Joint Batch Implementation of Blind Equalization and Timing Recovery

Bin Yang, Dalei Wang, and Ying Wu
Institute of Zhengzhou Information Science and Technology, Zhengzhou, Henan Province 450002, China
Email: yangbin.pmail@gmail.com; daleipla@163.com; hnwuying22@163.com

Abstract**—**In conventional design of a digital receiver, the timing recovery system and the equalizer are considered separately. Actually, the two processes are coupled and interacted. The paper proposes a batch scheme for jointly performing blind equalization and timing recovery, that embeds the timing recovery process inside the batch blind equalization algorithm. The proposed scheme adds the timing offset of sampling and interpolation filter in the objective function of constant modulus algorithm, iteratively updates the timing offset and equalizer coefficients in batch approach, thus achieves equalization, symbol timing error detector and timing recovery jointly. In this way, the timing recovery and the equalization processes can be coordinated. Simulation results show the performance of equalization and timing recovery.

Index Terms**—**blind equalization, timing recovery, constant modulus, open-loop batch

I. Introduction

In a digital communication receiver, an equalizer is used to eliminate the channel effect for reducing the intersymbol interference (ISI), and a timing recovery system is used to compensate for the timing offset between the transmitted data and the received sample. In conventional design, the equalizer and the timing recovery system are considered separately: the equalizer is designed assuming that the timing offset has been completely compensated and its processing data is sampled correctly, the interpolation filter of timing recovery system is designed assuming the channel is known and fixed [1], [2]. However actually, the processes of equalization and timing recovery are coupled. As a consequence, the adaptation circuits of the equalizer and of the timing synchronizer interact and thus the timing phase and the equalizer coefficients are drifting slowly, especially in the tracking mode [3]. This problem is targeted by some researchers, but most of them aimed to eliminate the interaction [3]-[7]. In [3], the solving approach is using the equalizer coefficients to estimate the timing error and feeding to the timing loop to cancel its effects. In [4], a new receiver structure is presented for joint timing recovery and equalization, which partitions the equalizer structure into a number of component parts, and position a magnitude equalizing portion prior to timing recovery. In [5]-[7], the asynchronous adaptive equalization approach is presented. Another kind of research is to design and realize timing recovery and equalization jointly. In [8], the interpolation filter for timing recovery and decision feedback equalizer is designed jointly to improve the performance. In [9], the Modified Constant Modulus Algorithm (MCMA) is extended to handle the timing offset parameter.

The performance of baud spaced equalizer (BSE) is very sensitive to the choice of the sampling phase. Therefore, highly accurate synchronization is required. Whereas the fractional spaced equalizer (FSE) is in principle independent of the sampling phase, although large drifts of the coefficients can also degrade the equalizer performance. Furthermore, the overall implementation complexity of FSE is significantly much higher than that of BSE. So, our research will focus on the jointly timing recovery and BSE. Due to its simplicity, constant modulus algorithm (CMA) [10] is the most commonly used algorithm in blind equalization from practical implementation point of view. Compared with the stochastic gradient descent realization of CMA (SGD-CMA), the open-loop batch approach using the 4-th order cumulants of the received signal represents faster convergence speed and better performance [11].

In this paper, we propose a joint batch blind equalization and timing recovery method. We add the timing offset of sampling and interpolation filter in the objective function of CMA, derive the iterative update formulas of the timing offset and equalizer coefficients in batch approach. Our simulation results show the convergence performance and the estimation accuracy of timing offset. The rest of the paper is organized as follows. In section II we introduce the system model for joint implementation of symbol timing recovery and blind equalization. Section III is devoted to the derivation of the joint batch iterative algorithm. Simulation results and performance analysis is the subject of section IV, which is followed by some conclusions in section V. Some pivotal proof is presented in Appendix.

II. System Model and Problem Formulation

We consider the baseband model of a single-input and single-output (SISO) system shown in Fig. 1. The

Manuscript received June 10, 2013; revised Month Day, 2013.
This work was supported by the National Natural Science Foundation of China under Grant No.61201381.
Corresponding author email: yangbin.pmail@gmail.com.
doi:10.12720/jcm.8.7.449-455

sequence of information bits I_k is applied to the digital modulator and converted into a complex valued symbol sequence a_k . The symbol sequence a_k enter the pulse shaping filter with the impulse response $g_T(t)$,then are distorted by the multipath channel $c(t)$,and further corrupted by additive white Gaussian noise(AWGN) $n(t)$ .In receiver, The received analog signal after the matched filter with the impulse response $g_R(t)$ is given by

$$r(t,\tau)=\sum_{k=-\infty}^{\infty} a_k h(t-\tau T-kT)+v(t) \qquad (1)$$

where $h(t)=g_T(t)*c(t)*g_R(t)$ is the overall baseband impulse response, $v(t)=n(t)*g_R(t)$ is the complex filtered noise with variance σ_v^2 . T is the symbol duration, and τ is the normalized fractional unknown timing offset between the transmitter and receiver ($|\tau|\le 1/2$).

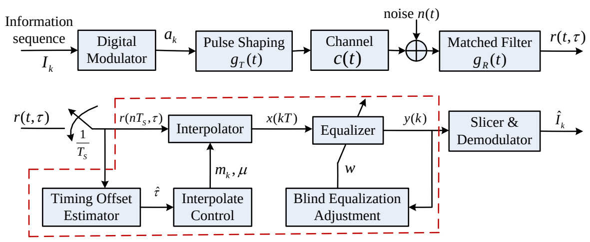

Figure 1. Baseband system model for timing recovery and blind equalization

After the analog to digital conversion, the received signal is oversampled with the timing offset at Q-times the symbol rate, that is $\frac{1}{T_s}=Q\frac{1}{T}$. So we get the digital signal

$$r(n)=r(nT_S,\tau)=\sum_{k=-\infty}^{\infty} a_k h(nT_S-\tau T-kT)+v(nT_S) \qquad (2)$$

where n is the sampling index.

