent, Singapore 119260

SINGAPORE

“Ian Li- Jin Thng” <eletlj@nus.edu.sg> <http://www.ece.nus.edu.sg/staff/webpages/eletlj.htm>

Abstract: - A new combined equalization/synchronization technique, known as SC-PMC (self-convoluting partial mass center), for use in partial response systems is presented. The work is significant as the new technique utilizes a single tap delay line LMS (Least Mean Square) filter for performing the task of equalization and synchronization for the partial response transceiver. Conventional partial response systems often require a separate LMS filter for equalization and an additional synchronization block for synchronizing the transceivers. The new work will reduce transceiver components significantly at the price a more advanced LMS firmware algorithm which is able to handle equalization and synchronization at the same time. This work, although reported for a communication application, is also applicable in HDD (hard disk drive) applications where partial response techniques are often employed for HDD read back functionalities.

Key-Words: - Timing, Equalization, Synchronization, Equalization, Partial response, Self-Convolution

1 Introduction

Partial response communications facilitate the transmission of information with minimum bandwidth by allowing controlled inter-symbol interference to occur. Hence the use of partial response systems has been seen in many applications particularly in the field of bandwidth-limited communications and in high density hard disk drives.

The focus of this paper is the contribution of a new technique for combined equalization and synchronization for a partial response communication system. Conventionally, a synchronization block made up of maximum-likelihood sequence detection [1] is required for symbol synchronization and thereafter, a separate FSLE (fractionally spaced linear equalizer) is often required for channel equalization. The use of maximum-likelihood sequence detection is indeed computationally intensive and can result in some delay with symbol decoding [2]. In this new and novel contribution, a single FSLE, based on the conventional LMS (Least Mean Square) algorithm and enhanced with an additional TED (Timing error detection) algorithm called SC-PMC (self-convoluting partial mass center), is used to perform both equalization and synchronization respectively. The result is a reduction in hardware cost at the price of additional firmware which essentially does not increase hardware cost.

Analog TED methods includes PLL (Phase Lock Loop) designs [1], square law timing recovery methods, early-late gate [3] and maximum likelihood estimation [1-3]. This paper focuses purely on digital TED methods since the FSLE is essentially a digital filter. With advancement in VLSI chip technology, timing recovery performed in the digital domain would allow for reduced hardware cost and increased computational efficiency as seen in [4].

The additional SC-PMC algorithm allows the FSLE to make additional adjustments to the sampling phase of the FSLE which compensates for the synchronization error.

Several numerical simulations are presented to illustrate the usefulness of the new algorithms. In addition, an actual hardware realization of the algorithm via firmware implementation the SC-PMC algorithm on two TI C6711 DSP cards is also presented to demonstrate the practical use of the new FSLE structure. This paper is organized as follows: In Section 2, several existing TED algorithms for use in full response systems are briefly reviewed including the FMC (Full Mass Center) approach. We contribute a modification to the FMC approach to obtain the PMC (partial mass center) approach. Reasons and justification for using the PMC approach rather the FMC approach will be provided. In Section 3, we demonstrate that the PMC approach, although useful for use in the full response system, is unsatisfactory for the partial response system. Consequently, another additional improvement, using self-convolution, is presented. The usefulness of the resulting approach, i.e. SC-PMC, is then demonstrated using numerical simulations from Matlab. Finally, in section 4, numerical results stemming from the firmware implementation of the SC-PMC technique on two TI (Texas Instruments) C6711 DSP cards is presented.

2 TED for Full Response Systems

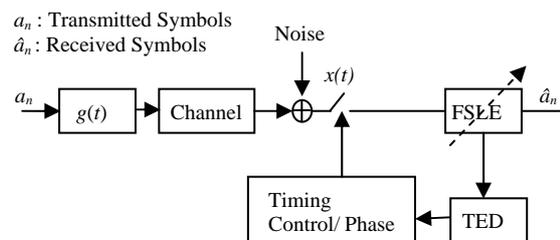


Fig. 1: Combined Equalization and Synchronization

Fig. 1 illustrates the block diagram of a typical FSLE structure for combined equalization and synchronization

