

Article

A Joint Parallel Timing Recovery Loop with Low Complexity for Terahertz Communication System and Its FPGA Implementation

Feifei Wang, Wentao Wang, Linshan Xue, Xianggang Liu and Huichao Zhou *

China Academy of Space Technology, Beijing 100094, China; doublefeif@163.com (F.W.);
xuelinshanx@alu.uestc.edu.cn (L.X.)

* Correspondence: zhouhuichao@buaa.edu.cn

Abstract

This paper proposes a low-complexity joint parallel timing recovery loop, which is well-suited for large-bandwidth terahertz (THz) communication systems. Specifically, the loop is jointly composed of a modified matched filter (MMF) and a timing error detector (TED), where sampling point offset correction is achieved by deleting, holding, or retaining data in parallel data caches (DCs), and timing phase error compensation is implemented by sliding the coefficients of the MMF. The feasibility of the proposed loop is verified using both Gardner and O&M TED. Numerical simulation results demonstrate that the loop operates efficiently, with a performance loss of less than 0.1 dB compared to the theoretical bit error rate (BER) curve. Furthermore, the loop is implemented on a THz field-programmable gate array (FPGA) platform, successfully realizing parallel demodulation of 15 Gbps 64QAM high-speed signals at 220 GHz. Notably, the proposed loop effectively reduces hardware resource consumption under a parallel architecture, providing a viable solution to address the current shortage of on-board resources in high-speed THz communication systems.

Keywords: terahertz communication; timing recovery loop; timing error detector; FPGA implementation

1. Introduction

With the rapid development in communication technology, people put forward higher requirements for information transmission rate. However, the current communication systems cannot meet the increasing demand for high-speed data transmission. Terahertz communication has the advantages of abundant spectral resources, large capacity, high energy efficiency, and large transmission bandwidth. Using the terahertz frequency band for high-speed communication can efficiently alleviate the situation of increasingly strained spectrum resources and break through the capacity limitation of the current wireless communication systems, which is an important technical path for the future development in wireless communication. So we take terahertz communication as the background to expand this paper.

There have been numerous studies focusing on high-speed wireless communication systems in the terahertz band, with breakthroughs spanning transmission performance [1–6]. Meanwhile, the modem will directly determine the performance of the communication systems. At present, the modulation mode of high-speed data transmission system is mostly MPSK, which is relatively simple and mature. With the increase of transmission rate and shortage of spectrum resources, the problem of low transmission efficiency caused by low-order modulation has become increasingly severe, failing to meet practical demands.



Academic Editors: Peter Han Joo Chong and Jiaguang Han

Received: 20 November 2025

Revised: 9 January 2026

Accepted: 29 January 2026

Published: 11 February 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Therefore, higher-order modulation methods (MQAM, MAPSK, etc.) begin to be used in high-speed modems [7,8]. However, high-order modulation remains a major challenge in terahertz (THz) communication systems. Its implementation involves higher complexity and reduces the Euclidean distance between adjacent constellation points, rendering the system extremely susceptible to timing errors. Even a tiny deviation from the optimal sampling point can induce intersymbol interference (ISI). Moreover, the stronger phase fluctuations and path loss inherent in the THz band further amplify this effect, thereby imposing much more stringent requirements on the synchronization performance of the communication systems.

In the actual communication systems, transceiver transmission delays prevent receivers from sampling at each symbol's optimal point, introducing timing errors that degrade demodulation performance, especially for high-order modulated terahertz systems. Therefore, timing recovery plays a crucial role in the overall performance of the receiver. Some mainstream timing error detection algorithms have been proposed in [9–12]. In 1978, Mueller and Muller (M&M) first proposed a timing error detection algorithm that each symbol only needs one sampling point, but it is decision-directed and affected by frequency offset [9]. In 1980, L.E. Franks proposed a timing error detection algorithm based on the maximum-likelihood estimation theory [10]. In order to solve the problem of carrier frequency offset influence and high complexity, F. M. Gardner proposed a timing error detection algorithm in 1986, which only needs two sampling points for each symbol, and was widely used in all-digital communication systems [11]. In 1993, F. M. Gardner extended his algorithm [13], and proposed a feedback architecture of all-digital timing recovery loop based on Farrow interpolation algorithm [14]. In 1988, M. Oerder and H. Meyr proposed a timing error detection algorithm with better performance for higher-order modulation, and the timing error estimate is unbiased [12]. Both feedforward [15] and feedback architectures are used in timing recovery loop. Feedforward enables fast acquisition but has low synchronization accuracy, suiting burst communications; feedback requires longer capture/convergence time but delivers high accuracy, needing only error magnitude/direction guidance for adjustment and thus being widely adopted in long-term continuous systems [16–21]. With the rapid development in high-speed wireless communication, Traditional serial timing recovery schemes are constrained by logic device clock frequencies in high-speed wireless systems, driving demand for parallel timing synchronization. Numerous timing recovery architectures applicable to various usage scenarios have been proposed [5,6,15,22–26]. Parallel Gardner-based via interpolation filters were proposed in [22,23,25–27], while C. Lin designed a frequency-domain parallel architecture based on O&M TED with a dual-feedback loop, which is firstly reported in Alternate Parallel Receiver (APRX) [24,28,29].

