

Article

Efficient Iterative Timing Recovery of Low-Density Parity-Check Decoding Metrics Using the Steepest Descent Algorithm for Satellite Communications at Low SNRs

Yu Qiu ^{1,2} , Chao Liu ¹ , Jianrong Bao ^{1,3} , Bin Jiang ¹  and Yanhai Shang ^{1,*} 

¹ School of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, China; qiuyuvip@163.com (Y.Q.); liuchao@hdu.edu.cn (C.L.); baojr@hdu.edu.cn (J.B.); jiangbin@hdu.edu.cn (B.J.)

² Department of Technological Research and Development, Proheng High-Technology Co., Ltd., Hangzhou 310016, China

³ National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China

* Correspondence: yhshang@hdu.edu.cn

Abstract: An efficient iterative timing recovery via steepest descent of low-density parity-check (LDPC) decoding metrics is presented. In the proposed algorithm, a more accurate symbol timing synchronization is achieved at a low signal-to-noise (SNR) without any pilot symbol by maximizing the sum of the square of all soft metrics in LDPC decoding. The principle of the above-proposed algorithm is analyzed theoretically with the evolution trend of the probability mean of the soft LDPC decoding metrics by the Gaussian approximation. In addition, an efficiently approximate gradient descent algorithm is adopted to obtain excellent timing recovery with rather low complexity and global convergence. Finally, a complete timing recovery is accomplished where the proposed scheme performs fine timing capture, followed by a traditional Mueller–Müller (M&M) timing recovery, which acquires timing track. Using the proposed iterative timing recovery method, the simulation results indicate that the performance of the LDPC coded binary phase shift keying (BPSK) scheme with rather large timing errors is just within 0.1 dB of the ideal code performance at the cost of some rational computation and storage. Therefore, the proposed iterative timing recovery can be efficiently applied on occasions of the weak signal timing synchronization in satellite communications and so on.

Keywords: iterative timing recovery; timing synchronization; LDPC decoding; soft metrics; steepest descent algorithm



check for updates

Citation: Qiu, Y.; Liu, C.; Bao, J.; Jiang, B.; Shang, Y. Efficient Iterative Timing Recovery of Low-Density Parity-Check Decoding Metrics Using the Steepest Descent Algorithm for Satellite Communications at Low SNRs. *Electronics* **2021**, *10*, 3055. <https://doi.org/10.3390/electronics10243055>

Academic Editor: Mauro Tropea

Received: 12 November 2021

Accepted: 6 December 2021

Published: 8 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

LDPC codes can approach Shannon's capacity with moderate decoding complexity at low SNRs [1,2]. However, maximizing the full potential of the LDPC codes requires accurate timing recoveries in the baseband process of the receivers. Traditionally, timing recovery is performed independently from channel decoding. However, the metrics generated from channel decoding can generate more reliable information in iterative timing recovery [3]. The generated information can be applied to assist timing recovery in order to obtain accurate synchronization at very low SNRs. Traditional methods are usually ineffective at low SNRs. Iterative timing recovery uses the metrics from the codewords output of the channel decoding process [3]. A method with LDPC decoding hard decision metrics (i.e., code constraint feedback) to obtain good timing recovery has been proposed in [4]. However, it suffered from insufficient hard metrics, or unreliable satisfaction percentages of decoded codewords for the parity check equation for just few iterations [4,5]. Studies also suggested the use of timing recovery based on soft LDPC metrics with complex optimization calculations [6]. Other expectation maximization (EM)-based timing recovery methods for turbo or LDPC coding have also been suggested [7,8]. The EM-based timing

recovery method is implemented between the detection and estimation processes, where the expectation of symbols is calculated by a maximum a posteriori (MAP) decoder. Afterwards, the cost function is maximized to acquire excellent symbol timings. For instance, the EM algorithm has been applied in solving the time and frequency synchronization in a multiple-input multiple-output (MIMO) orthogonal frequency-division multiplexing (OFDM) system [9]. However, the function has highly complex EM steps. Several turbo decoding-aided timing recovery, with timing recoveries incorporated into iterative decoding, have been proposed for blind turbo detection of synchronization parameters [10]. Here, a turbo component decoder is applied in the synchronization parameters to perform blind joint synchronization detection and decoding with decoded metrics. In [11], synchronization is incorporated into Turbo decoding as the blind detection of synchronization parameters. It uses a turbo component decoder to perform blind synchronization detection and Turbo decoding by the decoding metrics. In [12], a new objective function is maximised to exploit the output turbo decoder's soft decision to improve synchronization iteratively. Thus, it is suited for short bursts of data transmissions at low SNRs. In [13], a Turbo soft metric based cost function is maximized to exploit the turbo soft decision and iteratively improve the synchronization. Thus, turbo-decoding metrics are calculated for obtaining good timing recoveries at low SNRs. However, turbo code-based methods are highly difficult due to complex turbo decoding. Because large sampling frequency offset can cause huge synchronization searching calculation and delay, a sampling frequency synchronization method to eliminate initial sampling frequency offset can also help reduce the overall synchronization time for the coarse symbol timing recovery [14]. So the communication receivers require the iterative timing recovery to acquire rather large timing frequency and phase offsets for good timing synchronizations. In addition, the iterative timing recovery can be performed by the means of MAP criterion as the iterative carrier synchronization but with much high complexity [15]. Therefore, an iterative timing recovery with high precision and low complexity is needed for using in space communications which usually work at low SNRs.

In this study, we mainly propose a new iterative timing recovery scheme by using the LDPC decoding metrics to correct timing errors for better timing precision and lower complexity. The timing recovery mechanism are modelled as an optimization problem of maximizing the sum of the squares of selected LDPC decoding metrics for precise estimation of timing offsets. Then, a theoretical verification is demonstrated analytically with closed expressions of the transition trend of the probability mean of the soft decoding metrics by the Gaussian approximation. In addition, an efficient numerical technique of approximate stochastic gradient descent is employed to solve the above optimization problem of excellent timing recovery with rational computations. Finally, a traditional M&M timing recovery algorithm [16] is adopted to rectify residual timing errors by using the LDPC code words as the anticipated data needed in timing track loops. Thus, it also saves some more bandwidth resources brought by trained data in traditional timing recovery schemes, which endows it more pragmatic in spectrum efficient satellite communications.

The remainder of the paper is organized as follows. In Section 2, the concept of the timing error model is introduced. Section 3 gives the principle of the suggested iterative timing recovery with soft LDPC decoding metrics. In this section, Gaussian approximation method is employed to explain the evolution trend of the soft metrics, which are influenced by the timing errors. The proposed iterative timing recovery scheme, accompanied by the traditional M&M timing track algorithm, is stated in Section 4 for the final timing synchronization scheme. In Section 5, 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 proposed scheme and evaluate the computation complexity. Finally, the conclusion is drawn in Section 6.

2. Timing Error System Model

Each transmitted symbol in a binary phase shift keying (BPSK) system is composed of N point impulse responses $g(mT)$ from a square root raised cosine (SRRC) pulse shaping filter with L times sampling rate, where T is the symbol interval. The signals are shaped as sequences (b_i) in the above filter, where $a_i \in \{\pm 1\}$ is the i -th modulated symbol and b_i is the sample symbol interpolated with $(L - 1)$ zeros between every two adjacent a_i to fit the sampling rate. τ_m is the timing offset to influence the m -th sample. Thus, the signal $s(mT)$ for transmission is given by

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

A received sequence with L -times up-sampling rate without timing offsets is expressed as

$$r(kT/L) = s(kT/L) + n(kT/L), \quad (2)$$

where $r(kT/L)$ is the received signal, $n(kT/L)$ is the additional white Gaussian noise (AWGN) from the channel, and T/L is the sampling interval. The assumed time reference for the m -th sample at the receiver when timing errors occur differs from the corresponding sample at the transmitter with timing offset τ_m . The m -th sampled signal after L times down-sampling is presented as

$$r(mT) = s(mT + \tau_m) + n(mT). \quad (3)$$

Finally, $r(mT)$ is obtained by replacing $s(\cdot)$ with the above expression. The resulting equation is presented as follows.