In all-digital receiver, timing offset estimator is used to deal with the oversampled signal and estimate the timing offset $\hat{\tau}$ sent to the interpolator controller. The digital interpolator filters the oversampled signal $r(nT_S,\tau)$ using estimated parameters m_k (interpolator basepoint index) and μ (interpolator fractional interval), then produce the resumed digital signal $x(kT)$ which is equivalent to the signal sampled at exact sampling point. We denote $x(kT)$ as $x(k,\mu)$ to indicate that resumed samples are determined by μ . In order to remove ISI introduced by multipath channel, $x(k,\mu)$ is passed through the FIR equalizer with order L_w+1 , whose adjustable coefficients is denoted as $\mathbf{w}=[w_0 \quad w_1 \quad \cdots \quad w_{L_w}]^T$. The output of equalizer $y(k)$ becomes

$$y(k)=\sum_{i=0}^{L_w} w^*(i)x(k-i,\mu)=\mathbf{w}^H\mathbf{x}_{k,\mu} \qquad (3)$$

where $\mathbf{x}_{k,\mu}=[x(k,\mu) \quad x(k-1,\mu) \quad \cdots \quad x(k-L_w,\mu)]^T$.

At last the slicer and demodulator process the output of equalizer, and give the estimated result of the information sequence.

The purpose of our research is to jointly implement the timing offset loop and equalizer, which are included by the dashed line in Fig. 1. Showed as Fig. 2, the proposed algorithm adopts the open loop batch processing method, calculate the statistics of received samples, iteratively achieve accurate estimation of interpolator fractional interval μ and equalizer weights $\mathbf{w}$, at last accomplish interpolation and equalization.

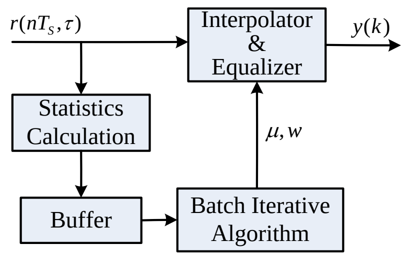

Figure 2. Baseband system model for timing recovery and equalization.

III. Proposed Algorithm

A. *Interpolator for Timing Recovery(TR)*

An FIR interpolator with impulse response $h_I[(i+\mu)T_s]$ is used to filter the received signal with timing offset [12], [13], and computes its output at sampling time $kT=(m_k+\mu)T_s$, that is

$$\begin{aligned} x(k,\mu) &= x[(m_k+\mu)T_s] \\ &= \sum_{i=-N_1}^{N_2} r[(m_k-i)T_s]\cdot h_I[(i+\mu)T_s] \end{aligned} \qquad (4)$$

where N_1+N_2+1 is the interpolator tap order. $m_k=\mathrm{INT}(kT/T_s)$ is called interpolator basepoint index, and decides which sample and succeeding N_1+N_2 samples is sent to the interpolator.

$\mu=kT/T_s-m_k$ is called interpolator fractional interval. It determines the coefficients of interpolator, that is to say, variable μ requests the recomputation of the filter coefficients.

The Farrow [14] structure of the interpolation filter is suited for signal interpolation by machine. It consists of L+1 parallel FIR branch components with fixed coefficients denoted as $c_l(i)$, for l =0,1,…,L, and only one variable parameter μ .

$$h_I[(i+\mu)T_s]=\sum_{l=0}^{L} c_l(i)\mu^l$$

So (4) can be denoted as

$$x(k,\mu)=\sum_{i=-N_1}^{N_2} r(m_k-i)\sum_{l=0}^{L} c_l(i)\mu^l \qquad (5)$$

In terms of hardware implementation complexity, The obvious advantage of Farrow structure is that the filter coefficients are fixed and the output samples is only related to the parameter μ . The design of the Farrow interpolator is based on polynomials, traditionally Lagrange polynomials. In this paper we use the filter coefficients $c_l(i)$ for the Lagrange interpolator polynomials, which can be expressed in the matrix form as [15]

$$\mathbf{Vc} = \mathbf{z}$$

where $\mathbf{V}$ is a Vandermonde matrix,

$$\mathbf{V} = \begin{bmatrix} 0^0 & 0^1 & 0^2 & \cdots & 0^L \\ 1^0 & 1^1 & 1^2 & & 1^L \\ 2^0 & 2^1 & 2^2 & & 2^L \\ \vdots & & & \ddots & \vdots \\ L^0 & L^1 & L^2 & \cdots & L^L \end{bmatrix}.$$

Vectors $\mathbf{c}$ and $\mathbf{z}$ are defined respectively as

$$\mathbf{c} = \left[C_0(z) \quad C_1(z) \quad C_2(z) \quad \cdots \quad C_L(z) \right]^T$$

$$\mathbf{z} = \left[1 \quad z^{-1} \quad z^{-2} \quad \cdots \quad z^{-L} \right]^T$$

And $C_l(z)$ is the transfer function of the lth FIR branch filter, which can be expressed as

$$C_l(z) = \sum_{i=0}^{N_1+N_2} c_l(i) z^{-i}$$

So, the solution of above equation is expressed as

$$\mathbf{c} = \mathbf{V}^{-1}\mathbf{z}. \tag{6}$$

Then we can get the interpolator coefficients $c_l(i)$ from the vector $\mathbf{c}$.