when the transmitter response $g(t)$ is a full response pulse, for example, a square-root raised cosine pulse. In a full response system, inter-symbol interference is totally eliminated but the minimum bandwidth of a full response system is twice that of the partial response system. In Fig. 1, it is noted that as adaptive equalization is performed, it is possible to obtain further information on the degree of synchronization between the FSLE clock and the input clock. This is done by tracking the values of the adaptive FSLE weights. To illustrate further, let the weights of the FSLE be $\mathbf{w} = [w_0, w_1, \dots, w_{m-1}, w_m, w_{m+1}, \dots, w_k]^T$ where w_m is the middle weight, i.e. $m = k/2$, and k is an even number (so that the total number of weights is odd, i.e. $k+1$). In addition, w_n is the weight corresponding to the tap delay value of nTF where T is the symbol period of the input and F is the fractional sampling period of the FSLE. For example, if $F=1$, then we have a T -spaced linear equalizer and if $F=1/4$, then we have a $T/4$ -spaced FSLE. It is noted that if the transmitting waveform $g(t)$ is square root raised cosine, then with perfect synchronization, the optimal (i.e. converged) weights of the FSLE is a sampled version of $g(-t)$ where specifically, $w_m^* = g(0)$ and $w_{m+i}^* = g(-iTF)$ for $i \in Z_{-m}^m$ where Z_i^j represents a ring of integers from i to j . Now assume there is a glitch in the set of training signals such that the training sequence of the FSLE is suddenly and instantaneously time-shifted ahead by a value of TF , in that case the optimal weights of the FSLE will shift left so that $w_{m-1}^* = g(0)$, $w_{m-1+i}^* = g(-iTF)$ for $i \in Z_{-(m-1)}^{m-1}$. Conversely, if the same glitch instead time-shifted the FSLE training sequence backwards by a value of TF , then the optimal weights of the FSLE will shift right accordingly so that $w_{m+1}^* = g(0)$, $w_{m+1+i}^* = g(-iTF)$ for $i \in Z_{-(m+1)}^{m-1}$. By the property of the square root raised cosine pulse, $g(0)$ is the maximum value of $g(t)$. This means that by monitoring the position of the maximum weight value wrt (with respect to) the middle weight position, one will be able to know whether the glitch is a forward glitch or backward glitch. It is clear that a forward glitch represents a sudden and instantaneous increase of the FSLE clock wrt the input clock. Conversely, a backward glitch represents a sudden and instantaneous decrease of the FSLE clock wrt the input clock. In the examples mentioned so far, we have assumed that the glitch is instantaneous and the optimal weights are also obtained instantaneously. In the real system, the FSLE clock is several ppm (parts per million) faster/slower than the input clock. Consequently, when adaptive equalization begins, what occurs is that initially, a maximum value will build up around the middle weight value and then as time passes, the maximum weight value will gradually shift left or shift right due to the increasing difference in clock values. Therefore, by noting the shift behavior of the FSLE weights, one will be able to make corrections to the

sampling time of the FSLE to compensate for the difference in clock as illustrated in Fig. 1. Now, noting only the maximum weight value is perhaps the most rudimentary TED method. It should be noted that the FSLE is a digital filter and therefore the weights are samples of $g(t)$. Therefore, if the weights are moving towards a certain direction, there must be occasions where two neighboring weights will both have the same maximum value. Hence a much improved method for TED is to track the FMC (full mass centre) of the weights where

$$FMC = \frac{\sum_{i=0}^k i |w_i|}{\sum_{i=0}^k |w_i|} \quad [5,6].$$

Ideally $FMC = m$ where m is the index of the middle weight. If the FMC shifts away from m , then timing compensation must be done to bring FMC back to m . However, one of the problems associated with using FMC for TED detection is that it is sluggish. It is well known that in adaptive filters, the weights that converge slowest are the weights at the edge of the tap-delay line. Hence, using the FMC is definitely not the method for fast acquisition of the timing error since the edge weights are often not accurate enough for use. Hence for fast and efficient acquisition, a new timing function called PMC (partial mass center), which involves only three weights, is introduced as follows:

$$PMC = \frac{\sum_{i=M_0-1}^{M_0+1} i \cdot (w_i - \min[w_{M_0-1}, w_{M_0}, w_{M_0+1}])}{\sum_{i=M_0-1}^{M_0+1} (w_i - \min[w_{M_0-1}, w_{M_0}, w_{M_0+1}])} \quad (1)$$

where M_0 correspond to the index position of the maximum weight. The PMC subtracts $\min[w_{M_0-1}, w_{M_0}, w_{M_0+1}]$ away from the weights so that only positive values are encountered. This avoids the use of the absolute operator which can introduce distortion. Similar to FMC , the ideal value of PMC is also m . Notice

