Iterative Timing Recovery with Turbo Decoding at Very Low SNRs

Jianrong Bao, Minjian Zhao, Jie Zhong and Yunlong Cai
Zhejiang Provincial Key Laboratory of Information Network Technology,
Department of Information Science and Electronic Engineering,
Zhejiang University, Hangzhou, Zhejiang, P. R. China, 310027
Email:{ibaojr, mjzhao, zhongjie, ylcai}@zju.edu.cn

***Abstract*— Turbo codes are near Shannon limit channel codes widely used in space communication systems and so on at low signal-to-noise rate (SNR). Timing recovery is one of the key technologies for these systems to work effectively. In this paper, an efficiently iterative timing recovery with Turbo decoding is presented. By maximizing the sum of the square of soft decision metrics from Turbo decoding, it can obtain accurate timing acquisition. And a computation-efficient approximate gradient descent method is adopted to obtain rough estimate of timing offset. Another merit of it is that, by the proposed method, a rate-1/6 Turbo coded binary phase shift keying (BPSK) system can even work at very low SNR (E_s/N_0) about -7.44 dB without any pilot symbol. Finally, the whole timing recovery scheme is accomplished where the proposed method is combined with Mueller-Müller (M&M) timing recovery which performs the timing track. Simulation results indicate that the Turbo coded BPSK system with rather large timing errors by the proposed scheme can achieve performance within 0.1 dB of the ideal code with reasonable computations and storages.**

I. Introduction

Turbo codes have received extensive attention for their ability to approach Shannon capacity [1]. Since Turbo encoding is much simpler than that of another similar Shannon capacity approaching codes, i.e. low-density parity-check (LDPC) codes, they can be suitably applied in hardware resource and energy limited transmitter of space probes or satellites. However, although maximum a posteriori (MAP) algorithm for Turbo decoding is optimal, it still encounters complex numerical calculation problems as non-linear functions and a large number of additions and multiplications [1]. So an equivalent but much simpler logarithm version, i.e. Log-MAP algorithm, has been developed with lower complexity which makes the codes much more practicable [2]. But realizing full potential of Turbo codes requires that the baseband processing of the receivers can successfully acquire timing recovery. Traditionally, timing recovery has worked independently of channel decoding. However, in the iterative timing recovery [3], the metrics generated from channel decoding can be fed back to the timing recovery loop and thus obtain rather accurate timing synchronization at low SNRs where traditional timing recovery methods usually fail.

Previous treatments of iterative timing recovery mainly focused on the use of output codewords produced by decoding of channel codes [3]. Recently, a method with decision metrics of LDPC decoding as code constraint feedback was proposed to achieve good timing recovery with acceptable complexity [4], [5]. Also an LDPC related timing recovery was proposed to estimate the timing error by metrics as absolute value of the sum of all soft metrics from LDPC decoding [6]. Another Turbo code aided timing recovery was also suggested to get right timing phase offset by calculating the weighted soft decision metrics [7]. But it did not include the recovery of the timing frequency offset. In addition, a more complex expectation maximization (EM) based turbo synchronization was put forward for timing recovery [8] which was more complex. In this paper, based on [6], we suggest a new timing recovery algorithm by exploiting the soft decision metrics of Turbo decoding and it can obtain both timing frequency and phase recovery. In our algorithm, maximizing the sum of the square of the soft decision metrics of Turbo decoding provides potentially precise measure for the estimation of the timing offsets of the received signals. And it is carried out with an approximate gradient descent method [9] efficiently. Moreover, we also use the Turbo decoded codewords aided traditional timing recovery method (i.e. the M&M timing recovery algorithm [10]) like [4] as a supplement to rectify the remain distortions, induced by the residual error of our algorithm and other timing errors, more accurately.

The paper is organized as follows. In Section II, the timing error model is introduced. Section III gives the principle of the suggested iterative timing recovery with Turbo decoding. And the implementation of the proposed scheme is also stated in this Section. In Section IV, the numerical simulations are given to manifest the good performance of the proposed scheme. In this section, we also analyze the possible reasons for the good performance of our algorithm and evaluate the computation complexity. Finally, the conclusion is drawn in Section V.