B. Open-Loop Batch Method of Constant Modulus Algorithm (OLB-CMA)

The cost function of constant modulus algorithm (CMA) is defined by

$$J_{CMA} = E\left[\left(|y(k)|^2 - R_2 \right)^2 \right] \tag{7}$$

where $E[\bullet]$ denotes statistical expectation and R_2 is a positive real constant defined by

$$R_2 = \frac{E\left[|a(k)|^4 \right]}{E\left[|a(k)|^2 \right]}$$

The equalizer coefficient vector update of steepest descent method is given by

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \Delta \cdot \nabla_{\mathbf{w}} J_{CMA} \tag{8}$$

where $\nabla_{\mathbf{w}} J_{CMA}$ denotes stochastic gradient of the CMA cost function with respect to the equalizer coefficients vector $\mathbf{w}$, Δ is the step-size parameter, $(\bullet)^{(k)}$ denotes the kth iterative value.

With limited samples of channel output data, the adaptation of the stochastic gradient can only be approximated. One well known approximation is to adopt the instantaneous value instead of the statistical expectation in the gradient calculation, which is called SGD-CMA. The weights update of SGD-CMA is given by:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \Delta \cdot \left(|y(k)|^2 - R_2 \right) y(k) \cdot \mathbf{x}^H(k) \tag{9}$$

Because of the rough estimation of the gradient, the traditional SGD-CMA usually requires a large number of iterations or samples to approximate the steepest descent counterpart, it means that SGD-CMA converge slowly.

On the other hand, the batch processing method of CMA [11] calculate the stochastic gradient directly from a block of channel output samples and achieve a much more accurate estimation of the gradient. Furthermore, it doesn't have to refilter the input of equalizer in each iteration. That is to say, it is a open loop batch iterative adaptation, called OLB-CMA. OLB-CMA converges much faster and more smoothly than SGD-CMA, its weights update is given by

$$\begin{aligned} \mathbf{w}^{(k+1)} &= \mathbf{w}^{(k)} - \Delta \cdot \nabla_{\mathbf{w}} J_{CMA} \\ &= \mathbf{w}^{(k)} - \Delta \cdot \left\{ \nabla_{\mathbf{w}} E\left[|y(k)|^4 \right] - 2R_2 \nabla_{\mathbf{w}} E\left[|y(k)|^2 \right] \right\} \end{aligned} \tag{10}$$

where: , $\mathbf{x}_k = [x(k) \quad x(k-1) \quad \cdots \quad x(k-L_w)]^T$ and $\mathbf{X}_k = \mathbf{x}_k \mathbf{x}_k^H$.

$$E\left[|y(k)|^2 \right] = \mathbf{w}^H E[\mathbf{X}_k] \mathbf{w} \tag{11}$$

$$\nabla_{\mathbf{w}} E\left[|y(k)|^2 \right] = 2E[\mathbf{X}_k] \mathbf{w} \tag{12}$$

$$E\left[|y(k)|^4 \right] = \mathrm{vec}^H(\mathbf{W}) E\left[\mathbf{X}_k^T \otimes \mathbf{X}_k \right] \mathrm{vec}(\mathbf{W}) \tag{13}$$

$$\nabla_{\mathbf{w}} E\left[|y(k)|^4 \right] = 4\,\mathrm{mat}\left(E\left[\mathbf{X}_k^T \otimes \mathbf{X}_k \right] \mathrm{vec}(\mathbf{W}) \right) \mathbf{w} \tag{14}$$

Here, $\mathbf{W} = \mathbf{w}\mathbf{w}^H$, $\otimes$ denotes the Kronecker product, $\mathrm{vec}(\bullet)$ function converts a matrix into a vector formed by stacking all columns of the matrix sequentially and the reverse operation $\mathrm{mat}(\bullet)$ converts the vector back to its matrix form.

According to (10)-(14), the update of weight vector do not require the output of equalizer, that is to say, OLB-CMA doesn't need convolution operation to produce the equalizer output in every step. It only computes the statistics about channel out $E[\mathbf{X}_k]$ and $E\left[\mathbf{X}_k^T \otimes \mathbf{X}_k \right]$ in starting step, then use (10) to optimize the equalizer iteratively.

C. Proposed Joint Equalization and TR Algorithm

As mentioned in section II, when we receive the channel out signal with timing offset , we want to eliminate the timing error and channel multipath interference simultaneously and cooperatively, like Fig. 2.

Since the input of equalizer is determined by channel output signal and interpolation filter, i.e. the interpolator fractional interval μ. So Introducing new parameter μ to the CMA cost function (7), we can use the only one cost function J_{CMA} to optimize $\mathbf{w}$ and μ.

Since the input of equalizer is determined by channel output signal and interpolation filter, i.e. the interpolator fractional interval μ. So Introducing new parameter μ to the CMA cost function (7), we can use the only one cost function J_{CMA} to optimize $\mathbf{w}$ and μ.

$$\begin{aligned}\mathbf{w}^{(k+1)} &= \mathbf{w}^{(k)} - \Delta_{\mathbf{w}} \cdot \nabla_{\mathbf{w}} J_{CMA} \\ &= \mathbf{w}^{(k)} - \Delta_{\mathbf{w}} \cdot \left\{ \nabla_{\mathbf{w}} E\left[|y(k)|^4 \right] - 2R_2 \nabla_{\mathbf{w}} E\left[|y(k)|^2 \right] \right\}\end{aligned} \tag{15}$$

$$\begin{aligned}\mu^{(k+1)} &= \mu^{(k)} - \Delta_{\mu} \cdot \nabla_{\mu} J_{CMA} \\ &= \mu^{(k)} - \Delta_{\mu} \cdot \left\{ \nabla_{\mu} E\left[|y(k)|^4 \right] - 2R_2 \nabla_{\mu} E\left[|y(k)|^2 \right] \right\}\end{aligned} \tag{16}$$

Obviously, the key of the above two iterative update equations is to calculate the gradient of the second and fourth statistics of equalizer out $y(k)$ with respect to $\mathbf{w}$ and μ, $\Delta_{\mathbf{w}}$ and Δ_{μ} is the step-size parameter of $\mathbf{w}$ and μ respectively.