Amid the rapid development in reconfigurable hardware platforms and the increasing trend toward full digitization of receiver algorithms, hardware resource constraints remain a critical challenge for FPGA implementations in large-bandwidth terahertz communication systems. Numerous efforts have been devoted to reducing FPGA resource consumption of various receiver modules, timing recovery still stands out as a resource bottleneck [5,30–32]. In this paper, we propose a low-complexity joint parallel timing recovery loop based on a feedback time-domain architecture, which innovatively integrates timing error compensation with MMF and TED to jointly accomplish efficient timing recovery. In the proposed design, sampling point offset correction is realized by deleting, holding, or retaining data in parallel DCs, while timing phase error compensation is achieved by sliding/adjusting the MMF coefficients directly in the time domain, eliminating the need for resource-intensive FFT/IFFT operations and significantly reducing hardware resource consumption. And we

use high-order modulation to improve spectral efficiency and use parallel polyphase filter technique to save resources.

The rest of this paper is organized as follows: in Section 2, the N parallel time-domain timing recovery system architecture is given. Section 3 gives the specific working details of the timing recovery module. In Section 4, the methods of timing sampling point offset correction and timing phase error compensation in time domain are proposed. Then the simulation and implementation results are given in Section 5. Finally, we draw conclusions in Section 6.

2. Parallel Timing Recovery System Model

The joint N parallel paths timing recovery loop architecture is shown in Figure 1; all modules within the feedback loop contribute to the timing recovery process. The loop is mainly composed of four modules: data caches (DCs) module, modified matched filter (MMF) module, timing error detector (TED) module and controller module. Different from the traditional timing recovery loop, the loop proposed in this paper combines matched filter and timing error detector to jointly complete the timing recovery by shifting the matched filter coefficients to compensate the timing errors. This loop adopts high-order modulation to improve frequency band utilization and can be used in high-speed communication systems.

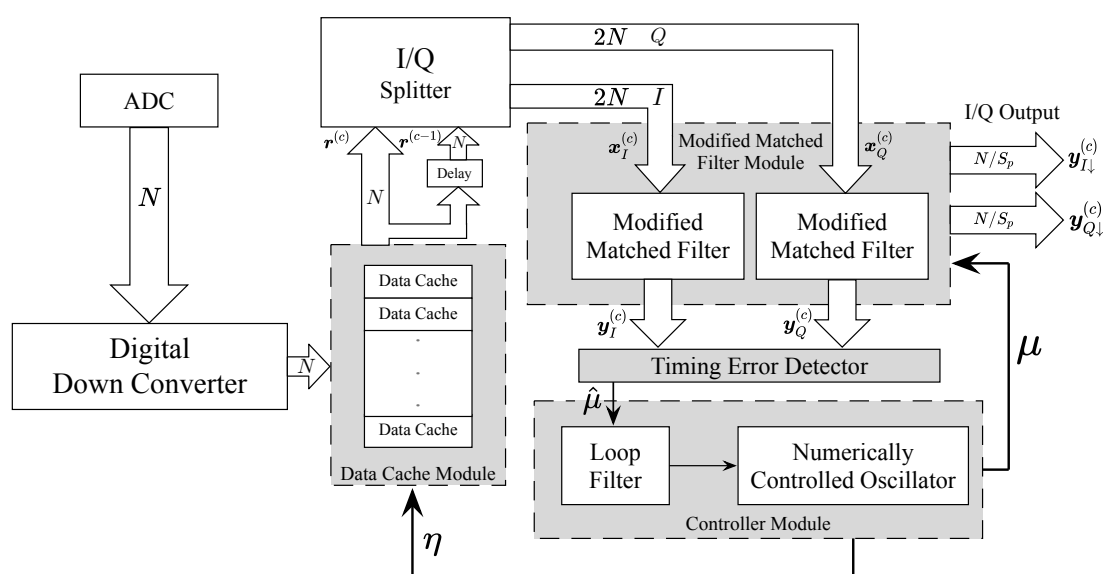


Figure 1. The N parallel time-domain feedback timing recovery loop architecture.

Firstly, a time-continuous intermediate frequency (IF) signal $r(t)$ is received. The received signal can be written as

$$\begin{aligned} r(t) &= \text{Re}[u(t)] \cos(2\pi f_c t) + \text{Im}[u(t)] \sin(2\pi f_c t) \\ &= I(t) \cos(2\pi f_c t) + Q(t) \sin(2\pi f_c t), \end{aligned} \quad (1)$$

where f_c is the carrier frequency of the IF signal, $I(t)$ and $Q(t)$ represent the in-phase and quadrature-phase of the sent symbol respectively. And

$$u(t) = \sum_{n=-\infty}^{\infty} s_n g_T(t - nT - \tau), \quad (2)$$

where $u(t)$ is the baseband equivalent complex envelope of the received modulated signal after down-conversion from the carrier frequency, $g_T(t)$ is the transmission signal pulse,

T is symbol duration, s_n are information symbols, and τ is an unknown, slowly varying timing delay that mainly caused by transmission delay and different sources of transceivers clocks. For simplicity, the effects of noise are not taken into account. In M-QAM modulation, $s_n = a_n + jb_n$ with $a_n, b_n \in \{\pm 1, \pm 3, \dots, \pm\sqrt{M} - 1\}$.

The received signal is input to the matched filter and gets the outputs

$$y(t) = r(t) * g_R(t), \quad (3)$$

where $*$ represents the convolution operation and $g_R(t)$ is a receiving filter (impulse response).