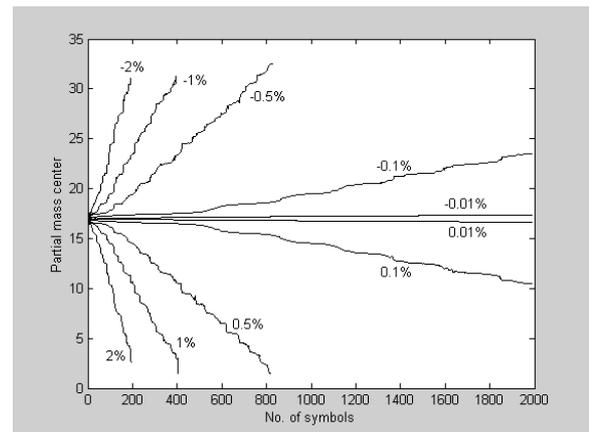


Fig. 2: Shifting of PMC for different timing errors

that the PMC computation does not increase if the number of weights in the FSLE increases. To appreciate the usefulness of the PMC method, Fig. 2 plots PMC against increasing number of symbols processed by a 33-taps $T/4$ -FSLE for different frequency errors between the

input clock and the FSLE clock. The frequency errors range from -2% to 2%. A 2% frequency error is equivalent to 20000 ppm which is extremely large. Most hardware clock components did not differ by more than 500 ppm. Notice in Fig. 2 that the PMC value shifts away from the middle tap point as time passes. In addition, the PMC provide an almost linear timing error cost so that the timing error can be given by

$$T_{err} = -\frac{PMC(n_2) - PMC(n_1)}{n_2 - n_1} \times \frac{F}{f_B} \quad (2)$$

where $PMC(n)$ is the PMC value after the FSLE has processed the n^{th} symbol. Consequently, the symbol rate of the FSLE can be adjusted using $T_r(n) = T_r(n-1) - T_{err}$ where $T_r(n)$ is the symbol period of the FSLE at the n^{th} update.

3 TED for Partial Response Systems

For convenience, we denote the following in relation to a partial response communication system. The overall response of a partial response communication system is given by $s_{pr}(t) = \text{sinc}(2t) + \text{sinc}(2(t-1/2))$. To provide maximum SNR at the output in the presence of AWGN influence, the transmitter frequency response is the square root of $S_{pr}(f) = \mathbf{F}[s_{pr}(t)]$ so that the transmitter impulse response of Fig. 1 is given by

$$g(t) = \mathbf{F}^{-1} \left[\sqrt{\mathbf{F}[S_{pr}(t)]} \right] \quad (3)$$

Accordingly, the optimum receiver impulse response is the matched response $g(-t)$. The input to the FSLE of Fig. 1 is $\sum_k a_k g(t-kT) + n(t)$ where $n(t)$ is AWGN and $a_n \in \pm 1, \pm 3$ is the n^{th} symbol to be transmitted. If the channel in Fig. 1 is lossy, then $g(t)$ is simply the convolution of (3) with the lossy channel impulse response. Let $M = 4$ denote the number of distinct symbols being transmitted, then the conventional algorithm for precoding and decoding partial response symbols is as follows:

Pre-coding:

Since $M = 4$, distinct Symbols = $\{\pm 1, \pm 3\}$

Let d_m be the uncoded data sequence, thus

Pre-coded sequence $p_m = d_m - p_{m-1}$

Transmitted Sequence $i_m = 2p_m - (M-1)$

Following which, the steps to decode the data are as follows:

Decoding:

Received Sequence $b_m = i_m + i_{m-1}$ and finally, the

Decoded Sequence $de_m = \frac{1}{2}b_m + M - 1$.