Define:

$$\begin{aligned}\mathbf{r}_{m_k} = [r(m_k + N_1) \quad & r(m_k + N_1 - 1) \quad \cdots \\ & r(m_k) \quad r(m_k - 1) \quad \cdots \quad r(m_k - N_2)]\end{aligned} \tag{17}$$

$$\boldsymbol{\mu} = \begin{bmatrix} 1 & \mu & \mu^2 & \cdots & \mu^L \end{bmatrix}^T \tag{18}$$

$$\mathbf{C} = \begin{bmatrix} c_0(0) & c_1(0) & \cdots & c_L(0) \\ c_0(1) & c_1(1) & \cdots & c_L(1) \\ \vdots & \vdots & \ddots & \vdots \\ c_0(N_1+N_2) & c_1(N_1+N_2) & \cdots & c_L(N_1+N_2) \end{bmatrix} = [c_l(i)]_{(N_1+N_2+1)\times(L+1)} \tag{19}$$

where $\mathbf{r}_{m_k}$ is the channel out signal vector with timing offset, μ in the vector $\boldsymbol{\mu}$ is the interpolator fractional interval. $\mathbf{C}$ means the interpolator coefficients matrix, its lth column is the coefficient of the lth FIR branch filter, described as (6).

So we get the matrix form of the interpolator out denoted by (5) as.

$$x(k,\mu) = \mathbf{r}_{m_k}^T \mathbf{C}\boldsymbol{\mu}. \tag{20}$$

Furthermore, we give the interpolator out vector

$$\mathbf{x}_{k,\mu} = [x(k,\mu) \quad x(k-1,\mu) \quad \cdots \quad x(k-L_w,\mu)]^T = \begin{bmatrix} \mathbf{r}_{m_k}^T \\ \mathbf{r}_{m_{k-1}}^T \\ \vdots \\ \mathbf{r}_{m_{k-L_w}}^T \end{bmatrix} \mathbf{C}\boldsymbol{\mu} = \mathbf{R}_{m_k}\mathbf{C}\boldsymbol{\mu} \tag{21}$$

where we call $\mathbf{R}_{m_k}$ the channel out data matrix.

Define: $\mathbf{U} = \boldsymbol{\mu} \cdot \boldsymbol{\mu}^T$,

$$\begin{aligned}&\mathbf{X}_{k,\mu} = \mathbf{x}_{k,\mu}\mathbf{x}_{k,\mu}^H, \\ &\mathbf{C}^{[2]} = \mathbf{C} \otimes \mathbf{C}, \\ &\mathbf{C}^{[4]} = \mathbf{C} \otimes \mathbf{C} \otimes \mathbf{C} \otimes \mathbf{C}, \\ &\mathbf{R}_{m_k}^{[2]} = \mathbf{R}_{m_k}^* \otimes \mathbf{R}_{m_k} \\ &\mathbf{R}_{m_k}^{[4]} = \mathbf{R}_{m_k} \otimes \mathbf{R}_{m_k}^* \otimes \mathbf{R}_{m_k}^* \otimes \mathbf{R}_{m_k} \\ &\mathbf{U}^{[2]} = \mathbf{U}^T \otimes \mathbf{U}\end{aligned}$$

We can derive the expression of the variable and corresponding statistics that included in (15) and (16).

$$E\left[|y(k)|^2\right] = \mathrm{vec}^H(\mathbf{W}) E\left[\mathbf{R}_{m_k}^{[2]}\right] \mathbf{C}^{[2]} \mathrm{vec}(\mathbf{U}) \tag{22}$$

$$E\left[|y(k)|^4\right] = \mathrm{vec}^H(\mathbf{W})\, \mathrm{mat}\left\{ E\left[\mathbf{R}_{m_k}^{[4]}\right] \mathbf{C}^{[4]} \mathrm{vec}\left(\mathbf{U}^{[2]}\right) \right\} \mathrm{vec}(\mathbf{W}) \tag{23}$$

$$\nabla_{\mathbf{w}} E\left[|y(k)|^2\right] = 2\,\mathrm{mat}\left\{ E\left[\mathbf{R}_{m_k}^{[2]}\right] \mathbf{C}^{[2]} \mathrm{vec}(\mathbf{U}) \right\} \mathbf{w} \tag{24}$$

$$\nabla_{\mu} E\left[|y(k)|^2\right] = \mathrm{vec}^H(\mathbf{W}) E\left[\mathbf{R}_{m_k}^{[2]}\right] \mathbf{C}^{[2]} \mathrm{vec}(\nabla_{\mu}\mathbf{U}) \tag{25}$$

$$\nabla_{\mathbf{w}} E\left[|y(k)|^4\right] = 4\,\mathrm{mat}\left\{ \mathrm{mat}\left\{ E\left[\mathbf{R}_{m_k}^{[4]}\right] \mathbf{C}^{[4]} \mathrm{vec}\left(\mathbf{U}^{[2]}\right) \right\} \mathrm{vec}(\mathbf{W}) \right\} \mathbf{w} \tag{26}$$

$$\nabla_{\mu} E\left[|y(k)|^4\right] = \mathrm{vec}^H(\mathbf{W})\, \mathrm{mat}\left\{ E\left[\mathbf{R}_{m_k}^{[4]}\right] \mathbf{C}^{[4]} \left[\nabla_{\mu} \mathrm{vec}\left(\mathbf{U}^{[2]}\right)\right] \right\} \mathrm{vec}(\mathbf{W}) \tag{27}$$

So, the update of (15) and (16) can be completed with calculation of equation (24)-(27), don't need to refilter the channel out signal. The proof of (22-27) is showed in Appendix.