II. Timing Error Model (TEM)

In a Turbo coded BPSK communication system, each transmitted signal is composed of N point impulse responses $g(mT)$ from a square root raised cosine (SRRC) pulse shaping filter with sampling rate L where T is the symbol interval. The waveforms of the signals are shaped as the sequences of b_i through the above SRRC filter where $a_i \in \{\pm 1\}$ is the BPSK modulated value of the i-th symbol and b_i is the sampled symbol interpolated with $(L-1)$ zeros between each adjacent

978-1-4673-0990-5/12/$31.00 ©2012 IEEE

a_i to fit the sampling rate of the filter. So the transmitted signal $s(mT)$ is represented as

$$s(mT) = \sum_{i=0}^{N-1} b_i \cdot g(mT - iT/L). \tag{1}$$

Given a system without timing errors, a received sequence with L-times up-sampling rate can be expressed as

$$r(kT/L) = s(kT/L) + n(kT/L), \tag{2}$$

where $r(kT/L)$ is the received signal, $n(kT/L)$ is the additional white Gaussian noise (AWGN) introduced by the channel and all of them are sampled at interval T/L. When timing errors occur, the assumed time reference for the m-th sample at the receiver differs from the corresponding time reference at the transmitter according to timing offset τ_m. After L times down-sampling, the m-th sampled signal is

$$r(mT) = s(mT + \tau_m) + n(mT). \tag{3}$$

Finally, replacing $s(\cdot)$ with (1) , we get $r(mT)$ as

$$r(mT) = \sum_{i=0}^{N-1} b_i \cdot g(mT + \tau_m - iT/L) + n(mT). \tag{4}$$

Since there have been proper TEMs as constant timing phase offset, constant timing frequency offset and random walk [3], we will consider our TEM as the combination of these models and it can be represented as

$$\tau_m = \tau_{m-1} + T{\cdot}N(0, {\sigma_\tau}^2) + T{\cdot}F_{ppm}/10^6, (\tau_0 = D), \tag{5}$$

where initial sample time is offset by the constant timing phase offset D with respect to ideal sampling time and then the timing error is further perturbed according to a zero mean Gaussian random variable (random walk) with variance ${\sigma_\tau}^2$, denoted by $N(0, {\sigma_\tau}^2)$ and a constant timing frequency offset F_{ppm} measured in parts per million (ppm).

III. Proposed Iterative Timing Recovery Scheme

A. The MAP and Log-MAP Algorithms for Turbo Decoding

Turbo decoding has been proposed as MAP algorithms in an iterative manner [1]. In addition, an equivalent Log-MAP algorithms has been suggested for simplifying Turbo decoding with less complexity [2]. Then, with the notations in [2], the MAP and Log-MAP algorithm can be simply stated as follows [2] and metrics output from the Turbo decoder can be used for iterative timing recovery.

Given state S_k at time instance k, it take on values between 0 and $2^M - 1$ where M is the number of memory elements of the Turbo component code. The soft decision metric of the k-th bit d_k, i. e. $\Lambda(d_k)$, can be expressed as

$$\Lambda(d_k) = \ln\left(\frac{\sum\limits_{S_k}\sum\limits_{S_{k-1}} \gamma_1(y_k, S_{k-1}, S_k)\cdot\alpha_{k-1}(S_{k-1})\cdot\beta_k(S_k)}{\sum\limits_{S_k}\sum\limits_{S_{k-1}} \gamma_0(y_k, S_{k-1}, S_k)\cdot\alpha_{k-1}(S_{k-1})\cdot\beta_k(S_k)}\right), \tag{6}$$