In the previous section, it was demonstrated that the PMC provided a linear timing detection curve for the full response system. However, in the partial response

domain, the PMC is no longer an effective TED tool. Fig. 3 illustrates three snapshots of the changing weight values of the FSLE for a small timing error. There are two peaks in the snapshots marked as *Main peak* and *Rising peak*. It is noted that as time passes, the *Main peak* decreases in value at the same position (rather than shift to the right) while the *Rising peak* increases in value at its current position. Thus the PMC of the system will essentially be localized at the tap position corresponding to the *Main peak* and appears to be stationary for a long time. As time passes, the height of the *Rising peak* will eventually overtake the decreasing height of the *Main peak*. When this happens, the *PMC* makes a discrete step size increment to the right corresponding to tap positions in the vicinity of the *Rising peak*. Essentially, rather than a smooth linear increase of the *PMC* as shown in Fig. 2, the *PMC* for the partial response system increases in discrete step-size jumps. Consequently, the timing compensation becomes very sluggish as the *PMC* appears to be stationary most of the time. One of the reasons contributing to the sluggishness of the *PMC* in the partial

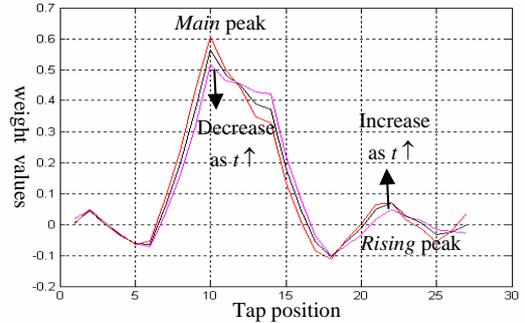


Fig. 3: Three snapshots of FSLE weights in the presence of synchronization error

response system is because of the introduction of controlled ISI. This essentially adds additional interference for each FSLE weights update event and the resulting adaptive process is not as reactive as the full response system. Hence the shifting of the *PMC* cannot be adequately captured by the adaptive mechanism of the FSLE. Hence an additional procedure is required and we refer to this as self-convolution. In SC-PMC, the weights are self-convoluted with each other first before the *PMC* position is decided. Consequently, this procedure accelerates the shifting effect of the *PMC*. Consider a simple weights snapshot example illustrated in Fig.4 where at initial state, $w_{0/1/2} = 10$ while $w_{4/5/6} = 1$ and all the other weights are zero. Assume that after every FSLE weight update, $w_{0/1/2}$ decreases by one while $w_{4/5/6}$ increases by one (i.e. same situation as Fig. 3) and all the other weights remain at zero. Clearly, PMC_w is at w_1 initially. Let $z = w \otimes w$. Then z consists of three symmetrical triangles with peaks at $[z_2 = 100, z_6 = 20, z_{10} = 1]$ as shown in Fig. 4. Clearly PMC_z is at z_2 initially. At the 3rd symbol update, the three peak values for z will change to $[49, 56, 16]$ so that

PMC_z is now shifted to z_6 . At the 7th symbol update, the three peak values for z will change to $[9, 48, 64]$ and PMC_z will move another four more positions to z_{10} . Conversely, PMC_w will shift (by four positions) only once at the 5th symbol update. In summary, self-convoluting the FSLE weights accelerates the PMC. In theory, one can self-convolute the weights as many times as possible at each update to obtain the desired level of PMC acceleration. For example, instead of stopping at $z = w \otimes w$, one can consider $q = w \otimes w \otimes w$ and even higher degrees of convolution where each degree of convolution increases the number of shift-multiply-sum operations as well as the length of the convoluted sequence. Modern DSP chips easily handles convolution in a few MCU cycles. Hence using $z = w \otimes w$ is still manageable for practical firmware implementation. In fact, a firmware demonstration of SC-PMC will soon be presented in Section 4.

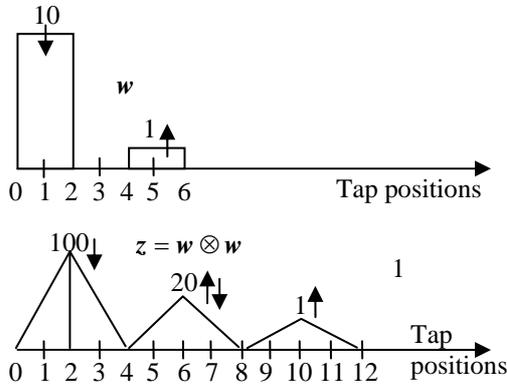


Fig. 4: PMC acceleration using self-convolution

Fig. 5 illustrates the timing mismatch tracking capability of the SC-PMC technique in a partial response communication system. The system is started with a positive timing error and then at the 200th symbol, the timing error is suddenly changed to the opposite sign. Timing errors range from 0 (no error) to as large as 0.01 (i.e. 1%). A $T/4$ -FSLE with 33 taps was used to obtain the results. The results clearly show that the SC-PMC timing function provides a reasonably good linear