Two statistics $E\left[\mathbf{R}_{m_k}^{[2]}\right]$ and $E\left[\mathbf{R}_{m_k}^{[4]}\right]$, and some variables, such as $\mathbf{C}^{[2]}$, $\mathbf{C}^{[4]}$, $\mathrm{vec}(\mathbf{U})$, $\nabla_{\mu}\mathbf{U}$, $\mathrm{vec}\left(\mathbf{U}^{[2]}\right)$, $\nabla_{\mu}\mathrm{vec}\left(\mathbf{U}^{[2]}\right)$ are involved in the computation, that make the computation seemed complicated. But the two statistics can be estimated over all available received data before the iterative process and stored in the memory. Variables $\mathbf{C}^{[2]}$ and $\mathbf{C}^{[4]}$ can be decided in advance when we choose the interpolator. Variables $\mathrm{vec}(\mathbf{U})$, $\nabla_{\mu}\mathbf{U}$, $\mathrm{vec}(\mathbf{U}^{[2]})$ and $\nabla_{\mu}\mathrm{vec}(\mathbf{U}^{[2]})$ are only dependent on estimated μ, their expression can be derived very easy according to (18).

The proposed joint equalization and TR algorithm can be summarized in 3 steps after initialization.

Initialization: Initialize equalizer tap weight vector $\mathbf{w}$ with zeros and substitute 1 for the central tap. Initialize interpolator fractional interval $\mu = 0$. Calculate constants $\mathbf{C}^{[2]}$ and $\mathbf{C}^{[4]}$ using the solution of (19) and save the results in $(L+1)^2 \times (N_1+N_2+1)^2$ and $(L+1)^4 \times (N_1+N_2+1)^4$ storage space respectively.

Step 1: calculate the second and fourth order statistics $E\left[\mathbf{R}_{m_k}^{[2]}\right]$ and $E\left[\mathbf{R}_{m_k}^{[4]}\right]$ by averaging corresponding expression of channel output signal $r(n)$. The results will require respectively $(L_w+1)^2 \times (N_1+N_2+1)^2$ and $(L_w+1)^4 \times (N_1+N_2+1)^4$ storage space.

Step 2: calculate $\mathrm{vec}(\mathbf{U})$, $\nabla_\mu \mathbf{U}$, $\mathrm{vec}\left(\mathbf{U}^{[2]}\right)$, $\nabla_\mu \mathrm{vec}\left(\mathbf{U}^{[2]}\right)$ and $\mathrm{vec}(\mathbf{W})$ using estimated $\mathbf{w}$ and μ.

Step 3: update $\mathbf{w}$ and μ according to (15-16) and (24-27).

Repeat step2 and 3 until convergence or repeated time reaches the desired limit.

Last: use the optimal $\mathbf{w}$ and μ, get the equalizer output signal $y(k)$ as below.

$$y(k) = \mathbf{w}^H \mathbf{x}_{k,\mu} = \mathbf{w}^H \mathbf{R}_{m_k} \mathbf{C} \boldsymbol{\mu} \tag{28}$$

IV. Simulation Results and Analysis

In this section, we present simulation results to illustrate the performance of the proposed joint batch algorithm in terms of convergence and estimate precision of μ. Simulations are carried out in 25 dB SNR environment with QPSK and 16-QAM. The transmitter pulse shaping filter and receiver matched filter are all root raised cosine filters with roll off factor be set to 0.25. The multipath channel is taken from Signal Processing information database SPIB [16], which is defined as chan2.

For simplicity, the oversampling factor Q is set to 2 for implementing interpolation, the Farrow structure linear interpolator is chose. The interpolator coefficient matrix $\mathbf{C}$ can be calculate according to (15). Estimated timing offset $\hat{\tau}$ showed below in simulation results is half of the estimated interpolator fractional interval μ.

The equalizer is selected as a 15 tap filter with central tap initialized to 1. To measure the jointly algorithm effectiveness, we consider the mean square error (MSE) between the equalizer output and the transmitted symbol. Because the CMA cost function is phase blind, the equalizer output signal will perhaps have a phase rotation. We apply an DD phase recovery loop [17] after the equalizer output to estimate the phase rotation and then correct the phase offset.

Fig. 3 and Fig. 4 demonstrate the convergence behavior and estimate performance of the proposed algorithm in one run. By accurately estimating the statistics of received data, the MSE and estimated timing offset curves converge very smoothly.

Fig. 3 shows the simulation result for QPSK modulated transmitted symbol, timing offset is set to 0.2. Step size parameters Δ_w and Δ_μ are selected by trial method, here $\Delta_w = 0.0005$ and $\Delta_\mu = 0.001$. When the number of channel output symbols is 800, the algorithm convergence is achieved after about 200 iterations (solid curve). When the number of channel output symbol is 200, the convergence is achieved after about 400 iterations (dash curve). The convergence speed is much faster than SGD-CMA, whose slow convergence speed is well known. In both situation, the MSE converges to about -27dB, the estimated timing offset converges to 0.2 approx.

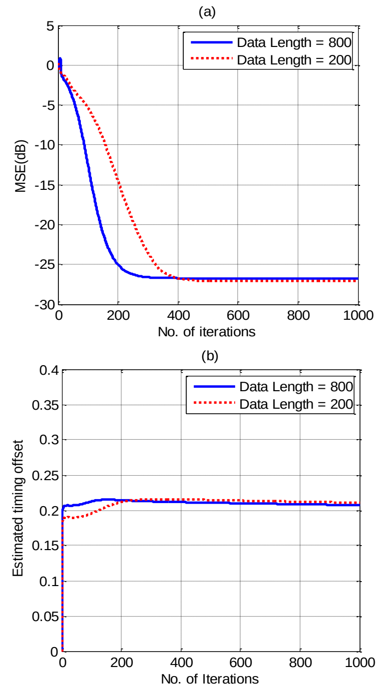

Figure 3. Simulation results for QPSK, timing offset is 0.2.(a) MSE performance. (b) Estimated timing offset