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

Constant timing frequency offset and random walk [4] can be combined in our model, because proper timing error models exist as constant timing phase offsets. The combined equation is presented as

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

where the initial time instance is offset by constant timing phase offset D , the timing error is disturbed by a zero mean Gaussian random variable (random walk) with variance σ_τ^2 denoted by $N(0, \sigma_\tau^2)$, and the constant timing frequency offset v is measured in parts per million (*ppm*). They should be estimated and compensated precisely for high transmission effect.

3. Influence of Timing Errors on LDPC Decoding

The received baseband signals are raised-cosine pulses from the combination of SRRC pulse-shaped filters located at the transmitters and receivers along with the AWGNs. In [6], the sampling positions approached the sampled points from LDPC decoding and corresponding analyses, which led to the increased absolute value of all LDPC decoding metrics, i.e., log-likelihood ratios (*LLRs*). A similar effect of the sum of all *LLR* squares under these timing errors also manifested. This phenomenon is explained below.

Given no noise, a received signal with perfect timing is properly sampled at the center of the symmetry raised cosine signal waveform and it has the most effective amplitude. However, under the overall timing offset as presented in Equation (5), the sampled signal deviates from the optimal center of the waveform with less effective amplitude and it is further deteriorate by the inter-symbol interference (ISI) by adjacent signals. At low SNR, the influence of the ISI can also be approximately modeled as the AWGN [10]. Therefore, the comprehensive effect of the symbol timing error causes a decreased effective SNR

for the coded BPSK modulation system. The equivalent noise variance is the sum of the original noise variance and the new generated equivalent noise variance caused by the ISI. With SNR analyses in [8], more deviations from optimal sampled points cause lower SNRs for the received signals, leading to a decrease of initial messages (i.e., *LLRs*) in LDPC decoding. The final output messages in the decoding are dependent on the square of the initial messages, because the messages in the iterations are mainly dependent on the square of previous messages. In addition, the sum of the square of each message increases the deviation effect. Thus, when timing offsets are introduced to reduce the effective SNRs, the amplitude of the initial messages decreases; in turn, this reduces the *LLRs* and decreases the sum of the square of all decoding messages. This process can be analyzed with density evolution and Gaussian approximation [17,18].

Firstly, we give a typical instance of the schematic diagram of the LDPC decoding by the iterative belief propagation (BP) [1,2] algorithm as shown in Figure 1.

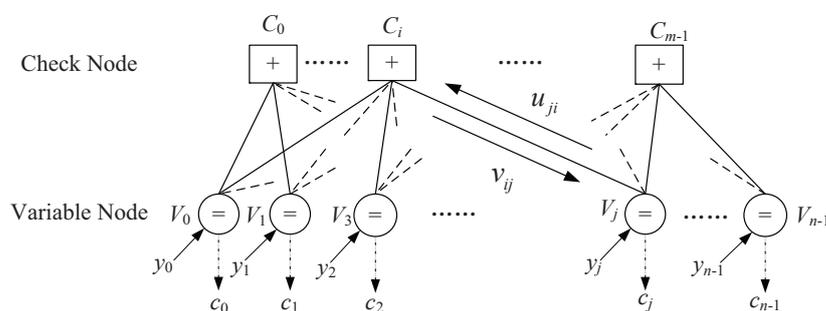


Figure 1. Schematic diagram of the LDPC soft message propagations between the variable nodes and check nodes in BP decoding.

From Figure 1, at an AWGN channel with zero mean and variance σ^2 , the logarithm version of the LDPC decoding can be expressed as below.

- (1) Initialization

The initial soft metrics of LDPC decoding, i.e., *LLRs*, are expressed as

$$L(c_i) = \ln \frac{p(x_i = 1|y_i)}{p(x_i = -1|y_i)} = \frac{2y_i}{\sigma^2}, \tag{6}$$

$$L(u_{ij}) = L(v_{ji}) = 0, \tag{7}$$

where $L(\cdot)$ is the logarithm likelihood ratio (*LLR*) function and $L(c_i)$ is the initial *LLR* of the codeword c_i . x_i and y_i are the i -th decoded and received signals respectively. u_{ij} and v_{ji} are the exchanged messages between the variable node j and the check node i .

- (2) Decoding procedure

- (2.1) Update the variable nodes and perform the final decoding judgment

The decoding metrics updated in the variable nodes and final decoding judgment can be expressed as

$$L(u_{ij}) = L(c_i) + \sum_{j' \in M(i) \setminus j} L(v_{j'i}), \tag{8}$$

$$L(Q_i) = L(c_i) + \sum_{j \in M(i)} L(v_{ji}), \tag{9}$$

$$\hat{c}_i = \begin{cases} 0, & L(Q_i) \geq 0 \\ 1, & L(Q_i) < 0 \end{cases}, \tag{10}$$

where the set of the i -th check node connected to the j -th variable node is expressed as $M(i) = \{j : \mathbf{H}_{j,i} = 1\}$ and $\mathbf{H}_{j,i}$ is the element of the check matrix \mathbf{H} with index (i, j) . The symbol “ $\setminus j$ ” behind a set means the set with removal of the

element “ j ”. Q_i is the LDPC soft decision metrics of the i -th variable node of the codeword c_i . N is the code length. $\mathbf{C} = [c_1, \dots, c_N]^T$ is the codeword vector and the estimate of this vector is $\hat{\mathbf{C}} = [\hat{c}_1, \dots, \hat{c}_N]^T$. In this step, if $\mathbf{H} \cdot \hat{\mathbf{C}} = \mathbf{0}(\text{Mod}2)$, $\hat{\mathbf{C}}$ is the decoding result. If $\hat{\mathbf{C}}$ is the result or the iteration exceeds the maximum times, the iteration is finished. Otherwise, go to next step (2.2).

- (2.2) Update the check nodes and then go to step (2.1)
The decoding metrics updated in the check nodes can be represented as

$$L(v_{ji}) = 2 \tanh^{-1} \left\{ \prod_{i' \in K(j) \setminus i} \tanh[L(u_{i'j})/2] \right\}, \tag{11}$$

and then go to step (2.1) for subsequent processing.

In Equation (11), the set of the j -th variable node connected to the i -th check node is $K(j) = \{i : \mathbf{H}_{i,j} = 1\}$.

Then, the messages from the j -th variable node and the i -th check node in LDPC decoding (given as $u - ji$ and v_{ij} , respectively) are symmetrical with $\sigma_m^2 = 2m$, where σ_m^2 and m are the variance and mean value of the messages, respectively [17]. The mean value of all messages for variables and check nodes (m_u and m_v) are updated in the iterations. The mean values represent the probability distribution of the messages. The mean value of the messages $m_v^{(k)}$ from the variable nodes to the check nodes in k -th iterations is presented as

$$m_v^{(k)} = m_{u0} + (d_v - 1)m_u^{(k-1)}, \tag{12}$$

where $m_u^{(k)}$ is the mean value of messages from all check nodes and variable nodes v after $k - 1$ iterations, m_{u0} is the mean value of the initial message $L(c_i)$, and d_v is the variable node degree. The indexes of u_{ji} and v_{ij} can be omitted for simplification, because they are independent and identically distributed (*i.i.d.*). Moreover, $m_u^{(k-1)}$ is assigned to a zero mean value, because initial messages from all check nodes are zero. The updated messages coming from check nodes to variable nodes are presented similar to [6] as

$$F[\tanh(u^{(k)}/2)] = F[\tanh(v^{(k)}/2)]^{(d_c-1)}, \tag{13}$$

where the indexes u_{ji} and v_{ij} are omitted because they are *i.i.d.*, and d_c is the check node degree. $F[\tanh(u/2)]$ is presented as

$$F[\tanh(u/2)] = \frac{1}{\sqrt{4\pi m_u}} \int_{\mathcal{R}} \tanh(u/2) e^{-(u-m_u)^2/4m_u} du, \tag{14}$$

where “ \mathcal{R} ” represents that the integral interval falls in the entire field of real number.