where the forward recursion of the MAP is

$$\alpha_k(S_k) = \frac{\sum\limits_{S_{k-1}}\sum\limits_{i=0}^{1} \gamma_i(y_k, S_{k-1}, S_k)\cdot\alpha_{k-1}(S_{k-1})}{\sum\limits_{S_k}\sum\limits_{S_{k-1}}\sum\limits_{i=0}^{1} \gamma_i(y_k, S_{k-1}, S_k)\cdot\alpha_{k-1}(S_{k-1})};$$

$$\alpha_0(S_0) = \begin{cases} 1 \ , for & S_0 = 0 \\ 0 \ , & others \ , \end{cases} \tag{7}$$

the backward recursion is

$$\beta_k(S_k) = \frac{\sum\limits_{S_{k+1}}\sum\limits_{i=0}^{1} \gamma_i(y_{k+1}, S_k, S_{k+1})\cdot\beta_{k+1}(S_{k+1})}{\sum\limits_{S_k}\sum\limits_{S_{k+1}}\sum\limits_{i=0}^{1} \gamma_i(y_{k+1}, S_k, S_{k+1})\cdot\alpha_k(S_k)};$$

$$\beta_N(S_N) = \begin{cases} 1 \ , for & S_N = 0 \\ 0 \ , & others \end{cases} \tag{8}$$

and the branch transition probabilities is given by

$$\gamma_i((y_k^s, y_k^p), S_{k-1}, S_k) = q(d_k = i|S_k, S_{k-1}) \cdot \\ p(y_k^s|d_k = i)p(y_k^p|d_k = i, S_{k-1}, S_k)Pr(S_k|S_{k-1}). \tag{9}$$

The value "$q(d_k = i|S_k, S_{k-1})$" is either "1" or "0" depending on whether bit i is associated with the transition from state S_{k-1} to S_k or not. it is in the last component that we use a priori information for bit d_k. And without parallel transitions, $Pr(S_k|S_{k-1}) = Pr(d_k = 1)$ if $q(d_k = 1|S_k, S_{k-1}) = 1$ and $Pr(S_k|S_{k-1}) = Pr(d_k = 0)$ if $q(d_k = 0|S_k, S_{k-1}) = 1$.

Then by taking the logarithm of $\gamma_i((y_k^s, y_k^p), S_{k-1}, S_k)$ in (9) and inserting

$$p(y_k^p|d_k = i, S_k, S_{k-1}) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{1}{N_0}(y_k^p - x_k^p(i, S_k, S_{k-1}))^2}, \tag{10}$$

the expression for $q(\cdot) = 1$ can be obtained as:

$$\ln \gamma_i((y_k^s, y_k^p), S_{k-1}, S_k) = \frac{2y_k^s x_k^s(i)}{N_0} + \\ \frac{2y_k^p x_k^p(i, S_k, S_{k-1})}{N_0} + \ln Pr(S_k|S_{k-1}) + C. \tag{11}$$

In (11), C is the constant item which can be wiped out in calculating $\ln\alpha_k(S_k)$ and $\ln\beta_k(S_k)$. And N_0 is estimated to correctly measure the channel information with the priori probability $Pr(S_k|S_{k-1})$. Then if we define

$$\bar{\alpha}_k(S_k) = \ln\alpha_k(S_k), \tag{12}$$

$$\bar{\beta}_k(S_k) = \ln\beta_k(S_k), \tag{13}$$

$$\bar{\gamma}_i((y_k^s, y_k^p), S_{k-1}, S_k) = \ln\gamma_i((y_k^s, y_k^p), S_{k-1}, S_k), \tag{14}$$

the logarithm version of MAP algorithm, i.e. Log-MAP algorithm, can be achieved with much lower complexity.

With some numerical techniques [11], $\Lambda(d_k)$ can be calculated in an iterative soft-input-soft-output manner as (15),

$$\Lambda(d_k) = 2y_k/\sigma^2 + \tilde{\Lambda}_n + L_{e_n}, \tag{15}$$

where σ^2 is the channel noise variance, $\tilde{\Lambda}_n$ is the prior information of last iteration and it is initialized with "0" in the first iteration. L_{e_n} is the extrinsic information of the decoder and it is output as $\tilde{\Lambda}^{(n+1)}$ in the next iteration.