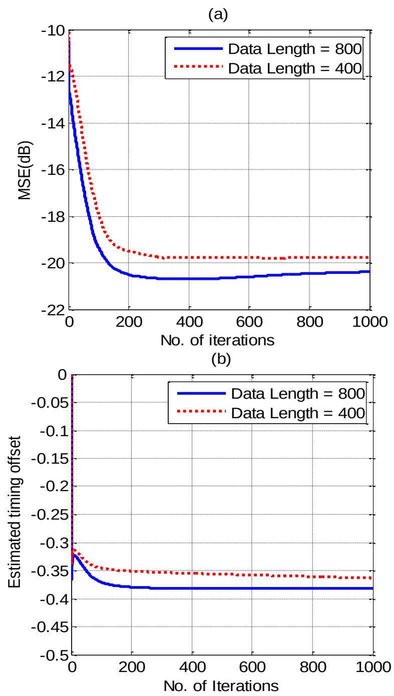

Figure 4. Simulation results for 16-QAM, timing offset is -0.3: (a) MSE performance. (b) Estimated timing offset

Fig. 4 shows the simulation result for 16-QAM modulated transmitted symbol, timing offset is set to -0.3, $\Delta_{\mathbf{w}} = 0.0005$ and $\Delta_{\mu} = 0.01$. We consider the situations in that data length is 800 and 400, the algorithm behaves similarly as previous simulation. MSE converges to about -20dB after about 200 iterations (solid curve and dash curve). The estimated timing offset converges to -0.35 approx.

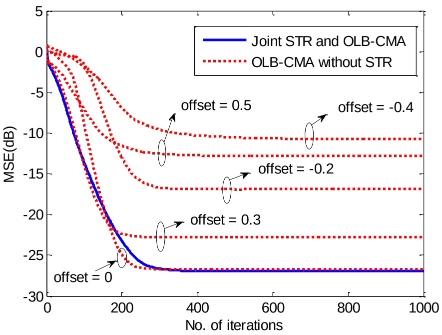

Figure 5. Convergence performance comparison

Fig. 5 compares the convergence performance of proposed algorithm (solid curve) and OLB-CMA (dash curves) with received data suffered various timing offset. Test condition and step size parameter are same as that of Fig. 3 with 800 data samples. The convergence performance of the proposed joint algorithm is close to the performance of OLB-CMA in situation that without timing offset. The faster convergence speed and better performance of OLB-CMA compared with traditional SGD-CMA has been described in [11].

V. Summary

In this paper, we present a new approach for jointly blind equalization and TR. The new approach complete the timing offset estimation, symbol timing recovery by interpolator and blind equalization simultaneously. Furthermore, the new approach modifies the CMA cost function, uses the only one cost function to optimize the timing offset and equalizer weight vector, lessens the interaction between the TR loop and blind equalizer, strengthens the joint effect. In the realization of the approach, batch method is adopted. By estimating the 2nd and 4th order statistics of received data in batch mode, we not only get the accurate estimated value, but also avoid refiltering received data in each iteration. Simulation results show that the joint OLB-CMA and TR algorithm can accomplish the timing recovery and blind equalization effectively, and its convergence performance is closed to the performance of OLB-CMA in situation that without timing offset existed.

Appendix

Proof of (22)—(27): According to (21), we rewrite $\mathbf{X}_{k,\mu}$ as

$$\begin{aligned}
\mathbf{X}_{k,\mu} &= \mathbf{x}_{k,\mu}\mathbf{x}_{k,\mu}^H = \mathbf{R}_{m_k}\mathbf{C}\boldsymbol{\mu}\cdot\left(\mathbf{R}_{m_k}\mathbf{C}\boldsymbol{\mu}\right)^H \\
&= \left(\mathbf{R}_{m_k}\mathbf{C}\right)\mathbf{U}\left(\mathbf{R}_{m_k}\mathbf{C}\right)^H \\
&= \mathrm{mat}\left\{\mathrm{vec}\left(\left(\mathbf{R}_{m_k}\mathbf{C}\right)\mathbf{U}\left(\mathbf{R}_{m_k}\mathbf{C}\right)^H\right)\right\} \\
&= \mathrm{mat}\left\{\left[\left(\left(\mathbf{R}_{m_k}\mathbf{C}\right)^H\right)^T \otimes \left(\mathbf{R}_{m_k}\mathbf{C}\right)\right]\mathrm{vec}(\mathbf{U})\right\} \\
&= \mathrm{mat}\left\{\left(\mathbf{R}_{m_k}^* \otimes \mathbf{R}_{m_k}\right)\left(\mathbf{C}\otimes\mathbf{C}\right)\mathrm{vec}(\mathbf{U})\right\} \\
&= \mathrm{mat}\left\{\mathbf{R}_{m_k}^{[2]}\mathbf{C}^{[2]}\,\mathrm{vec}(\mathbf{U})\right\} \qquad (29)
\end{aligned}$$