Moreover, a new function $\phi(x)$ is defined in [17] as

$$\phi(x) = \begin{cases} 1 - \frac{1}{\sqrt{4\pi x}} \int_{\mathcal{R}} \tanh(u/2) e^{-(u-x)^2/4x} du, & x > 0 \\ 1, & x = 0 \end{cases}. \tag{15}$$

Equation (15) is continuous and monotonically decreases on $[0, +\infty)$ with $(\phi(0) = 1, \phi(+\infty) = 0)$. The mean value update of all messages from the check nodes to the variable nodes is given by

$$m_u^{(k)} = \phi^{-1} \{ 1 - [1 - \phi(m_{u0} + (d_v - 1)m_u^{(k-1)})]^{d_c-1} \}, \tag{16}$$

where $m_u^{(0)}$ is zero, which is the initial value of m_u [18].

The output messages from the variable nodes $m_o^{(k)}$ after k iterations are represented as

$$m_o^{(k)} = m_{u0} + d_v m_u^{(k-1)}. \tag{17}$$

From Equation (17), $m_o^{(k)}$ is mainly decided by initial message m_{u0} and message $m_u^{(k-1)}$ after k iterations. In addition, $m_u^{(k)}$ can be recursively calculated by Equation (16) and $m_u^{(k-1)}$ with an initial value of $m_u^{(0)} = 0$. Thus, $m_o^{(k)}$ relies on the mean value of initial message m_{u0} . The properties of $\phi(x)$ indicate that a larger m_{u0} leads to similarly larger values of $m_u^{(k)}$ and $m_o^{(k)}$ in Equations (16) and (17), respectively; thus, $m_o^{(k)}$ is mainly decided by m_{u0} . In addition, a larger m_o leads to a larger $m_o^{(k)}$ and sum of all message squares. It can be explained as follows.

For the same LDPC codeword, the main code parameters, i.e., the degree distribution, are the same. So the variable node degree d_v and check node degree d_c are equal for a specific node message processing under different channel state information, or the initial LLR represented in Equation (6). From Equation (15), the function $\phi(x)$ is continuous and monotonically decrease on its definition domain $[0, +\infty)$ with $(\phi(0) = 1, \phi(+\infty) = 0)$. Then, due to the relationship that a function and its inverse function is just mirror symmetrical about the function $y = f(x) = x$, the inversion function $\phi^{-1}(y)$ is also decrease on $(0, +1]$ with $(\phi(0) = +\infty, \phi(1) = 0)$. So the first recursive iteration of the messages from the check nodes to the variable nodes in Equation (16) is given as

$$m_u^{(1)} = \phi^{-1}\{1 - [1 - \phi(m_{u0})]^{d_c-1}\}, \tag{18}$$

where m_{u0} is the initial message and it is got from Equation (6). Obviously, due to the monotonically decrease property of $\phi(\cdot)$ and $\phi^{-1}(\cdot)$, larger m_{u0} causes the larger $m_u^{(1)}$. For the second recursive iterations, Equation (16) is expressed as

$$m_u^{(2)} = \phi^{-1}\{1 - [1 - \phi(m_{u0} + (d_v - 1)m_u^{(1)})]^{d_c-1}\}. \tag{19}$$

From Equation (19), other parameters being the same, the larger initial message m_{u0} along with the corresponding larger previous generated message $m_u^{(1)}$ leads to a much larger $m_u^{(2)}$ than that generated from a smaller initial message. The rest can be done in the same manner and in the higher order of recursive iterations, larger initial message m_{u0} surely causes larger solution result $m_u^{(k)}$ after k recursive iterations.

Therefore, the conclusion is true for an LDPC-coded system with an all-0-bit input. A symmetry condition ($f(x) = f(-x)e^x$) is used to preserve the above conclusion for all messages, because initial negative messages can be processed for positive ones as follows [18]. First, the channel symmetries are given by

$$p(y_i = q|x_i = 1) = p(y_i = -q|x_i = -1). \tag{20}$$

The sign factor of the check node message map $\phi_c^{(k)}$ is given by:

$$\phi_c^{(k)}(b_1 m_1, \dots, b_{d_c-1} m_{d_c-1}) = \phi_c^{(k)}(m_1, \dots, m_{d_c-1}) \prod_{i=1}^{d_c-1} b_i, \tag{21}$$

where b_i belongs to $\{\pm 1\}$ sequence.

The most important sign inversion invariance property of the variable node message map $\phi_v^{(k)}$ is expressed as:

$$\phi_v^{(k)}(-m_0, \dots, -m_{d_v-1}) = -\phi_v^{(k)}(m_0, \dots, m_{d_v-1}), k \geq 1. \tag{22}$$

The final output message $m_o^{(k)}$ is dependent on the square of initial messages based on Equations (12)–(22). The non-0-input messages, which correspond to the minus data in the iterations, are also dependent on the square of message m_i based on the symmetry properties shown in Equations (20)–(22). Thus, larger squares of initial legal LDPC codeword messages lead to larger squares of final output messages. The sum of the above equations amplifies this effect. Initial message amplitudes (i.e., LLRs) decrease when timing offsets are introduced to reduce the effectiveness of SNR. The decrease reduces the cost function, i.e., the sum of all the square of the LLRs. Thus, a timing offset estimate (including timing

phase and frequency offsets) is obtained by searching possible ranges in obtaining the maximum cost function. Moreover, random walk has limited destructive effects, because it is limited by oscillation quality. Therefore, optimal timing offsets are estimated and corrected by maximizing the sum of all the square of the *LLRs*.

4. Proposed Iterative Timing Recovery

The *i*-th message, (i.e., *LLR_i*) in LDPC decoding [2] is presented as

$$LLR_i = \log \Lambda(x_i|\mathbf{r}) + E(x_i|x_{j,j \neq i}), \quad (23)$$

$$\Lambda(x_i|\mathbf{r}) = \log \frac{p(x_i = 1|\mathbf{r})}{p(x_i = -1|\mathbf{r})} = \frac{2y_i}{\sigma_{ch}^2}, \quad (24)$$

where \mathbf{r} is the received signal vector with element y_i , σ_{ch}^2 is the variance of channel noise, x_i is the decision of the *i*-th element of \mathbf{r} , $p(x_i = 1|\mathbf{r})$ or $p(x_i = -1|\mathbf{r})$ is the posterior probability of the decision of the received signal \mathbf{r} , and $E(x_i|x_{j,j \neq i})$ is the extrinsic information brought by other variable nodes with relation to parity check constraint. $E(x_i|x_{j,j \neq i})$ is updated for each LDPC decoding iteration from the parity check nodes. Correct *LLRs* are crucial in ensuring proper LDPC decoding. Based on the analyses and conclusions presented in Section 3, an efficient timing recovery can be developed as follows. First, an object function $\Psi(\tau, v)$ related to the timing phase and frequency offsets, τ and v (measured in *ppm*), is defined as

$$\Psi(\tau, v) = \sum_{i=0}^K [\Lambda(x_i|\mathbf{r}, \tau, v) + E(x_i|x_{j,j \neq i}, \tau, v)]^2, \quad (25)$$

where Λ_i is $\Lambda(x_i|\mathbf{r}, \tau, v)$, and K is the LDPC codeword length. Thus, the optimal $((\tau, v))$ can be iteratively searched and updated by solving the optimization function given by

$$(\tau_{opt}, v_{opt}) = \arg \max_{\tau \in [-T/2, T/2], v \in \text{Freq.Region}} \Psi(\tau, v). \quad (26)$$