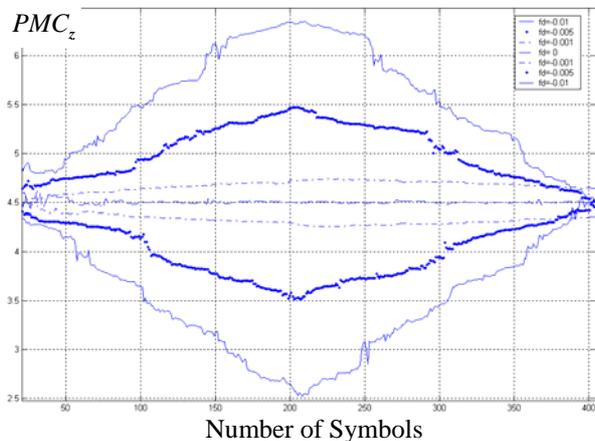


Fig. 5: SC-PMC tracking of the Timing Error

prediction of the timing error.

The algorithm for adjusting the sampling clock of the FSLE is based on a phase shift algorithm, followed by a control timing algorithm. The flow charts for these algorithms are illustrated in the Appendix. The phase shift algorithm ensures that at initial start-up, PMC_z is located at the midway position of the FSLE. The control timing algorithm will subsequently kick in to adjust the clock of the FSLE to ensure that PMC_z stays close to the middle

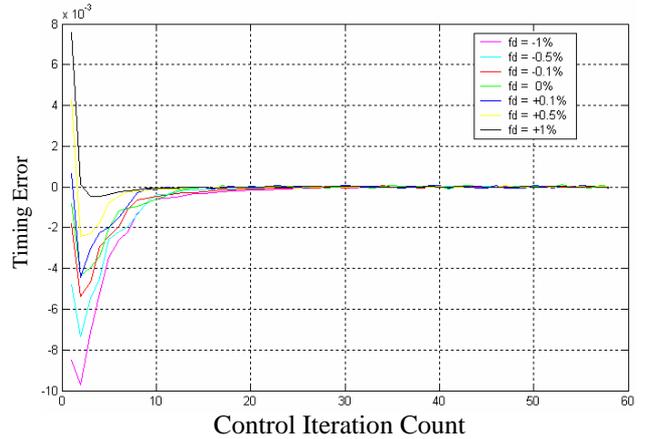


Fig. 6: Convergence of timing control algorithm

tap position. Fig. 6 thus illustrates several timing error trace for a 33-taps combined equalization/synchronization FSLE under lossy channel conditions. It is clear that after 10 control iterations, the timing error is totally eliminated (1 control iteration = 50 symbols).

Finally, Table 1 illustrates the margin performance of the combined equalization/ synchronization FSLE for symbol detection after convergence. The margin performance is a related to the BER of the transceiver. A margin of 0dB correspond to a BER of 10^{-7} . The higher the margin, the lower the BER. The first two rows of the table illustrate the importance of the self-convoluting weights operation associated with SC-PMC. The last two rows demonstrate that the margin performance in a lossy environment is no different from an ideal channel condition. This demonstrates clearly that the FSLE has indeed achieved combined equalization and synchronization for the partial response channel.

Freq. Error (%)	0	-0.1	0.1	-0.5	0.5	-1	1
PMC (dB) (ideal channel)	31.4	1.7	1.5	1.5	1.12	0.21	0.19
SC-PMC (dB) (ideal channel)	30.3	28.5	20.3	20.2	19.0	17.4	16.5
SC-PMC (dB) (lossy channel)	30.3	28.5	20.5	20.2	19.1	17.4	16.5

Table 1: Margin Performance of combined partial response $T/4$ -FSLE under various conditions

4 Hardware realization using two TI C6711 DSP cards

This section illustrates the practicality of the combined equalization/synchronization FSLE system for use in an

actual hardware partial response transceiver system as illustrated in Fig. 7. One C6711 card was programmed as the partial response transmitter while another C6711 card was programmed as the receiver. Several hardware programming constraints had to be programmed into the DSP cards to ensure that when the weights are self-convoluted, the real-time deadline is not missed. The clocks of the DSP cards run at a nominal rate of 8 kHz and are not adjustable. This means that weight interpolation [7,8] must be used to compensate for the timing error as well as prevent the input vector from “sliding out” of the FSLE tap delay line. The results for the hardware based system are as follows: Without timing recovery, the margin performance was -3.92 dB. With the SC-PMC timing recovery algorithm, the margin performance increased significantly to 19.65 dB. Fig. 8 illustrates the margin performance (first line) from the results window of the TI C6711 Code Composer Studio debugger software.