$$\begin{aligned}
\mathbf{X}_{k,\mu}^T\otimes\mathbf{X}_{k,\mu} &= \left(\left(\mathbf{R}_{m_k}\mathbf{C}\right)\mathbf{U}\left(\mathbf{R}_{m_k}\mathbf{C}\right)^H\right)^T \otimes \left(\left(\mathbf{R}_{m_k}\mathbf{C}\right)\mathbf{U}\left(\mathbf{R}_{m_k}\mathbf{C}\right)^H\right) \\
&= \left(\left(\mathbf{R}_{m_k}\mathbf{C}\right)^*\otimes\left(\mathbf{R}_{m_k}\mathbf{C}\right)\right)\left(\mathbf{U}^T\otimes\mathbf{U}\right)\left(\left(\mathbf{R}_{m_k}\mathbf{C}\right)^T\otimes\left(\mathbf{R}_{m_k}\mathbf{C}\right)^H\right) \\
&= \left(\left(\mathbf{R}_{m_k}\mathbf{C}\right)^*\otimes\left(\mathbf{R}_{m_k}\mathbf{C}\right)\right)\left(\mathbf{U}^T\otimes\mathbf{U}\right)\left(\left(\mathbf{R}_{m_k}\mathbf{C}\right)^*\otimes\left(\mathbf{R}_{m_k}\mathbf{C}\right)\right)^H \\
&= \mathrm{mat}\left\{\mathrm{vec}\left(\left(\left(\mathbf{R}_{m_k}\mathbf{C}\right)^*\otimes\left(\mathbf{R}_{m_k}\mathbf{C}\right)\right)\left(\mathbf{U}^T\otimes\mathbf{U}\right)\left(\left(\mathbf{R}_{m_k}\mathbf{C}\right)^*\otimes\left(\mathbf{R}_{m_k}\mathbf{C}\right)\right)^H\right)\right\} \\
&= \mathrm{mat}\left\{\left[\left(\left(\left(\mathbf{R}_{m_k}\mathbf{C}\right)^*\otimes\left(\mathbf{R}_{m_k}\mathbf{C}\right)\right)^H\right)^T\otimes\left(\left(\mathbf{R}_{m_k}\mathbf{C}\right)^*\otimes\left(\mathbf{R}_{m_k}\mathbf{C}\right)\right)\right]\mathrm{vec}\left(\mathbf{U}^T\otimes\mathbf{U}\right)\right\} \\
&= \mathrm{mat}\left\{\left[\left(\left(\mathbf{R}_{m_k}\mathbf{C}\right)\otimes\left(\mathbf{R}_{m_k}\mathbf{C}\right)^*\right)\otimes\left(\left(\mathbf{R}_{m_k}\mathbf{C}\right)^*\otimes\left(\mathbf{R}_{m_k}\mathbf{C}\right)\right)\right]\mathrm{vec}\left(\mathbf{U}^T\otimes\mathbf{U}\right)\right\} \\
&= \mathrm{mat}\left\{\left[\left(\mathbf{R}_{m_k}\otimes\mathbf{R}_{m_k}^*\otimes\mathbf{R}_{m_k}^*\otimes\mathbf{R}_{m_k}\right)\left(\mathbf{C}\otimes\mathbf{C}\otimes\mathbf{C}\otimes\mathbf{C}\right)\right]\mathrm{vec}\left(\mathbf{U}^T\otimes\mathbf{U}\right)\right\} \\
&= \mathrm{mat}\left\{\left[\mathbf{R}_{m_k}^{[4]}\mathbf{C}^{[4]}\right]\mathrm{vec}\left(\mathbf{U}^{[2]}\right)\right\} \qquad (30)
\end{aligned}$$

Substitute (29) and (30) into (11) and (13), we get (22)—(27) which given in Section III:

$$\begin{aligned}
E\left[\left|y(k)\right|^2\right] &= \mathbf{w}^H E\left[\mathbf{X}_{k,\mu}\right]\mathbf{w} = Tr\left(E\left[\mathbf{X}_{k,\mu}\right]\mathbf{W}\right) \\
&= \mathrm{vec}^H(\mathbf{W}^H)\,\mathrm{vec}\left(E\left[\mathbf{X}_{k,\mu}\right]\right) \\
&= \mathrm{vec}^H(\mathbf{W}^H)\,\mathrm{vec}\left(E\left[\mathrm{mat}\left\{\mathbf{R}_{m_k}^{[2]}\mathbf{C}^{[2]}\,\mathrm{vec}(\mathbf{U})\right\}\right]\right) \\
&= \mathrm{vec}^H(\mathbf{W})E\left[\mathbf{R}_{m_k}^{[2]}\right]\mathbf{C}^{[2]}\,\mathrm{vec}(\mathbf{U}) \qquad (22)
\end{aligned}$$

$$\begin{aligned}
E\left[\left|y(k)\right|^4\right] &= \mathrm{vec}^H(\mathbf{W})E\left[\mathbf{X}_{k,\mu}^T\otimes\mathbf{X}_{k,\mu}\right]\mathrm{vec}(\mathbf{W}) \\
&= \mathrm{vec}^H(\mathbf{W})\,\mathrm{mat}\left\{E\left[\mathbf{R}_{m_k}^{[4]}\right]\mathbf{C}^{[4]}\,\mathrm{vec}\left(\mathbf{U}^{[2]}\right)\right\}\mathrm{vec}(\mathbf{W}) \qquad (23)
\end{aligned}$$

Then we can derive (24-27) easily.

$$\begin{aligned}
\nabla_{\mathbf{w}}E\left[\left|y(k)\right|^2\right] &= 2E\left[\mathbf{X}_{k,\mu}\right]\mathbf{w} \\
&= 2\,\mathrm{mat}\left\{E\left[\mathbf{R}_{m_k}^{[2]}\right]\mathbf{C}^{[2]}\,\mathrm{vec}(\mathbf{U})\right\}\mathbf{w} \qquad (24)
\end{aligned}$$

$$\begin{aligned}
\nabla_{\mu}E\left[\left|y(k)\right|^2\right] &= \nabla_{\mu}\left[\mathrm{vec}^H(\mathbf{W})E\left[\mathbf{R}_{m_k}^{[2]}\right]\mathbf{C}^{[2]}\,\mathrm{vec}(\mathbf{U})\right] \\
&= \mathrm{vec}^H(\mathbf{W})E\left[\mathbf{R}_{m_k}^{[2]}\right]\mathbf{C}^{[2]}\,\mathrm{vec}(\nabla_{\mu}\mathbf{U}) \qquad (25)
\end{aligned}$$