To illustrate the objective function $\Psi(\tau, v)$, a curve of it with respect to (τ, v) after 3 LDPC iterations is calculated and shown in Figure 2, where the LDPC code (1944, 972) in Draft IEEE 802.11n [19] is selected at E_b/N_0 of 0.5 dB. Based on the above numeric simulations, the object function $\Psi(\tau, v)$ in Equation (25) is symmetrical and approximately convex to its variable, i.e., (τ, v) . It indicates that the function $\Psi(\tau, v)$ has a maximum value located at the position without any timing offset. Other cases are the same under different LDPC codes and E_b/N_0 , where one global optimum value and some local ones are featured in the plots. Therefore, this being the case, the optimal estimation of (τ, v) can be obtained by maximising the function $\Psi(\tau, v)$.

Thus, gradient descent algorithm (steepest descent) can be applied in solving Equation (26) similar to [13]. Afterwards, a two-dimensional gradient descent algorithm is carried out to iteratively search and update (τ, v) for Equation (26). An approximate gradient descent algorithm is used to implement the iterative timing recovery, because the object function $\Psi(\tau, v)$ is discontinuous without direct analytic expressions. The iterative timing recovery process is discussed below.

Firstly, $E(x_i|x_{j,j \neq i}, \tau, v)$ in Equation (25) is considered constant to (τ, v) . $E(x_i|x_{j,j \neq i}, \tau, v)$ and (τ, v) are set to zero for the 1st iteration (i.e., no message from the check nodes to the variable nodes) and hide in the change of $\Lambda(x_i|\mathbf{r}, \tau, v)$ in other iterations of the subsequent derivations. Thus, $E(x_i|x_{j,j \neq i}, \tau, v)$ can be omitted in approximate derivations of $\Psi(\tau, v)$ to (τ, v) . The omission is calculated as

$$\frac{\partial \Psi(\tau, v)}{\partial \tau} = 2 \sum_{i=0}^K [\Lambda_i + E(x_i|x_{j,j \neq i}, \tau, v)] \frac{\partial \Lambda_i}{\partial \tau}, \quad (27)$$

$$\frac{\partial \Psi(\tau, v)}{\partial v} = 2 \sum_{i=0}^K [\Lambda_i + E(x_i | x_{j,j \neq i}, \tau, v)] \frac{\partial \Lambda_i}{\partial v}. \tag{28}$$

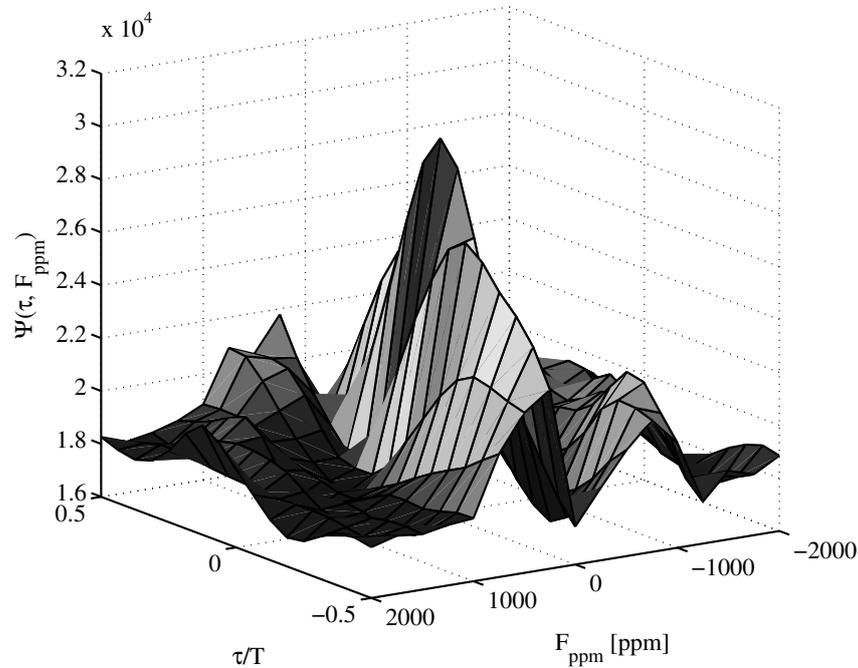


Figure 2. Typical plot of $\Psi(\tau, v)$ (τ, v) with LDPC code (1944, 972) by 3 decoding iterations.

The received signal Λ_i , or mainly the variable y_i , is affected by the timing offset parameters (τ, v) based on Equations (23) and (24). So the approximate derivation of $\Psi(\tau, v)$ to (τ, v) can be represented as

$$\frac{\partial \Psi(\tau, v)}{\partial \tau} = 2 \sum_{i=0}^K [\Lambda_i + E(x_i | x_{j,j \neq i}, \tau, v)] \frac{\partial y_i}{\partial \tau} \cdot \frac{1}{2\sigma^2}, \tag{29}$$

$$\frac{\partial \Psi(\tau, v)}{\partial v} = 2 \sum_{i=0}^K [\Lambda_i + E(x_i | x_{j,j \neq i}, \tau, v)] \frac{\partial y_i}{\partial v} \cdot \frac{1}{2\sigma^2}. \tag{30}$$

By similar techniques in [13], given normalized parameters, the received signal y_i can be approximately modeled as follows:

$$y_i = 1 - c_1 \text{sgn}(\tau)\tau - c_2 i \text{sgn}(v)v + n(t_i), \tag{31}$$

where the amplitudes of the transmitted signals are normalized as “1”, $n(t_i)$ is the channel noise at that time instance, and “sgn” is the sign function in mathematics. Here, the approximate transition trend of $\Psi(\tau, v)$ to (τ, v) is used to incorporate a more accurate value into positive constant c_1 and c_2 , which can be determined in simulations. Equations (29) and (30) is then approximately calculated as

$$\frac{\partial \Psi(\tau, v)}{\partial \tau} \approx 2 \left(\frac{2}{\sigma_{ch}^2}\right) (-c_1) \sum_{i=0}^K [\Lambda_i + E(x_i | x_{j,j \neq i}, \tau, v)] \text{sgn}(\tau), \tag{32}$$

$$\frac{\partial \Psi(\tau, v)}{\partial v} \approx 2 \left(\frac{2}{\sigma_{ch}^2}\right) (-c_2) \sum_{i=0}^K [\Lambda_i + E(x_i | x_{j,j \neq i}, \tau, v)] \text{sgn}(v) \cdot i. \tag{33}$$

Equations (32) and (33) can be further simplified by uniting all fixed variables or constants into the negative value C_1 and C_2 , respectively, as

$$\frac{\partial \Psi(\tau, v)}{\partial \tau} \approx C_1 \sum_{i=0}^K [\Lambda_i + E(x_i | x_{j,j \neq i}, \tau, v)] \text{sgn}(\tau), \tag{34}$$

$$\frac{\partial \Psi(\tau, v)}{\partial v} \approx C_2 \sum_{i=0}^K [\Lambda_i + E(x_i | x_{j,j \neq i}, \tau, v)] \text{sgn}(v) \cdot i. \tag{35}$$

Thus, the calculation complexity can be reduced by Equations (34) and (35), because we use the adaptive parameter update algorithm, which just needs the approximate trend of the gradient rather than the more accurate numerical results. So the adaptive timing parameter correction along with the iterative LDPC decoding jointly contribute the good features of both good performance and low complexity.