The timing error in the system is essentially caused by the mismatch between the transceiver sampling instants, which can be characterized by a time delay τ . The core goal of timing recovery is to compensate for this time delay to ensure optimal symbol sampling. To compensate for the timing delay τ , we need to obtain the filter output at the delayed instant $(t + \tau)$, i.e., $y(t + \tau)$. Based on the basic property of convolution, it can be expressed as follows

$$\begin{aligned} y(t + \tau) &= r(t) * g_R(t) * \Delta(t + \tau) \\ &= r(t + \tau) * g_R(t) \\ &= r(t) * g_R(t + \tau), \end{aligned} \quad (4)$$

where $\Delta(t)$ is the impulse function. From the analysis of (4), timing recovery can be expressed in two equivalent forms, which directly correspond to two timing error compensation methods. One realized by $r(t + \tau) * g_R(t)$, which requires shifting the input signal $r(t)$ to align with the original matched filter $g_R(t)$. And another realized by $r(t) * g_R(t + \tau)$, which requires shifting the impulse response of the matched filter to $g_R(t + \tau)$. In practical digital communication systems, the received signal $r(t)$ is sampled by the ADC, and it is difficult to flexibly implement time-domain sliding of the sampled signal. Therefore, we select the second method (sliding the matched filter) to compensate for the time delay τ . As illustrated in Figure 2, the time-domain shift of $g_R(t)$ to $g_R(t + \tau)$ does not alter the shape of the matched filter, only its time-domain position to adjust the filter's effective time delay to realize precise timing error compensation.

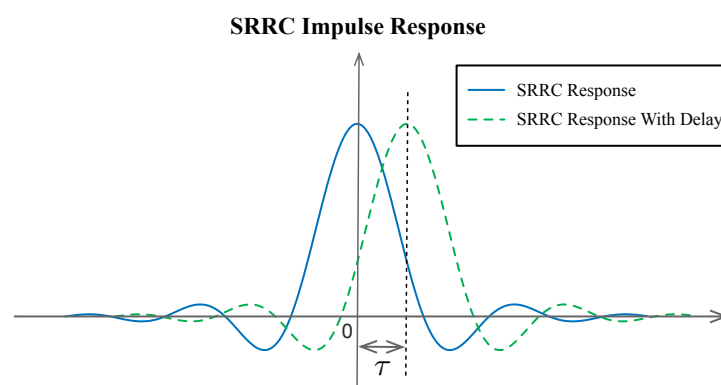


Figure 2. The impulse response of the SRRC filter (blue solid line). The filter has a delayed τ impulse response (green dashed line).

In practical communication systems, the accumulation of timing delay τ will cause optimal symbol slip at the receiver, resulting in the loss or repetition of symbols. Thus, the timing error compensation function of the recovery loop is divided into two core components: timing sampling point offset correction and timing phase error compensation.

The received signal $r(t)$ firstly sampled by Analog to Digital Converter (ADC) at rate $f_s = 1/T_s$, where T_s is the sampling period. And S_p represents the up-sampling ratio of the system. In line with Nyquist theorem, to ensure the sampled signal has no aliasing,

choosing a suitable S_p is very important. In this paper, to intuitively illustrate the I/Q separation principle and sequence mapping rule, we adopt a specific example assumption of $S_p = 4$, $f_s = 4f_c$. It should be emphasized that proposed scheme is compatible with any rational ratio of f_c/f_s that satisfies the Nyquist sampling criterion. Thus, the received signal can be expressed in the in-phase and quadrature-phase (I/Q) form.

$$\begin{aligned} r(mT_s) &= I(mT_s) \cos\left(2\pi \frac{f_c}{f_s} m\right) + Q(mT_s) \sin\left(2\pi \frac{f_c}{f_s} m\right) \\ &= I(mT_s) \cos\left(\frac{\pi}{2} m\right) + Q(mT_s) \sin\left(\frac{\pi}{2} m\right). \end{aligned} \quad (5)$$

Under this assumption, the received digital sequence can be expressed as

$$\mathbf{r} = [I(0), Q(1), -I(2), -Q(3), I(4), Q(5) \dots]. \quad (6)$$

The received signal sequence is parallelized into N parallel paths and written into N parallel data caches, and timing sampling point offset correction is achieved by controlling read operation of the data caches. For the convenience of operation, N is usually taken as an integer multiple of S_p when designing the system. Reading the data $\mathbf{r}^{(c)}$ from the data caches module at c -th clock, where $\mathbf{r}^{(c)}$ is an $N \times 1$ vector of received sequence and c is an index of the clock signal.

In the actual communication system, the signal sequence is infinitely long, so it needs to be segmented to reduce system output delay and ensure the real-time performance of the system. There are two main methods of segmentation group convolution processing, including overlapping addition and overlapping reservation. To simplify the implementation, overlapping reservation is used in this loop. The last N sampling points spliced with the current N sampling points that read from the data cache module to form a new sequence, where the last N sampling points are placed at the front of the new sequence. After that, a new sequence $\mathbf{x}^{(c)} = [\mathbf{r}^{(c-1)T}, \mathbf{r}^{(c)T}]^T$ is obtained, where $\mathbf{x}^{(c)}$ is an $2N \times 1$ vector and $(\cdot)^T$ represents the transpose of a vector. Sending $\mathbf{x}^{(c)}$ to the I/Q splitter and get sequences $\mathbf{x}_I^{(c)}$ and $\mathbf{x}_Q^{(c)}$ respectively, where $\mathbf{x}_I^{(c)}$ and $\mathbf{x}_Q^{(c)}$ are $L \times 1$ vector, $L = 2N$.