$$\begin{aligned}
\nabla_{\mathbf{w}}E\left[\left|y(k)\right|^4\right] &= 4E\left[\mathbf{X}_{k,\mu}\mathbf{W}\mathbf{X}_{k,\mu}\right]\mathbf{w} \\
&= 4\,\mathrm{mat}\left\{\mathrm{vec}\left(E\left[\mathbf{X}_{k,\mu}\mathbf{W}\mathbf{X}_{k,\mu}\right]\right)\right\}\mathbf{w} \\
&= 4\,\mathrm{mat}\left\{E\left[\mathbf{X}_{k,\mu}^T\otimes\mathbf{X}_{k,\mu}\right]\mathrm{vec}(\mathbf{W})\right\}\mathbf{w} \\
&= 4\,\mathrm{mat}\left\{\mathrm{mat}\left\{E\left[\mathbf{R}_{m_k}^{[4]}\right]\mathbf{C}^{[4]}\,\mathrm{vec}\left(\mathbf{U}^{[2]}\right)\right\}\mathrm{vec}(\mathbf{W})\right\}\mathbf{w} \qquad (26)
\end{aligned}$$

$$\nabla_{\mu} E\left[|y(k)|^{4}\right] = \nabla_{\mu}\left\{\operatorname{vec}^{H}(\mathbf{W})\right.$$
$$\left.\operatorname{mat}\left\{E\left[\mathbf{R}_{m_k}^{[4]}\right]\mathbf{C}^{[4]}\operatorname{vec}\left(\mathbf{U}^{[2]}\right)\right\}\operatorname{vec}(\mathbf{W})\right\}$$
$$= \operatorname{vec}^{H}(\mathbf{W})\operatorname{mat}\left\{E\left[\mathbf{R}_{m_k}^{[4]}\right]\mathbf{C}^{[4]}\left[\nabla_{\mu}\operatorname{vec}\left(\mathbf{U}^{[2]}\right)\right]\right\}\operatorname{vec}(\mathbf{W}) \quad (27)$$

REFERENCES

[1] G. Watkins, "Optimal farrow coefficients for symbol timing recovery," *IEEE Commun. Lett.* , vol. 5, no. 9, pp. 381-383, 2001.

[2] D. Kim, M. J. Narasimha, and D. C. Cox, "Design of optimal interpolation filter for symbol timing recovery," *IEEE Trans. Commun.*, vol. 45, no. 7, pp. 877-884, Jul 1997.

[3] H. Schenk and D. Daecke, "Solving the interaction problem of timing synchronization and equalization," in *Proc. IEEE International Zurich Seminar on Communications*, 2008, pp. 52-55.

[4] D. Miniutti and R. A. Kennedy, "Novel receiver structure for joint timing recovery and equalization in frequency selective channels," in *Proc. Asia-Pacific Conference on Communications*, 2005, pp. 926-930.

[5] J. W. Bergmans, H. Pozidis, and M. Y. Lin, "Asynchronous zero - forcing adaptive equalization," *European Transactions on Telecommunications*, vol. 16, pp. 545-556, 2005.

[6] J. W. Bergmans, M. Y. Linb, D. Modriec, and R. Otte, "Asynchronous LMS adaptive equalization," *Signal Processing*, vol. 85, pp. 1301-1313, 2005.

[7] Y.-R. Chien, C.-Y. Lin, and H.-W. Tsao, "Reduction of loop delay for digital symbol timing recovery systems using asynchronous equalization," *IEEE International Symposium on Circuits and Systems*, 2009, pp. 193-196.

[8] M. H. Cheng and T. S. Kao, "Joint design of interpolation filters and decision feedback equalizers," *IEEE Trans. Commun.*, vol. 53, pp. 914-918, 2005.

[9] A. A. Nasir, S. Durrani, and R. A. Kennedy, "Modified constant modulus algorithm for joint blind equalization and synchronization," *Communications Theory Workshop*, 2010. pp. 59-64.

[10] D. N. Godard, "Self-Recovering equalization and carrier tracking in two-dimensional data communication systems," *IEEE Trans. Commun.*, vol. 28, no. 11, pp. 1867-1875, 1980.

[11] H.-D. Han, Z. Ding, J. Hu, and D. Qian, "On steepest descent adaptation: A novel batch implementation of blind equalization algorithms," in *Proc. IEEE Global Telecommunications Conference*, 2010, pp. 1-6.

[12] F. M. Gardner, "Interpolation in digital modems. I. fundamentals," *IEEE Trans. Commun.*, vol. 41, pp. 501-507, 1993.

[13] L. Erup, F. M. Gardner, and R. A. Harris, "Interpolation in digital modems. II. implementation and performance," *IEEE Trans. Commun.*, vol. 41, pp. 998-1008, 1993.

[14] C. W. Farrow, "A continuously variable digital delay element," in *Proc. IEEE International Symposium on Circuits and Systems*, 1988, pp. 2641-2645.

[15] V. Valimaki, "A new filter implementation strategy for Lagrange interpolation," in *Proc. IEEE International Symposium on Circuits and Systems*, 1995, pp. 361-364.

[16] S. P. I. B. [Online]. Available: http://spib.rice.edu

[17] K. N. Oh and Y. O. Chin, "Modified constant modulus algorithm: Blind equalization and carrier phase recovery algorithm," in *Proc. IEEE International Conference on Communications*, vol. 1, 1995, pp. 498-502.

Bin Yang received the M.S. degree in electrical engineering from Institute of Zhengzhou Information Science and Technology, Henan, China in 1996. He is currently an associate professor of Institute of Zhengzhou Information Science and Technology. His research interests are in signal processing and communications, include blind adaptive equalization.

Dalei Wang received the M.S. degree in electrical engineering from Institute of Zhengzhou Information Science and Technology, Henan, China in 2009. He is currently an graduate student. His research interests is blind adaptive equalization.

Ying Wu received the M.S. degree in electrical engineering from Beijing Institute of Technology, China in 1985. She is currently professor of Institute of Zhengzhou Information Science and Technology. Her current research interests are in communications and array signal processing.