Finally, according to the principles of the adaptive algorithms, (τ, v) can be estimated and updated with the following Equations (36) and (37) as

$$\tau^{(n+1)} = \tau^{(n)} - \lambda_1 [\partial \Psi(\tau, v) / \partial \tau]^{(n)}, \tag{36}$$

$$v^{(n+1)} = v^{(n)} - \lambda_2 [\partial \Psi(\tau, v) / \partial v]^{(n)}, \tag{37}$$

where λ_1 and λ_2 are step parameters for iteratively updating (τ, v) , respectively, y_i in Equation (31) decodes LDPC codes, and the superscript (n) demotes the (n) -th iteration. The partial derivation function $\partial \Psi(\tau, v) / \partial \tau$ and $\partial \Psi(\tau, v) / \partial v$ are computed in Equations (34) and (35). The initial parameter pair $(\tau^{(0)}, v^{(0)})$ can be set as the zero vector $(0, 0)$. The other constants can be verified in the simulations.

In addition, a conventional first-order PLL-based structure with a decision-directed Mueller–Müller timing error detector (M&M TED) [6] can also be used as a supplement like [4] to implement the iterative timing tracking. At each iteration, the M&M TED is executed by provided with more accurate symbols decoded by the LDPC decoder and the re-sampled received signals, which are compensated with the optimal estimate of timing phase and frequency offset by our proposed algorithm. Finally, according to the above analyses, the whole timing recovery is designed in Figure 3.

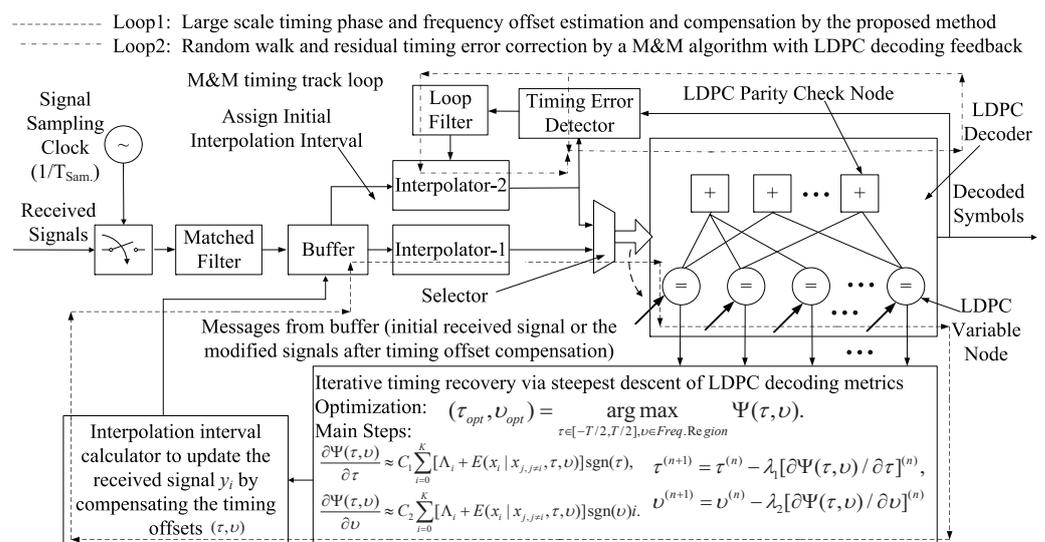


Figure 3. Block diagram of the whole timing recovery system by the proposed scheme via steepest descent of LDPC decoding metrics and the M&M timing track.

In the whole timing recovery system shown in Figure 3, there are two loops to execute the timing recovery, i.e., Loop1 and Loop2. The Loop1 is performed at first by our proposed

iterative timing recovery via steepest descent of LDPC decoding metrics to obtain large scale timing phase and frequency offset estimations, which are used to compensate the error of the received signal y_i by Equation (31) stored in the buffer in Figure 3. The main steps of the procedure are calculated by Equations (34)–(37) to obtain the final optimized goal denoted in Equation (26). So this loop mainly performs the timing capture, especially at low SNRs. After Loop1, the selector switches the working mode to Loop2. Then, random walk and residual timing errors are corrected by a M&M algorithm with the reference LDPC decoding feedback, which works as the timing track.

According to the above iterative timing recovery scheme, the computational complexity can be analyzed as follows. Here, we just calculate the additional computations other than the compulsory LDPC decoding. It is assumed that L iterations of timing offset estimation and compensation in the length- K LDPC code are required for the satisfied symbol timing recovery. From Equations (34) and (35), there are K additions in the top equation and K additions and K multiplications in the bottom equation, which sum up as $2K$ additions and K multiplications. The coefficients of C_1 and C_2 in Equations (34) and (35) are empirical parameters and they can be incorporated into the step size of λ_1 and λ_2 in Equations (36) and (37). So the calculation of them can be neglected. Then, there are two multiplications and two subtractions (equal to additions in computational complexity) in the update of timing offset estimates in Equations (36) and (37). The step size parameters are decided by the practical occasions, which can be neglected, when compared with the above calculation. Moreover, there are three multiplications and three additions in the update of y_i in Equation (31), plus 1 division (equal to multiplication) in the calculation of Λ_i , i.e., $L(c_i)$, in Equation (6). Therefore, the total computational complexity, except from the necessary LDPC decoding, are approximate $(2K + 5)L$ additions and $(K+6)L$ multiplications. By later simulations, since the required iterations of the timing recovery are limited, i.e., $L \leq 6$, the total computational complexity is acceptable in practice.

5. Simulation Results and Analyses

To verify the effectiveness of the proposed scheme, numerical simulations of a LDPC coded BPSK system in an AWGN channel are carried out in this section. The LDPC code (1944, 972) [19] is adopted in the simulations with a maximum of 20 decoding iterations. Generally, the performance is often weighed by bit-error-rate (BER). So Monte Carlo simulations of LDPC decoding will run until either a specified number (e.g., 10 in high SNR and 50 in low SNR) of error frames occur or a total 1 million trials have been run. We choose the above simulation parameters of iteration stop judgment for LDPC decoding by compromising the result reliability and calculation efficiency and also the more practical experiments in some powerful work stations. The SRRC shaped and matched filters both have a roll-off factor of 0.3 and 13 taps. Several timing offset estimation iterations are used. The timing phase and frequency offsets are set over $\pm 0.5 T$ (e.g., $0.3 T$ and T is the sampling period) and ± 500 ppm (e.g., 300 ppm). The parameter σ/T of the random walk is 0.5%, which is similar to that used in [5]. The step size parameters λ_1 and λ_2 in Equations (36) and (37) are chosen as 0.2 and 0.5, respectively. Finally, the parameter analyses and the BER performances by the proposed iterative timing recovery are simulated and shown as follows.

The partial curves of $\Psi(\tau, v)$ to (τ, v) , and the $\Psi(\tau, 0)$ to τ and $\Psi(0, v)$ to v , with two, four, and six iterations at E_b/N_0 of 1.5 dB are shown in Figures 4 and 5, respectively. The timing phase offsets within $0.1 T$ occur near the optimal position (Figure 4). The closeness to the optimal position is mainly due to the property of the raised cosine waveform by the SRRC-shaped and matched filters, where τ within $0.1 T$ slightly reduces the effective amplitude of the received signals. Furthermore, the influence of v on $\Psi(\tau, v)$ changes dramatically (Figure 5). The largest frequency error discrimination occurs within 20 of the optimal point. Thus, approximately 6 iterations are enough for the timing recovery to distinguish the ideal sampling point. Further from this simulation, the effective search ranges for the timing phase and frequency offsets are chosen as $(-0.5 T, 0.5 T)$ and

(−500 ppm, 500 ppm). This may be the deficiency of the proposed timing recovery, where traditional timing recovery should be performed at first to guarantee the residual timing offsets fall into the range of this requirement. However, the proposed timing recovery can obtain much better estimation precision.