$$\begin{aligned} \mathbf{x}_I^{(c)} &= [I(0), 0, -I(2), 0, I(4) \dots -I(L-2), 0], \\ \mathbf{x}_Q^{(c)} &= [0, Q(1), 0, -Q(3), 0, Q(5) \dots -Q(L-1)]. \end{aligned} \quad (7)$$

Then I and Q-channel signal sequences are input to the MMF module separately to complete matched filter and timing phase error compensation by convolution. After the above operation, we discard the aliased data and get the output signal $\mathbf{y}_I^{(c)}$ and $\mathbf{y}_Q^{(c)}$, where $\mathbf{y}_I^{(c)}$ and $\mathbf{y}_Q^{(c)}$ are $N \times 1$ vector. Decimating $\mathbf{y}_I^{(c)}$ and $\mathbf{y}_Q^{(c)}$ at intervals S_p to get loop output sequence $\mathbf{y}_{I\downarrow}^{(c)}$ and $\mathbf{y}_{Q\downarrow}^{(c)}$, which compose the output $\mathbf{y}^{(c)}$ of the system.

$$\mathbf{y}^{(c)} = \mathbf{y}_{I\downarrow}^{(c)} + j\mathbf{y}_{Q\downarrow}^{(c)}, \quad (8)$$

where $\mathbf{y}_{I\downarrow}^{(c)}$, $\mathbf{y}_{Q\downarrow}^{(c)}$ and $\mathbf{y}^{(c)}$ are $N/S_p \times 1$ vector. $\mathbf{y}^{(c)}$ is made up of the optimal sampling points of the system and all of which were sampled at the position where the signal SNR is maximized.

In MMF module, we introduce an index Θ_{tap} to control the matched filter coefficients sliding step to achieve timing phase error compensation. $\mathbf{y}_I^{(c)}$ and $\mathbf{y}_Q^{(c)}$ calculated by the MMF module are input to the TED module to calculate timing error estimate $\hat{\mu}$. Then input $\hat{\mu}$ to controller module to extract the timing phase out μ and the timing sampling offset indicator η . And the η extracted from the numerical control oscillator(NCO) is fed back to

the data cache module. It controls the data caches control vector $\mathbf{r}_d^{(c)} \in \{0, 1\}^N$, which is used to achieve the timing sampling point offset correction. In order to compensate for the timing phase error, μ needs to be fed back to the MMF module to control the index Θ_{tap} . These will be covered in detail in the next section.

3. Parallel Timing Recovery Loop

3.1. Modified Matched Filter

In communication systems, data transmission over a bandlimited channel requires pulse shaping to eliminate intersymbol interference (ISI). In bandlimited digital communication systems, using a pair of matched square-root-raised-cosine (SRRC) filters in transmitter and receiver which satisfies Nyquist first criterion, can achieve ISI-free theoretically. The delay τ between the transmitter and receiver needs to be compensated to recover the symbol information of the sender correctly. In this work, we propose a MMF, whose primary function is to implement timing phase error compensation.

Firstly, the received time-continuous, high-order QAM signal $r(t)$ is sampled by ADC at rate $1/T_s$. Through parallel processing and the I/Q splitter, the received signal is separated into $x_I^{(c)}$ and $x_Q^{(c)}$. Due to the calculation of I and Q-channel is the same, for convenience, here we take I-channel as an example to analyze. Samples $x_I^{(c)}(mT_s) = x_I^{(c)}[m]$, $m \in [(c-1)L, cL)$ are taken at uniform intervals T_s . Sending $x_I^{(c)}$ to the matched filter (MF), and we get the output. The process of matched filtering is actually to convolve the signal with $g_R(t)$, and then keep the correct convolution data to get a valid output $y_I(t)$,

$$y_I(t) = \sum_m x_I^{(c)}(mT_s)g_R(t - mT_s) \quad (N+1)T_s \leq t \leq 2NT_s, \quad (9)$$

In the timing recovery loop, the timing error estimate $\hat{\mu}$ calculated by the TED is then sent to the controller module to get the appropriate parameter timing phase out μ . And then it is fed back to the MMF module to compensate the delay τ .

After compensation, the matched filter output $y_I(t)$ can be written as

$$y_I(t + \mu) = \sum_m x_I^{(c)}(mT_s)g_R(t + \mu - mT_s). \quad (10)$$

In N parallel paths digital receiver, we propose a modified matched filter which is slidable. Thus, timing error compensation can be achieved by shifting $g_R(t)$. Resample $y_I(t + \mu)$ at time instants $t + \mu = kT_i$, where k is the index of the sequence calculated by the MMF module and $y_I^{(c)}(kT_i) = \mathbf{y}_I^{(c)}[k]$, $k \in [(c-1)N, cN)$ at the adjustable intervals T_i ,

$$y_I^{(c)}(kT_i) = \sum_m x_I^{(c)}(mT_s)g_R(kT_i - mT_s), \quad (11)$$

where m is signal index, we define the filter index i as follows

$$i = \lfloor kT_i/T_s \rfloor - m - \mu, \quad (12)$$

where $\lfloor p \rfloor$ means largest integer not exceeding p . Also define the basepoint index m_k , which originates from the sampling index m , that is, $m_k = mT_s$ and $m_{k+1} = (m + S_p)T_s$.