Moreover, the calculation of (6) in logarithm domain can be further simplified by using the Jacobian logarithm (See [2]) as

$$\begin{aligned} \ln(e^{\delta_1} + \cdots + e^{\delta_n}) &= \ln(e^{\delta} + e^{\delta_n}) \\ &= \max(\delta, \delta_n) + f_c(|\delta - \delta_n|) \\ with \quad e^{\delta} &= e^{\delta_1} + \cdots + e^{\delta_{n-1}} \\ and \quad f_c(|\delta_1 - \delta_2|) &= \ln(1 + e^{-|\delta_1 - \delta_2|}). \end{aligned} \tag{16}$$

Therefore, the calculation of (6) can be performed in the Log-MAP by $\bar{\alpha}_k(S_k)$, $\bar{\beta}_k(S_k)$, $\bar{\gamma}_i((y_k^s, y_k^p), S_{k-1}, S_k)$ and the Jacobian logarithm with lower computation complexity.

B. Iterative Timing Recovery via Soft Decision Metrics of Turbo Decoding with Log-MAP Algorithm

From Turbo decoding [1], [2] and the numerical simulations later, we find that the more adjacently the optimal sampling point is reached by the sampled symbol the larger square of the soft metrics (or the log-likelihood ratios, L-values, e.g. the L-value of the k-th codeword as $\Lambda(d_k)$ in (6)) of Turbo decoding will be because the square of the $\Lambda(d_k)$ will be larger with bigger square of the amplitude of the received signals and the AWGN with zero mean. This argument can be explained as the SNR analysis in [6], [7] where the more away from the optimal sampled points, the lower SNR will be for all data in Turbo iterations thus the sum of square of the L-values of them will decrease too. In addition, this argument can also be manifested by numerical simulations later. Meanwhile, similar argument is also true for timing frequency offsets. And correct L-values are crucial to Turbo decoding. Based on these analyses, we develop an efficient algorithm of timing recovery via soft metrics of Turbo decoding as follows.

Firstly, we defined an object function $\Psi(\tau)$ related to the timing offset τ as

$$\Psi(\tau) = \sum\nolimits_{i=0}^{N} [\Lambda(d_k|\tau)]^2, \tag{17}$$

where N is the length of the Turbo codeword and $\Lambda(d_k|\tau)$ is defined in (6) with τ from the second Turbo component decoder. So we can obtain the optimal $\tau(\tau_{opt})$ by searching possible region of τ, i.e. $[-T/2, T/2]$, to maximize $\Psi(\tau)$. And τ_{opt} can be got by solving

$$\tau_{opt} = \underset{\tau \in [-T/2, T/2]}{\text{argmax}} \; \Psi(\tau). \tag{18}$$

Then, similar techniques can be applied to estimate timing frequency offset $F_{ppm,opt}$ by searching possible region of F_{ppm} (e.g. $[-2000ppm, 2000ppm]$) to maximize similar object function like (17) except that τ is replaced by F_{ppm}. Finally, it is also suit for jointly optimal estimation of τ_{opt}

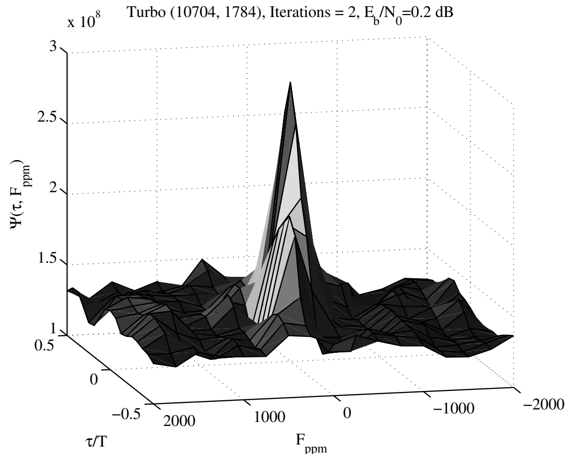

Fig. 1. Typical plot of $\Psi(\tau, F_{ppm})$ ˜ (τ, F_{ppm}).