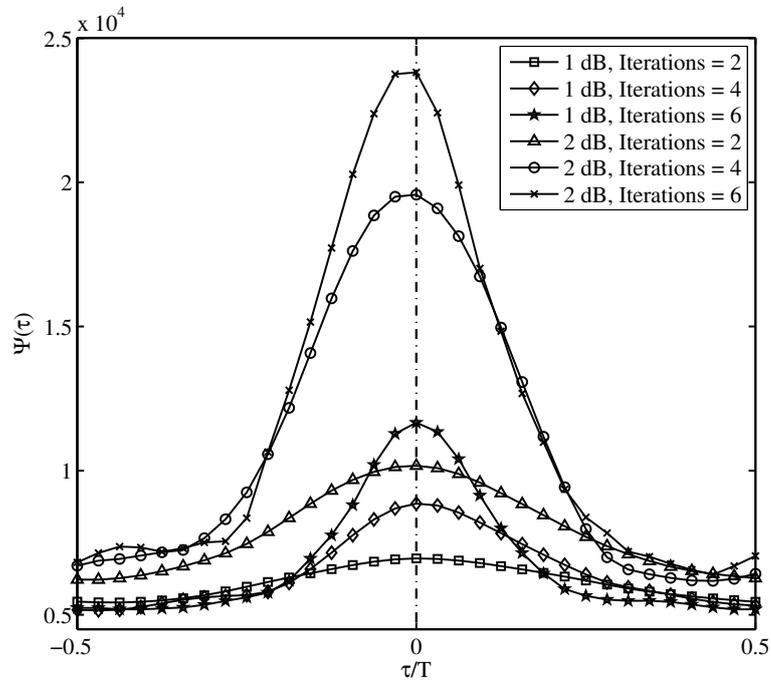


Figure 4. Partial curves of $\Psi(\tau, 0)$ to τ with two, four, and six iterations.

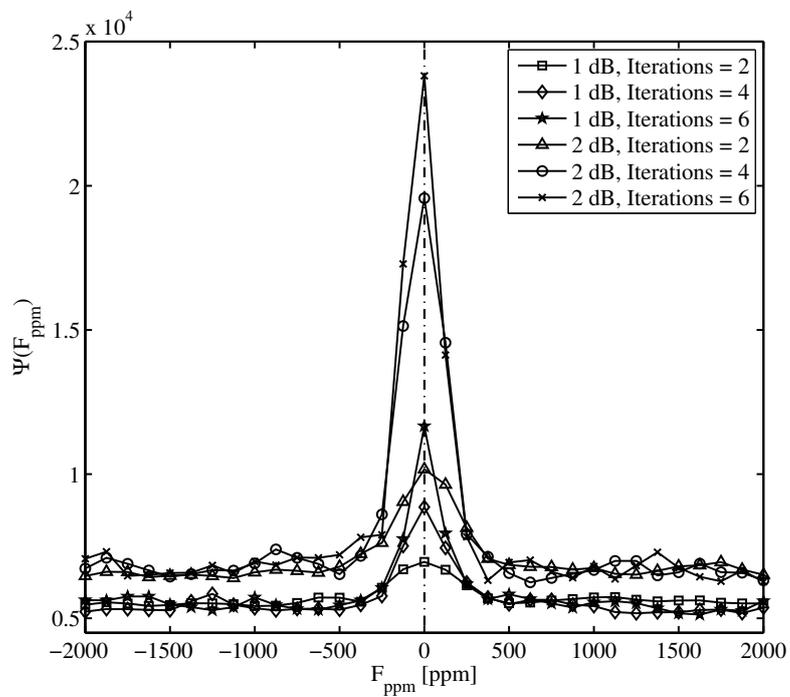


Figure 5. Partial curves of $\Psi(0, \nu)$ to ν with two, four, and six iterations.

Here, we define the normalized mean square error (NMSE) simulations for (τ, ν) as $NMSE_\tau$ and $NMSE_\nu$ in Equations (38) and (39) as

$$NMSE_{\tau} = [(\tau - \tilde{\tau})/\tau]^2, \tag{38}$$

$$NMSE_v = [(v - \tilde{v})/v]^2, \tag{39}$$

where (τ, v) are real timing phases and frequency offsets, whereas $(\tilde{\tau}, \tilde{v})$ are their estimates. Moreover, the estimated timing error parameter pair $(\tilde{\tau}, \tilde{v})$ involves an average of 50 times calculation. Meanwhile, the true timing error parameter pair (τ, v) for test cannot be the vector with any element zero to make sense of Equations (38) and (39) avoiding from divided by zero. Then, the $NMSE_{\tau}$ and the $NMSE_v$ with four, six, and eight iterations at different SNRs are simulated and shown in Figure 6 as follows.

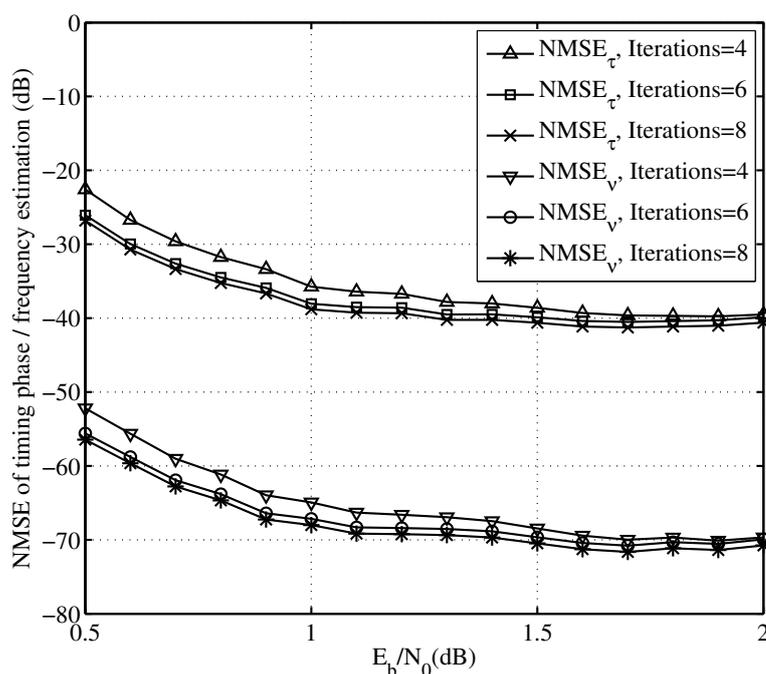


Figure 6. NMSE of the timing phase and frequency estimation with an average of 50 times for each E_b/N_0 .

The $NMSE_v$ in Figure 6 is smaller than $NMSE_{\tau}$, because the influence of v on object function $\Psi(\cdot)$ exerts more control than τ (Figure 6).

Therefore, a total of six iterations are enough for timing recovery (Figure 6). In our scheme, 6 timing recovery iterations are simulated and compared with the ideal code performance. The bit or frame error ratio (BER/FER) of the scheme with the above parameters is shown in Figure 7.

The performance of the LDPC coded system, which suffered from large timing errors, is within 0.1 dB of the ideal code performance (Figure 7). The LDPC coded system performs well at low SNRs (i.e., below 1 dB) without any pilot symbols. The cost is trivial, because the complexity linearly grows with the LDPC code length. In addition, incremental operations are limited.