Between each two basepoint, there are $S_p - 1$ samples. Thus, we get equation $(kT_i - mT_s) = (i + \mu)T_s$. The outputs of loop can be rewritten as

$$\begin{aligned} y_I^{(c)}(kT_i) &= y_I^{(c)}[(m + \mu)T_s] \\ &= \sum_{i=0}^{N_2-1} x_I^{(c)}[kT_i - (i + \mu)T_s] g_R[(i + \mu)T_s]. \end{aligned} \quad (13)$$

In high-speed terahertz communication, the traditional serial architectures are no longer applicable, so the timing recovery loop proposed in this paper adopts N -path parallel architecture. Through the data cache module, the sequence $x_I^{(c)}$ is sent into the MMF module and convolved with the matched filter coefficients, which has excellent ISI rejection capability. After MMF module, we get the output sequence $y^{(c)}$ with adjustable interval $S_p T_i$. Also get the sequences $y_I^{(c)}$ and $y_Q^{(c)}$ that are input to TED module to calculate timing error estimate.

3.2. Controller

Controller module is indispensable in the timing recovery loop because the timing error estimate calculated by TED cannot be directly applied to the MMF module for compensation and the data cache module for correction. As is shown in Figure 1, the controller module is composed of loop filter (LF) and NCO in the timing recovery loop proposed in this paper. The role of the controller module is to extract the available information about basepoint index m_k and timing phase out μ for timing sampling point offset correction and timing phase error compensation, respectively.

As is shown in Figure 3, the relation between ADC sampling points and optimal sampling points is illustrated. The sampling points are timed uniformly through at a fixed interval T_s by ADC. Due to the different sources of transmitter and receiver clock, timing delay will inevitably occur in the system. With the accumulation of timing delay, the basepoint will be offset and m_k is no longer the optimal basepoint index. Thus, we introduce a parameter η to represent the offset of the basepoint.

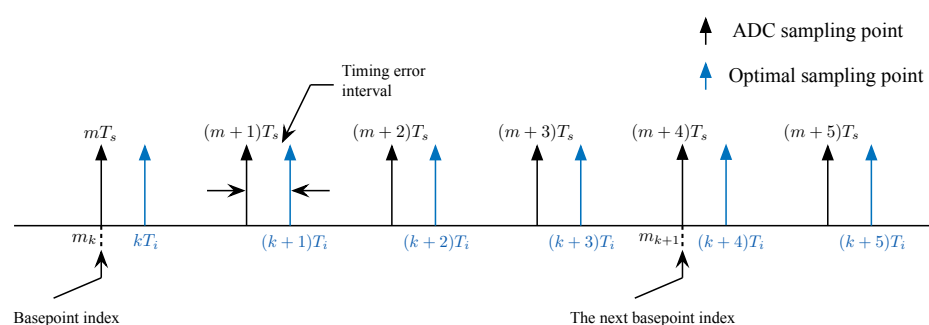


Figure 3. The relation between sampling points.

Under the N -path parallel architecture, the loop calculates the timing phase out every ρ symbols and $\rho = N/S_p$. Due to the slow change of timing error, sharing one μ among multiple sampling points will hardly introduce extra error. In parallel implementation, N sampling points are simultaneously input to the TED module to calculate the timing error

estimate $\hat{\mu}$. Then send it to the LF to filter out the interference components mainly caused by noise and make the output smooth. The LF output $\mu_l(c)$ at the c -th clock is

$$\begin{aligned} w_1(c) &= c_1 \times \hat{\mu}(c), \\ w_2(c) &= c_2 \times \hat{\mu}(c) + w_2(c-1), \\ \mu_l(c) &= w_1(c) + w_2(c), \end{aligned} \quad (14)$$

where c_1 and c_2 are LF coefficients, its design corresponds to the equivalent noise bandwidth of the loop [33]. In hardware implementation, the parameters are often set to integer multiples of $1/2$, which facilitates replacing multiplication operations with shift operations.

The smoothed timing error estimate μ_l is input to the NCO and used as the control word of the NCO. The difference equation can be expressed as

$$\mu'(c) = \mu(c-1) + \mu_l(c), \quad (15)$$

where μ' is a temporary value of the timing phase out and μ is the timing phase out that extracts from the NCO. According to (15), it can be seen that the theoretical timing phase out will continuously increase or decrease to infinity. In order to limit the timing phase out to a fixed interval, the temporary value μ' need to be recomputed to get the timing phase out μ , which is discussed as follow. According to the timing delay relation diagram between ADC sampling points and optimal sampling points given in Figure 4, we can see that with the continuous accumulation of timing delay, μ' will exceed or lag one sampling point. And the current basepoint index is no longer suitable, so we use the parameter η to control the offset.