and $F_{ppm,opt}$ as two-dimension search of possible region of τ and F_{ppm} to maximize object function $\Psi(\tau, F_{ppm})$ as (17) where τ is replaced by (τ, F_{ppm}). Therefore, parameters $(\tau_{opt}, F_{ppm,opt})$ can be calculated by

$$(\tau_{opt}, F_{ppm,opt}) = \underset{\substack{\tau \in [-T/2, T/2] \\ F_{ppm} \in Freq.Region}}{\arg\max} \{\Psi(\tau, F_{ppm})\}. \tag{19}$$

A typically numerical calculation result for object function $\Psi(\tau, F_{ppm})$ with independent variables τ and F_{ppm} is given in Fig. 1 where Turbo code (10704, 1784) [12] is chose at E_b/N_0 of 0.2 dB with just 2 decoding iterations. The two Turbo component convolutional codes are organized as follows: Both of them have feedback connections, or generate vector, (G_0=11001) in binary notation. The forward connections for the two component codes are identical as (G_1=11011, G_2=10101, G_3=11111). But the generate vector G_2 for component code two is punctured to satisfied code rate 1/6. Together with random interleaver and systematic information bits, there are 6 outputs per input bit. Results of other cases with different Turbo codes and E_b/N_0 are similar where there are one global peak value and some local peak values. From Fig. 1, the optimal estimation of τ and F_{ppm} can be gotten by searching maximum value of $\Psi(\tau, F_{ppm})$. Therefore, solving (19) is the key issue and listed below.

According to above analyses, the whole timing recovery system can be designed in Fig. 2 similar to [4]. From Fig. 1, due to symmetry of the object function with respect to independent variables, it is approximately convex and can be searched by gradient descent algorithm (also steepest descent) just as some adaptive equalization algorithm [9]. Then in our algorithm, we use a two-dimensional optimization search technique, i.e. gradient descent algorithm, by properly searching the vectors of τ and F_{ppm} in their searching region to update the optimal timing offset shown in (19). Since the object function derived is discontinuous without direct analytic expression thus it use a approximate gradient descent algorithm to implement the timing synchronization. Then the algorithm is listed as follows.

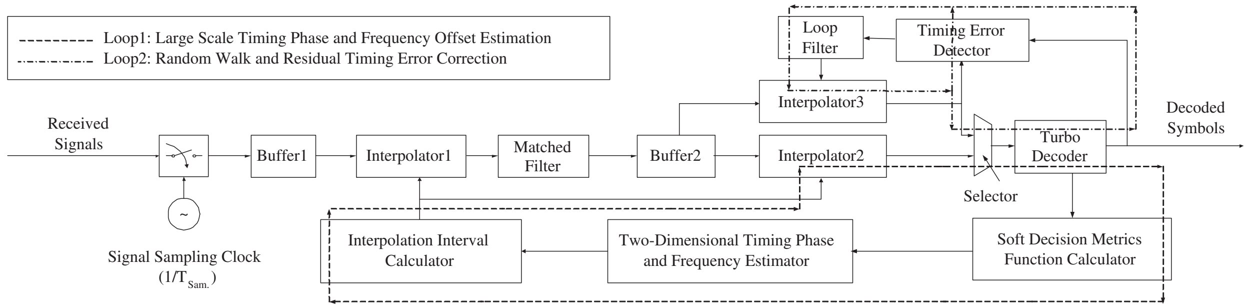

Fig. 2. Baseband equivalent receiver with iterative timing recovery in a BPSK communication system.