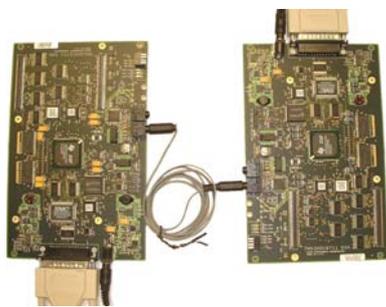


Fig. 7: The C6711 DSK Transceiver System

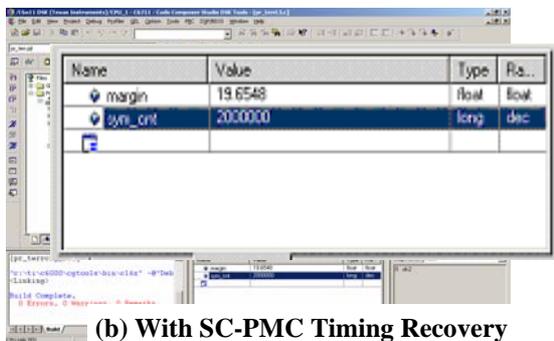
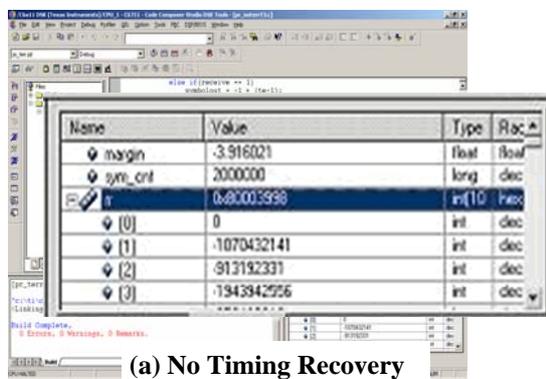


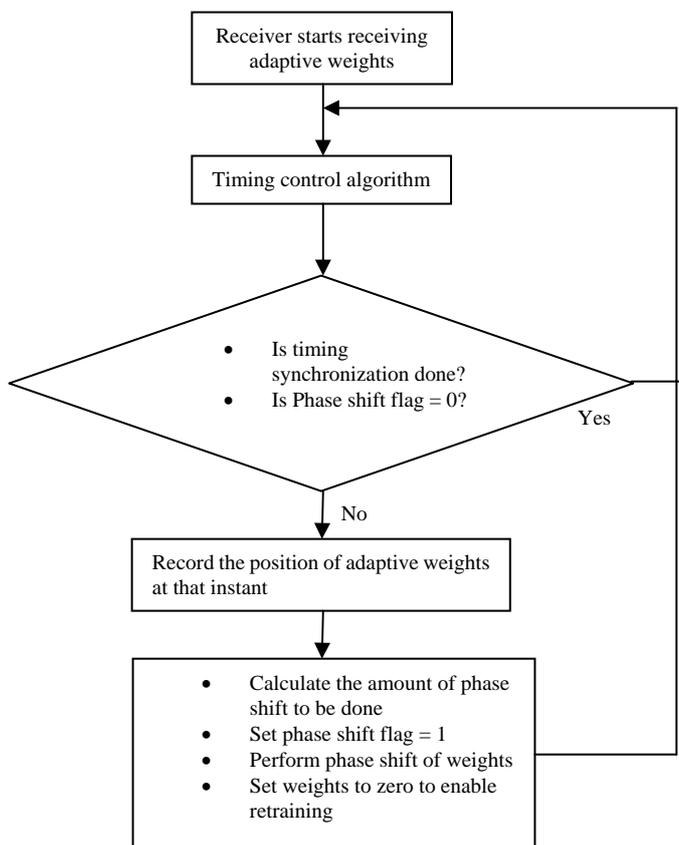
Fig. 8: Margin results from Code Composer Studio