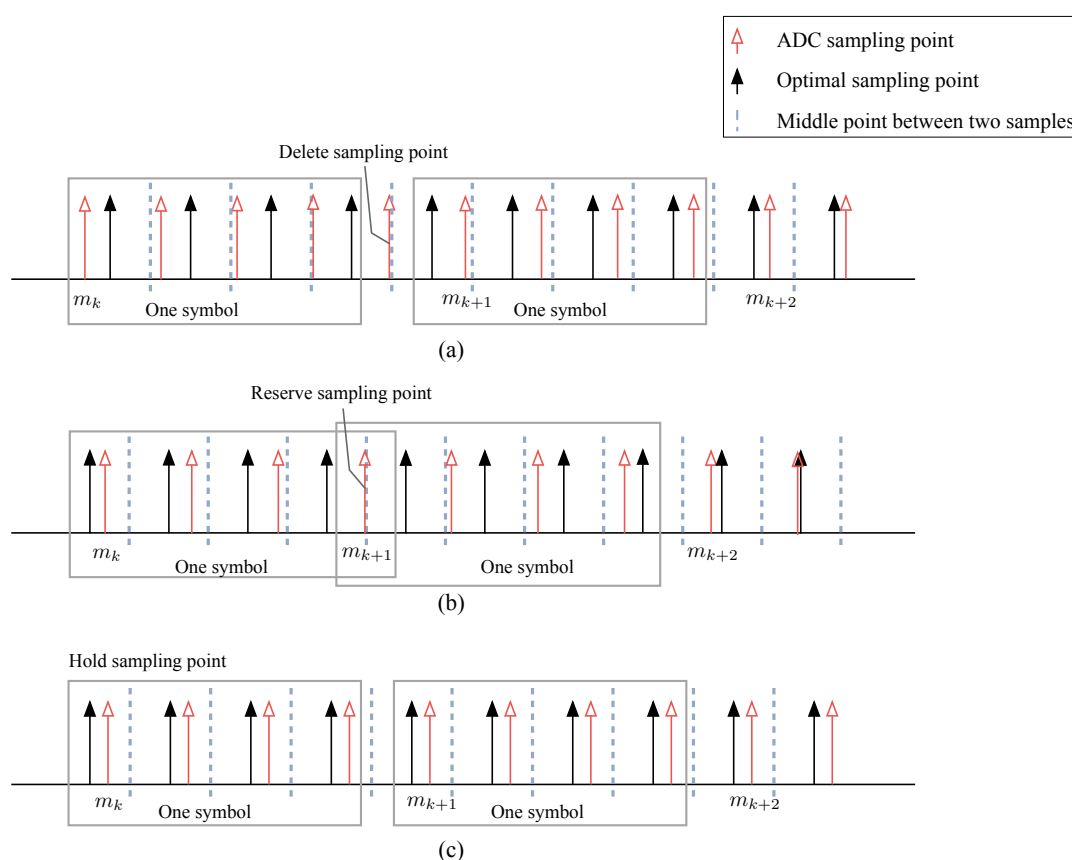


Figure 4. The operations of selecting basepoints for three different situations: (a) Delete. (b) Reserve. (c) Hold.

Figure 4a describes a situation that the ADC sampling rate is faster than the true symbol rate. As the result, the optimal sample point move forward as time pass and finally exceed the middle point of two adjacent ADC sampling points, thus, one ADC sampling point needs to be deleted (namely jumped over) and the basepoint moves one more point to the right side ($m_{k+1} = m_k + S_p + 1$). In practical, $\eta(c) = 1$ is fed back to the data cache module to achieve the deletion of the sampling point. In this situation, temporary value of the timing phase out μ' is larger than 0.5 in value and it should be subtracted by 1 to get the timing phase out μ which is limited to a fixed interval (i.e., $\mu \in [-0.5, 0.5)$). Figure 4b describes an opposite situation that the ADC sampling rate is slower than the true symbol rate, and one sampling point needs to be reserved (i.e., $\eta(c) = -1$). Simultaneously, the basepoint moves one fewer point to the right side ($m_{k+1} = m_k + S_p - 1$). Figure 4c demonstrates the situation that $\mu' \in [-0.5, 0.5)$, ADC sampling points do not need to be deleted or reserved (i.e., $\eta(c) = 0$), it means the basepoint index m_k has no more special operations to be done.

The timing recovery loop in this paper is a double feedback loop. After controller, we extract the timing phase out μ and the timing sampling point offset indicator η . μ is feedback to the MMF module to control the sliding step of the matched filter coefficients to achieve timing phase error compensation and η is feedback to the data cache module, which controls the reading of sampling points to delete, hold or reserve the ADC sampling points. After these two steps, timing error compensation at the receiver is completed. The controller algorithm for our timing recovery loop can be summarized as Algorithm 1:

Algorithm 1 Controller algorithm for the timing recovery loop

Input: The c -th clock timing error estimate $\hat{\mu}(c)$ from TED.

- 1: Input $\hat{\mu}(c)$ to the LF and get the smoothed timing error estimate μ_l according to (14).
- 2: $\mu'(c) = \mu(c-1) + \mu_l(c)$.
- 3: **if** $\mu'(c) > 0.5$ **then**
- 4: $\mu(c) = \mu'(c) - 1$ $\eta(c) = 1$
- 5: **else if** $\mu'(c) \leq -0.5$ **then**
- 6: $\mu(c) = \mu'(c) + 1$ $\eta(c) = -1$
- 7: **else**
- 8: $\mu(c) = \mu'(c)$ $\eta(c) = 0$
- 9: **end if**

Output: Output the c -th time clock timing error phase out $\mu(c)$ and timing sampling point offset indicator $\eta(c)$.