Firstly, we need to find relationship between object function $\Psi(\tau, F_{ppm})$ and independent variables τ and F_{ppm}, thus get approximate derivation of $\Psi(\tau, F_{ppm})$ to τ and F_{ppm}, respectively. So from (6)–(16) and (19),we get the derivations of them as

$$\frac{\partial \Psi(\tau, F_{ppm})}{\partial \tau} = 2\sum_{k=0}^{N} \Lambda(d_k|\tau, F_{ppm}) \frac{\partial \Lambda(d_k|\tau, F_{ppm})}{\partial \tau}$$
$$\frac{\partial \Psi(\tau, F_{ppm})}{\partial F_{ppm}} = 2\sum_{k=0}^{N} \Lambda(d_k|\tau, F_{ppm}) \frac{\partial \Lambda(d_k|\tau, F_{ppm})}{\partial F_{ppm}}. \quad (20)$$

From (15), the part $2y_k/\sigma^2$ is affected by the synchronization parameters τ and F_{ppm}, and it mainly decides $\Lambda(d_k|\tau, F_{ppm})$. So the approximation derivation of $\Psi(\tau, F_{ppm})$ to τ and F_{ppm} can be written as

$$\frac{\partial \Psi(\tau, F_{ppm})}{\partial \tau} \approx 2\sum_{k=0}^{N} \Lambda(d_k|\tau, F_{ppm}) \frac{\partial y_k}{\partial \tau} \cdot \frac{2}{\sigma^2}$$
$$\frac{\partial \Psi(\tau, F_{ppm})}{\partial F_{ppm}} \approx 2\sum_{k=0}^{N} \Lambda(d_k|\tau, F_{ppm}) \frac{\partial y_k}{\partial F_{ppm}} \cdot \frac{2}{\sigma^2}. \quad (21)$$

In section II and Fig. 1, given normalized parameters, we can approximately model the received signal y_k as

$$y_k \approx 1 - c_1 \cdot \text{sgn}(\tau) \cdot \tau - c_2 \cdot k \cdot \text{sgn}(F_{ppm}) \cdot F_{ppm} + n(t_k), \quad (22)$$

where the amplitude of transmitted signal is normalized as "1", the approximate affection (or transition trend) of τ and F_{ppm} to y_k is represented as "$-c_1 \cdot \text{sgn}(\tau) \cdot \tau$" and "$-c_2 \cdot k \cdot \text{sgn}(F_{ppm}) \cdot F_{ppm}$" and sgn is the sign function. Here, we just need approximate transition trend relationship of $\Psi(\tau, F_{ppm})$ to τ and F_{ppm} and more accurate value of them can be incorporated into positive constant number c_1 and c_2. Then the calculation of (21) is expressed as

$$\frac{\partial \Psi(\tau, F_{ppm})}{\partial \tau} \approx 2\sum_{k=0}^{N} \Lambda(d_k|\tau, F_{ppm})(-c_1 \text{sgn}(\tau)) \cdot \frac{2}{\sigma^2}$$
$$\frac{\partial \Psi(\tau, F_{ppm})}{\partial F_{ppm}} \approx 2\sum_{k=0}^{N} \Lambda(d_k|\tau, F_{ppm})(-c_2 k \cdot \text{sgn}(F_{ppm})) \cdot \frac{2}{\sigma^2}. \quad (23)$$

And (23) can be simplified by incorporating all constant value into negative value C_1 and C_2 as

$$\frac{\partial \Psi(\tau, F_{ppm})}{\partial \tau} \approx C_1 \sum_{k=0}^{N} \Lambda(d_k|\tau, F_{ppm}) \text{sgn}(\tau)$$
$$\frac{\partial \Psi(\tau, F_{ppm})}{\partial F_{ppm}} \approx C_2 \sum_{k=0}^{N} \Lambda(d_k|\tau, F_{ppm}) k \cdot \text{sgn}(F_{ppm}). \quad (24)$$

where C_1 and C_2 need to be conformed by simulations.