5 Conclusions

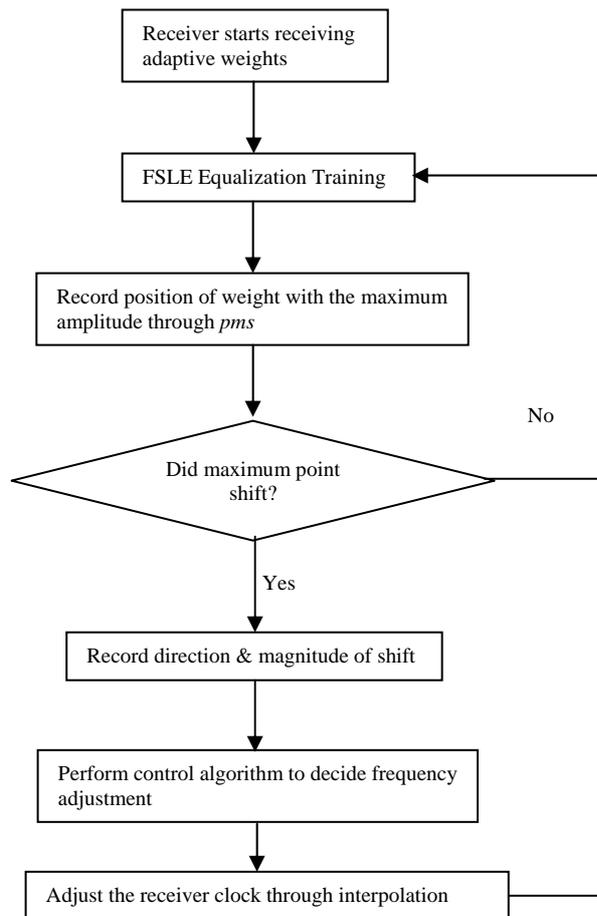
A new FSLE structure for combined equalization and synchronization is presented for use in a partial response communication system. The algorithm for TED (timing error detection) is based on a new and novel algorithm called SC-PMC (self-convoluting partial mass center). The SC-PMC technique is able to provide a good linear prediction of the timing error so that adequate timing compensation can be done. The new structure reduces hardware cost significantly since it removes the need for a separate synchronization block. The method has been numerically simulated and in addition successfully implemented in hardware DSP cards. This paper thus demonstrates not only the novelty of the new FSLE structure but also its practicality for hardware implementation.

6 Appendix

6.1 Flowchart of Phase Shifting Algorithm



6.2 Flow chart of Control Timing algorithm



7. Acknowledgements

The authors of this paper would like to thank Professor P. Stavrou. of the WSEAS Journals' Department for extending an invitation for this paper to be considered an invited paper. The authors are indeed well informed of the special status accorded to invited papers and the authors likewise extend their gratitude and appreciation to all reviewers who would be involved with the review process of this particular submission.

References:

- [1] J. G. Proakis, *Digital Communications*, 3rd edition, McGraw-Hill, Inc., 1995.
- [2] R.D. Gitlin & J. Salz, "Timing Recovery in PAM systems," *Bell Syst. Tech. J.*, vol. 5, May- June 1971, pp. 593-622.
- [3] Burton R. Saltzberg, "Timing Recovery for Synchronous Binary Data Transmission," *Bell Syst. Tech. J.*, vol. 46, Mar 1967, pp 593-622.
- [4] Kurt H. Mueller and Markus Muller, "Timing Recovery in Digital Synchronous Data Receivers," *IEEE Trans. On Commun.*, vol. COM-24, No.5, May 1976.
- [5] D. J. Artman, S. Chari and R. P. Gooch, "Joint Equalization and Timing recovery in a Fractionally-spaced Equalizer," *Proceedings of The 26th Asilomar Conference*

on Signals, Systems and Computers, vol.1, 1992, pp. 25-29.

[6] Y. Yuan and B. V. Kumar, "Use of adaptive filter for timing recovery for data storage channels," *Proceedings of the 2000 IEEE International Conference on Communications*, vol. 1, 2000.

[7] F. M. Gardner, "Interpolation in digital modems – Part I: Fundamentals," *IEEE Trans. on Commun.*, vol. 41, Mar 1993, pp. 502-508.

[8] L. Erup, F. M. Gardner and R. A. Harris, "Interpolation in digital modems – Part II: Implementation and performance," *IEEE Trans. on Commun.*, vol. 41, June 1993, pp. 998-1008.