3.3. Timing Error Detector

The TED plays a crucial role on the system performance of the timing recovery loop. Gardner TED algorithm [11] and O&M TED algorithm [12] are enduring for a long time and have been widely used. The Gardner TED algorithm (taking QPSK as an example) can be expressed as

$$\begin{aligned} \hat{\mu}(c) = & \mathbf{y}_I^{(c)}(m_k - S_p/2) \left[\mathbf{y}_I^{(c)}(m_k) - \mathbf{y}_I^{(c)}(m_{k-1}) \right] \\ & + \mathbf{y}_Q^{(c)}(m_k - S_p/2) \left[\mathbf{y}_Q^{(c)}(m_k) - \mathbf{y}_Q^{(c)}(m_{k-1}) \right]. \end{aligned} \quad (16)$$

In the case of high-order modulation, (16) needs to be modified. According to [34], we get the modified Gardner TED algorithm that suitable for high-order QAM modulation.

$$\begin{aligned}\hat{\mu}(c) = & \left\{ \mathbf{y}_I^{(c)}(m_k - S_p/2) - \frac{\mathbf{y}_I^{(c)}(m_k) + \mathbf{y}_I^{(c)}(m_{k-1})}{2} \right\} \\ & \times [\mathbf{y}_I^{(c)}(m_k) - \mathbf{y}_I^{(c)}(m_{k-1})] \\ & + \left\{ \mathbf{y}_Q^{(c)}(m_k - S_p/2) - \frac{\mathbf{y}_Q^{(c)}(m_k) + \mathbf{y}_Q^{(c)}(m_{k-1})}{2} \right\} \\ & \times [\mathbf{y}_Q^{(c)}(m_k) - \mathbf{y}_Q^{(c)}(m_{k-1})].\end{aligned}\quad (17)$$

O&M TED algorithm requires at least four samples for each symbol (e.g., $S_p \geq 4$), and the calculation is more complicated. The O&M TED algorithm can be expressed as

$$\hat{\mu}(c) = -\frac{1}{2\pi} \arg(Y_I(c)), \quad (18)$$

where $Y_I(c)$ is the I -th data segment's spectral components at c -th clock determined by computing the complex Fourier coefficient at the symbol rate. Every data segment includes EN samples, E is the cumulative length which means every data segment spans E clock.

$$Y_I(c) = \sum_{n=l}^{E+I-1} Y(n). \quad (19)$$

$Y(c)$ is the spectral components determined by N samples at c -th clock,

$$Y(c) = \sum_{k=0}^{N-1} \mathbf{F}^{(c)}[k] e^{-j2\pi k/S_p}, \quad (20)$$

where $\mathbf{F}^{(c)}[k] = (\mathbf{y}_I^{(c)}[k])^2 + (\mathbf{y}_Q^{(c)}[k])^2$.

The choice of the cumulative length E is related to the O&M TED's estimate accuracy and the capture range of the timing phase error. So, choosing an appropriate cumulative length E is important.

4. Timing Recovery Loop Implementation

In the implementation of the N -path parallel timing recovery loop, how to achieve timing sampling point offset correction and timing phase error compensation is very important. The specific implementation details are as follows.

4.1. Modified Matched Filter Implementation

In the timing recovery loop, the timing phase out μ extracted from the controller is fed back to the MMF module, and the timing phase error is compensated by sliding the matched filter coefficients. In digital communication system, finite-impulse response (FIR) filter with SRRC characteristic is often used as matched filter. Generally, the filter is truncated into S_l symbols, each symbol period contains S_p samples. The order of filter is $S_l \times S_p$ and the length of its coefficients is $N_2 = S_l \times S_p + 1$. And the selection of the S_l value is limited by the length of the signal sequence to ensure sufficient convolution. What's more, the value of roll-off factor α determines the bandwidth of the signal that $B = (1 + \alpha)R_s$, where R_s is symbol rate, and we need to choose a suitable α value according to different TED application conditions. For example, Gardner TED is considered to be applicable for approximately 40–100 percent excess bandwidth of the signal [11]. However, higher α value means wider bandwidth the channel need, so choosing a proper α value is important to balance the resource of bandwidth and TED performance.

From the analysis above, the value of μ is limited to a fixed interval (i.e., $\mu \in [-0.5, 0.5)$), namely the timing error is at range from the left middle point of one ADC sampling point to the right middle point of the same point. To achieve timing phase error compensation accurately between every two ADC sampling points according to (4), it is necessary to generate a long series of matched filter coefficients in advance. That is, K points between every two samples of the matched filter should be inserted, where K is called filter oversampling multiple value and it usually takes as an integer multiple of 2 for implementation. According to [35,36], we get the extended MMF coefficients sequence g_p . Figure 5 shows an example of a SRRC pulse g_p with $\alpha = 0.35$, $S'_p = S_p K = 4K$, $S_l = 4$ and $K = 4$. In this figure, blue points form a normal SRRC pulse with oversampling rate $S_p = 4$ and gray points, which quantity is associated with K , are theoretical points between every two adjacent blue points at uniform intervals.