Finally, the item $2y_k/\sigma^2$, especially the y_k, can be approximately calculated by (22) with τ and F_{ppm} updated as

$$\tau^{(n+1)} = \tau^{(n)} - \lambda_1 \left(\frac{\partial \Psi(\tau, F_{ppm})}{\partial \tau}\right)^{(n)}$$
$$F_{ppm}^{(n+1)} = F_{ppm}^{(n)} - \lambda_2 \left(\frac{\partial \Psi(\tau, F_{ppm})}{\partial F_{ppm}}\right)^{(n)}. \quad (25)$$

where λ_1 and λ_2 are step parameters to update τ and F_{ppm}, also y_k, in Turbo decoding, superscript (n) demotes the n-th iteration. And y_k joins in the Turbo decoding to obtain (19).

Meanwhile, a conventional first-order PLL-based structure with a decision-directed Miller-Müller timing error detector (M&M TED) [10] is also used as supplement like [4] to implement iterative timing tracking. At each iteration, the M&M TED is executed by provided with more accurate symbols decoded by the Turbo decoder and the re-sampled received signals which are compensated with the optimal estimate of timing phase and frequency offset by our algorithm.

IV. Simulation Results and Performance Analyses

For all experiments, the Turbo code (10704, 1784) [12] is employed and a maximum of 10 iterations are adopted in Turbo decoding. SRRC pulse shaped and matching filters

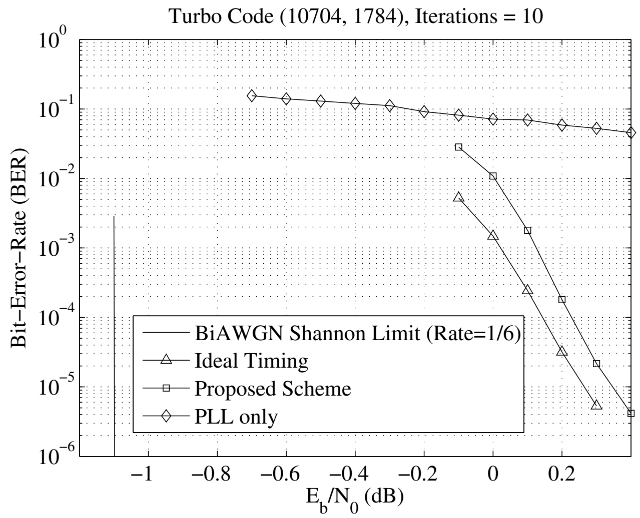

Fig. 3. BER performance of the proposed scheme to achieve timing recovery.

with a roll-off factor of 0.3 and a 25-tap FIR filter are used. Timing phase and frequency offsets over $\pm 0.5T$ (e.g. $0.3T$) and $\pm 1000ppm$ (e.g. $500ppm$) is chose respectively. The parameter σ_τ/T for the random walk will be 0.5% in all cases similar to [5]. Simulation at each E_b/N_0 will run until a specified number (e.g. 100) of error frames occur or a total of 1 million trials have been run. Simulation model of the receiver with iterative timing recovery and Turbo decoding is shown in Fig. 2. Step parameters λ_1 and λ_2 in (25) are selected as 0.01 and 0.03, respectively.

Fig. 3 gives the bit error rate (BER) performance of the proposed scheme with the TEM mentioned above. By our scheme, the performance of the Turbo coded system, suffered from large timing errors, is within 0.1 dB of the ideal code performance. Meanwhile, the performance of our scheme is also superior to that of the scheme with PLL only which exhibits a poor performance of more than 1 dB (at BER of 10^{-2}) away from the ideal code performance under the same TEM. Additional profit of our scheme is that it performs well at very low SNR without any pilot symbol. It is about -7.44 dB in SNR or 0.34 dB in E_b/N_0 at a BER of 10^{-5}. Here the relationship between SNR and E_b/N_0 in a BPSK system is