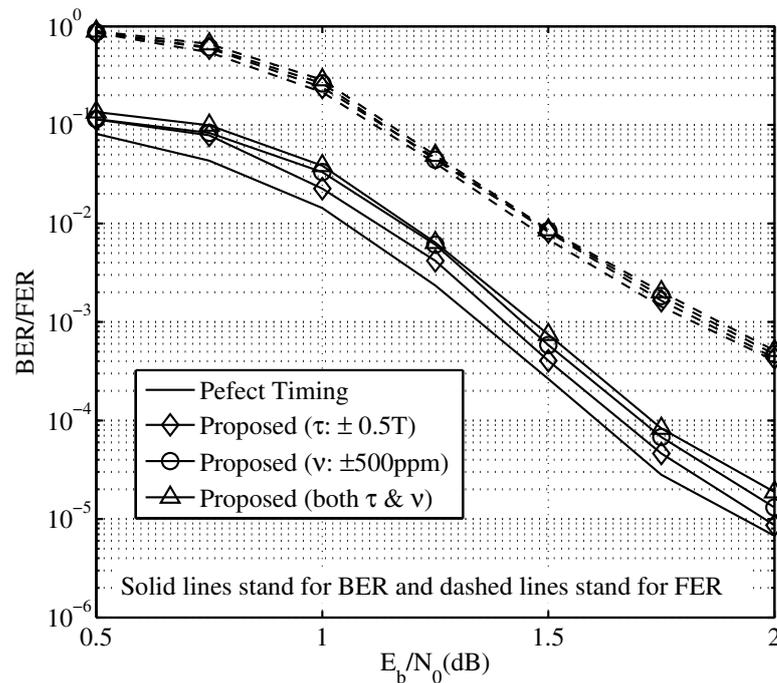


Figure 7. BER/FER of the proposed method with 6 iterations. The LDPC (1944, 972) decoding is performed with maximum iterations of 20.

Finally, our algorithm is more stable than the algorithm in [4] because our algorithm employs soft decisions, whereas [4] uses hard decisions as code constraint feedback. Hard decisions yield false judgments when even decoded bits are reversed falsely, which involves computing the same parity check equation. In comparison, no similar case of false judgment exists in our algorithm. Therefore, our scheme shows rational complexity and excellent performance similar to [4]. However, our algorithm is more stable in timing recoveries especially at low SNRs.

The complexity increment for the timing recovery scheme can be analysed in the comparison between the original scheme with traditional timing recovery and LDPC decoding, and the proposed scheme with the iterative timing recovery, the residual timing recovery and LDPC decoding. The traditional scheme performs the timing recovery and LDPC decoding separately. It has one timing tracking calculation for each sample and several LDPC decoding iterations. However, under the same condition and LDPC decoding iterations, our scheme has the same LDPC decoding computations and timing tracking calculation, including some extra calculations of iterative timing recovery. By Equations (34)–(37), and given iteration number N_i , there are about $N_i \cdot [2(2K - 1) + 2]$ additions and $N_i \cdot (K + 2)$ multiplications of additional computations in our scheme. Actually, with the gradual correction of timing errors by the proposed iterative timing scheme, the equivalent SNR is improved too. So the iteration number of LDPC decoding can be significantly reduced by sharp reduction of the decoding iterations. So it also magnificently cut down the computational complexity for the whole iterative system. However, it is difficult to illustrate this procedure analytically due to the nature of iterative process, which is closely related to the initial input signals and nonlinear iterative calculation. So the computational complexity cost of the proposed scheme is determined by initial input parameters and precision required in the joint decoding and demodulation. Thus, it can only be evaluated by numerical methods dependent on random input modulated signals.

In our former work [6], it employs similar metrics from LDPC decoding to achieve timing synchronization at low signal-to-noise ratios. It obtains the accurate timing acquisition with a search window based Nelder-Mead simplex algorithm [20] to maximize the sum of the absolute values of the metrics. It obtained a little performance gain (≤ 0.05 dB) than our

new proposed algorithm. However, it may encounter the problems of local convergence of the optimization searching algorithm under limited iterations. It had to add some more additional decision module to judge and overcome the convergence problem. In our proposed method, it can achieve the same performance and overcome the deficiency of local convergence of the suggested optimization brought by the iterative BP decoding. It can be explained as follows. Due to the analyses of the influence of timing offsets on LDPC decoding, the effect of the timing error can be modeled as the decrease of SNR in the decoding procedure. The actual SNR can be the synthesis of the real AWGN noise and the equivalent AWGN noise caused by the inter symbol interference (ISI) due to the false symbol sampling. It can be improved via our proposed method in the several timing parameter updates in Equations (36) and (37). As long as the LDPC decoding is convergent within the range of decoding threshold, the proposed iterative timing recovery is surely to be globally convergent too, thus guarantee the effectiveness of the proposed timing recovery. In addition, it can also turn to the help of an additional Mueller–Müller algorithm [16] for better symbol synchronization, where it plays a similar role of timing synchronization tracking. Furthermore, the proposed algorithm can be further simplified by accumulating the metrics from LDPC variable nodes with high degrees, which holds reliable metrics in the LDPC decoding. It has been verified by numerical simulations with the similar method, except that the calculation of Equation (25) just includes the more reliable variable nodes of high degree. Because the results are exactly similar for the further simplified algorithm with reduced variable node calculation, we will not repeat the redundant result here for the purpose of saving space. In addition, the portion of the low degree variable nodes, that do not take part in the accumulation of Equation (25), should be less than 20% via several simulations. Otherwise, there will be too many fluctuations to affect the approximately convex of the objective function $\Psi(\tau, \nu)$ shown in Figure 2, which greatly affect the effectiveness of the proposed iterative timing recovery. Suppose the portion of the high degree variable nodes to the whole code length is p , the whole complexity can be further reduced to the p of the proposed algorithm.

In our work, we mainly discuss the least complex BPSK modulation scheme. We mainly focus on the development of iterative timing recovery itself, so we just use it to illustrate our idea. Actually, long distances in satellite communications and the corresponding low signal-to-noise rate (SNR) transmission environments require power-efficient low-order modulation, such as BPSK, quadrature phase shift keying (QPSK) and some variations of it (e.g., offset quadrature phase shift keying (OQPSK), $\pi/4$ -differential quadrature phase shift keying ($\pi/4$ -DQPSK), etc.). Our scheme can be simply modified to satisfy these modulations, since the QPSK modulation series are actually the orthogonal implementation of two group of BPSK, and each branch of BPSK can adopt our proposed timing recovery scheme to accomplish the timing synchronization. The timing recovery scheme can be further improved with the well-known Costas loops for good implementation in practice. In these schemes, we obtain the metrics from the modulated signals in an orthogonal manner to feed the LDPC decoder, and the timing recovery parameters are iteratively calculated in LDPC decoding similar to those in our scheme. In addition, we also verify the roughly convex of the sum of the squares of selected LDPC decoding metrics with QPSK modulation similar as that shown in Figure 2. The efficient approximate stochastic gradient descent algorithm is also feasible. The complexity growth is still acceptable since only two branch of BPSK synchronization loops are required in the QPSK related schemes other than one branch in our scheme. The complexity increment is rather trivial. For more complex modulations, such as the Mary-quadrature amplitude modulation (M -QAM), Mary-phase shift keying (M -PSK) and even the Mary-amplitude phase shift keying (M -APSK) modulations, they can also be divided as the In-phase and Quadrature-phase mode of amplitude-shift-keying (ASK) modulations to properly implement the proposed timing recovery scheme. The complexities mainly lie in the metric extraction from the high-order modulations. They can be the further works in this aspect of research for low complex approach. Finally, due to the similar main mechanism of soft LDPC metric usage in timing

recovery, we do not expand these contents more for purpose of concise manuscript. However, they can be simply derived and constructed for practical applications in satellite and wireless communications.