$$SNR(dB) = E_b/N_0(dB) - 10\lg(1/R). \qquad (26)$$

where R is Turbo code rate 1/6 in our scheme. In addition, the cost of soft metric calculation for the object function is acceptable since it grows linearly with the length of Turbo codes and the incremental operation of multiplication and addition other than necessary Turbo decoding is trivial. Also, our scheme will efficiently compute and update the two dimensional search of optimal timing offset estimates by the gradient descent method. The method is much reliable since it uses more reliable soft metrics other than hard decision in [4], [5]. And it has lower computation complexity than EM like method [8] and so on. Furthermore, the reasons that our scheme can work at low SNRs other than traditional timing recovery without channel code feedback lies in the fact that the estimates of the timing errors in our scheme are carried out by more reliable reference data of a whole Turbo frame which is much more reliable than just by several data symbols with unreliable blind estimation in traditional schemes. Therefore, our scheme has acceptable complexity with good performance and can be used in timing recovery especially at low SNRs.

V. Conclusion

We have presented a new iterative timing recovery algorithm for estimating large time phase and frequency offsets via maximizing the sum of the square of the soft decision metrics of Turbo decoding. And a new two-stage timing recovery scheme is also suggested where large timing phase and frequency offsets are acquired at first by our algorithm and then random walks and residual time errors of our algorithm are tracked by the M&M TED with Turbo decoding feedback. By our scheme, the Turbo coded BPSK communication system, suffered from large timing phase and frequency offsets over $\pm 0.5T$ and $\pm 1000ppm$ respectively at very low SNR (E_s/N_0) near -7.44 dB, can achieve good performance within 0.1 dB of the ideal code performance at rational cost.

Acknowledgment

This work was supported by National Natural Science Foundation of China(No.61001133, No.61101103, No.61102066), China Postdoctoral Science Foundation (No.2011M500999) and Open Project of Zhejiang Provincial Key Laboratory of Information Network Technology (No.201101).

References

[1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error correcting coding and decoding: Turbo codes ," in Proc. IEEE ICC'1993,pp.1064–1070.May 1993.

[2] P. Robertson, E. Villebrun and P. Hoehes, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in Proc. IEEE ICC'1995, pp. 1009–1013, June 1995.

[3] R. B. John, K. Aleksandar, W. M. Steven, N. Aravind and Z. Wei, "Iterative timing recovery," IEEE Sig. Proc. Mag., 21, pp. 89–102, Jan. 2004.

[4] D. U. Lee, L. V. Esteban, D. V. John and R. J. Christopher, "Joint LDPC decoding and timing recovery using code constraint feedback," IEEE Comm. Lett., 10(3), pp. 189–191, March 2006.

[5] E.L. Valles, R.D. Wesel, J.D. Villasenor and C.R. Jones, "Carrier and timing synchronization of BPSK via LDPC code feedback," in Proc. 40th Asilomar Conf. on Sig., Sys. and Comp., pp. 2177–2181, Nov. 2006.

[6] J. R. Bao, Y. F. Zhan and J. H. Lu, "Iterative timing recovery via soft decision metrics of low-density parity-check decoding, "IET Communications, 4(14), pp. 1742–1751, Sep. 2010.

[7] B. Mielczarek, A. Svensson, "Timing error recovery in turbo-coded systems on AWGN channels," IEEE Trans. on Comm., vol. 50, pp. 1584–1592, Oct. 2002.

[8] N. Noels, V. Lottici, A. Dejonghe, H. Moeneclaey, M. Luise and M. Vandendorpe, "A theoretical framework for soft-information-based synchronization in iterative (turbo) receivers," EURASIP Journal on Wireless Comm. and Net., pp. 117–129, 2005.

[9] S. Haykin, Adaptive filter theory, 3rd ed., Prentice Hall, Upper Saddle River, N.J.,1996.

[10] K. H. Mueller and M. Müller, "Timing recovery in digital synchronous data receivers," IEEE Trans. on Comm., vol.24, pp. 516–531, May 1976.

[11] J.G. Proakis, Digital communications, 5th ed., McGraw Hall, Upper Saddle River, N.J.,2009.

[12] CCSDS, "TM synchronization and channel coding–summary of concept and pationale," Report Concerning Space Data System Standards(CCSDS 130.1–G–1),Green Book, pp. 62–65, June 2006.