6. Conclusions

In this paper, we presented a new iterative timing recovery method using the maximization of the sum of the LDPC decoding metric squares. The main contributions are summarized as follows. First, the relationship between LDPC decoding metrics and timing offsets is analyzed for the purpose of timing recovery. The introduce of the timing error can be explained as the phenomenon of the decrease of effective SNR. Afterwards, an approximate steepest gradient descent-based algorithm is proposed to maximize the sum of the square of LDPC decoding metrics efficiently, resulting in timing recoveries with reasonable complexity. The complexity of the algorithm is also simplified with limited iterations. In addition, the principle of the proposed algorithm is theoretically analyzed with the transition trend of the probability mean of the soft LDPC decoding metrics by the Gaussian approximation, which can be effectively used to explain the mechanism of the proposed timing recovery method. Finally, the complete timing recovery is accomplished where the proposed scheme is combined with a Mueller–Müller (M&M) timing recovery algorithm to perform both the timing capture and the timing track. The simulation results indicate that the LDPC-coded BPSK system suffered from a large timing phase and frequency offset of over ± 0.5 T and ± 500 ppm, respectively, and performs excellently within 0.1 dB of the ideal code performance with rational complexity. The global convergence of the timing recovery can be guaranteed by the natural of BP decoding, since the proposed method is completely merged into the decoding.

Author Contributions: Conceptualization: Y.Q., Y.S.; methodology: Y.Q., J.B.; software: Y.Q., C.L.; formal analysis: B.J.; investigation: Y.Q., J.B., C.L.; resources: Y.Q., B.J.; data curation: J.B., C.L., B.J.; writing—original draft preparation: Y.Q.; J.B.; writing—review and editing: J.B.; visualization: C.L., B.J.; supervision: J.B.; project administration: J.B.; funding acquisition: J.B., B.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Natural Science Foundation of China under Grant U1809201, by Fundamental Research Funds for the Provincial Universities of Zhejiang under Grant GK209907299001-003, by Zhejiang Provincial Natural Science Foundation of China under Grant LY20F010010, by 2020 Domestic Visiting Scholars Program at Higher Education Institutions of China under Grant FX2020008 and Grant FX2020010, and by Scientific Research Fund of Zhejiang Provincial Education Department under Grant Y201840534.

Acknowledgments: The authors would like to thank National Natural Science Foundation of China (NSFC), Zhejiang Provincial Natural Science Foundation of China (ZPNSFC), Zhejiang Provincial Education Department (ZPED) and Hangzhou Dianzi University for support this work and Xianghong Tang for his active participation in discussions and expert suggestions.

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Gallager, R.G. Low-density parity-check codes. *IRE Trans. Inf. Theory* **1962**, *8*, 21–28. [[CrossRef](#)]
2. MacKay, D.J. Good error-correcting codes based on very sparse matrices. *IRE Trans. Inf. Theory* **1999**, *45*, 399–431. [[CrossRef](#)]
3. Barry, J.R.; Kavcic, A.; LcLaughlin, S.W. Iterative timing recovery. *IEEE Signal Process. Mag.* **2004**, *21*, 89–102. [[CrossRef](#)]
4. Lee, D.U.; Valles, E.L.; Villasenor, J.D.; Jones, C.R. Joint LDPC decoding and timing recovery using code constraint feedback. *IEEE Commun. Lett.* **2006**, *10*, 189–191. [[CrossRef](#)]
5. Vallés, T.E.L.; Wesel, R.D.; Villasenor, J.D.; Jones, C.R. Carrier and timing synchronization of BPSK via LDPC code feedback. In Proceedings of the 2006 Fortieth IEEE Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 29 October–1 November 2006; Volume 8, pp. 2177–2181.
6. Bao, J.; Zhan, Y.; Lu, J. Iterative timing recovery via soft decision metrics of low-density parity-check decoding. *IET Commun.* **2010**, *4*, 1742–1751. [[CrossRef](#)]

7. Noels, N.; Lottici, V.; Dejonghe, A.; Steendam, H. A theoretical framework for soft-information-based synchronization in iterative (turbo) receivers. *EURASIP J. Wirel. Commun. Netw.* **2005**, *2*, 1–13. [[CrossRef](#)]
8. Herzet, C.; Ramon, V.; Vandendorpe, L. A theoretical framework for iterative synchronization based on the sum-product and the expectation-maximization algorithms. *IEEE Trans. Signal Process.* **2007**, *55*, 1644–1658. [[CrossRef](#)]
9. Saemi, A.; Meghdadi, V.; Cances, J.P.; Zahabi, M.R. Iterative (turbo) expectation-maximisation-based time and frequency synchronisation for multiple-input multiple-output-orthogonal frequency-division multiplexing systems. *IET Commun.* **2008**, *7*, 982–993. [[CrossRef](#)]
10. Mielczarek, B.; Svensson, A. Timing error recovery in turbo-coded systems on AWGN channels. *IEEE Trans. Commun.* **2002**, *50*, 1584–1592. [[CrossRef](#)]
11. Lehmann, F.; Kazem, A.; Salut, G. Blind turbo-detection in the presence of phase noise. *IET Commun.* **2009**, *3*, 1343–1353. [[CrossRef](#)]
12. Freedman, A.; Rahamim, Y.; Reichman, A. Maximum-mean-square soft-output (M(2)S(2)O): A method for carrier synchronisation of short burst turbo coded signals. *IEE Proc.-Commun.* **2006**, *153*, 245–255. [[CrossRef](#)]
13. Bao, J.; Zhao, M.; Zhong, J.; Cai, Y. Iterative timing recovery with turbo decoding at very low SNRs. In Proceedings of the 75th IEEE Vehicular Technology Conference (VTC2012 Spring), Yokohama, Japan, 6–9 May 2012; Volume 3, pp. 1–5.
14. Kwon, K.; Kim, S.; Hwang, J.; Paik, J. Performance evaluation of synchronization method for reducing the overall synchronization time in digital radio mondiale receivers. *KSII Trans. Internet Inf. Syst.* **2013**, *7*, 1860–1875.
15. Jeon, E.; Seo, J.; Yang, J.; Paik, J.; Kim, D. Iterative detection and ICI cancellation for MISO-mode DVB-T2 system with dual carrier frequency offsets. *KSII Trans. Internet Inf. Syst.* **2012**, *6*, 702–721. [[CrossRef](#)]
16. Mueller, K.; Müller, M. Timing recovery in digital synchronous data receivers. *IEEE Trans. Commun.* **1976**, *24*, 516–531. [[CrossRef](#)]
17. Chung, S.Y.; Richardson, T.J.; Urbanke, R.L. Analysis of sum product decoding of low-density parity-check codes using a Gaussian approximation. *IEEE Trans. Inf. Theory* **2001**, *47*, 657–670. [[CrossRef](#)]
18. Richardson, T.; Urbanke, R. The capacity of low density parity check codes under message passing decoding. *IEEE Trans. Inf. Theory* **2001**, *47*, 599–618. [[CrossRef](#)]
19. IEEE Std. P802.11n Draft. *Draft IEEE Standard for Local Metropolitan Networks—Specific Requirements. Annex P: LDPC Matrix Definitions*; Institute of Electrical and Electronics Engineers: New York, NY, USA, 2006; Volume 1, pp. 1–23.
20. Lagarias, J.C.; Reeds, J.A.; Wright, M.H.; Wright, P.E. Convergence properties of the Nelder-Mead simplex method in low dimensions. *IRE Trans. Inf. Theory* **1998**, *9*, 112–147. [[CrossRef](#)]