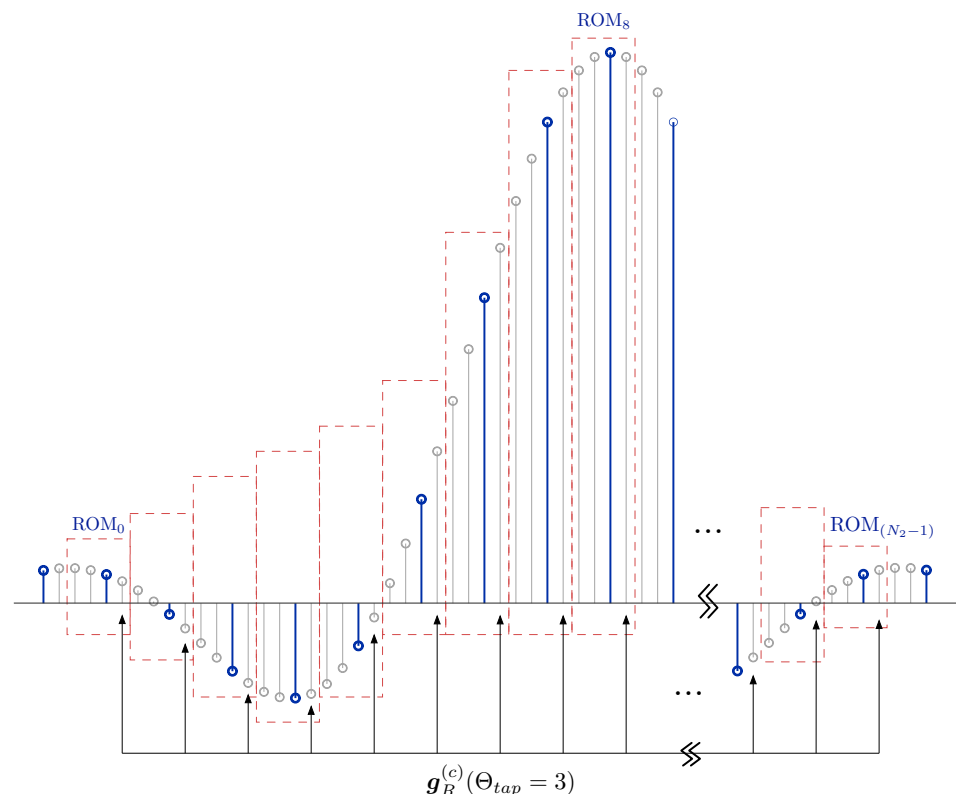


Figure 5. The extended SRRC pulse in discrete time domain with $K = 4$, $S_p = 4$ and $S_l = 4$.

The input value μ from controller module cannot be used directly to get the MMF coefficients, because it is not integer and it is not suitable to lookup coefficients in practice. For implementation, we introduce an index Θ_{tap} to extract $g_R^{(c)}$ from the extended sequence g_p ,

$$\begin{aligned}\Theta_{tap}(c) &= \frac{K}{2} + \lfloor \mu(c)K \rfloor \\ &= \left\lfloor \left(\frac{1}{2} + \mu(c) \right) K \right\rfloor.\end{aligned}\quad (21)$$

Since $\mu \in [-0.5, 0.5)$, the index of the MMF coefficients $\Theta_{tap} \in [0, K - 1]$. For parallel implementation, we split the coefficients g_p into $N_2 = S_l \times S_p + 1$ sections with uniform interval K and these sections are stored in N_2 independent ROMs (namely, ROM_i , $i = 0, 1, \dots, N_2 - 1$), respectively. As shown in Figure 5, each ROM is based on the samples corresponding to the matched filter of length N_2 without offset as the center (blue points), stores its $\lfloor K/2 \rfloor$ points in the negative direction and $\lfloor K/2 \rfloor - 1$ points in the positive di-

rection. Under this condition, its sliding range is $[-K/2, K/2)$. The slid MMF coefficients $\mathbf{g}_R^{(c)}$ is read from each ROMs by index Θ_{tap} . Take $\Theta_{tap} = 3$ as an example, $\mathbf{g}_R^{(c)}$ is composed with the 3rd data stored in each ROM, as shown in Figure 5.

After reading the $\mathbf{g}_R^{(c)}$ from ROMs, convolving sequences $\mathbf{x}_I^{(c)}$ and $\mathbf{x}_Q^{(c)}$ with it respectively to achieve timing phase error compensation and get output sequence $\mathbf{y}^{(c)}$. In order to reduce resource consumption and improve operation efficiency, we use polyphase filtering technique to achieve matched filtering by dividing the convolution process into several smaller steps. According to the polyphase filtering theory, (13) can be rewritten as the form of L path parallel polyphase filtering form in z -transform domain:

$$\begin{aligned} \mathbf{y}_I^{(c)}[k] &= \sum_{i=0}^k \mathbf{g}_R^{(c)}[i] \mathbf{x}_I^{(c)}[k-i] + z^{-L} \left\{ \sum_{i=k+1}^{L-1} \mathbf{g}_R^{(c)}[i] \mathbf{x}_I^{(c)}[L+k-i] \right\}, \quad 0 \leq k \leq L-2 \\ \mathbf{y}_I^{(c)}[L-1] &= \sum_{i=0}^{L-1} \mathbf{g}_R^{(c)}[i] \mathbf{x}_I^{(c)}[L-1-i]. \end{aligned} \quad (22)$$

Since the length of the filter is N_2 , it means $\mathbf{g}_R^{(c)}[N_2+1] \sim \mathbf{g}_R^{(c)}[L-1]$ are all equaling to 0, the equation can be further simplified to reduce the calculational complexity.

4.2. Data Cache Implementation

As the timing delay accumulate, timing sampling point offset which causes timing phase error compensation by MMF mentioned above failure is inevitable. According to the analysis in Section 3.2, η is used to control data cache module, that is, the offset correction is achieved by reserving or deleting the sampling points. Under the N -path parallel architecture, data sampled by the ADC are stored in N parallel data caches. We introduce a control vector $\mathbf{r}_d^{(c)}$ to determine the data reading behavior of data caches,

$$\mathbf{r}_d^{(c)} = [r_{d0}^{(c)}, r_{d1}^{(c)}, \dots, r_{d(N-1)}^{(c)}]^T \in \{0, 1\}^N, \quad (23)$$

where $r_{di}^{(c)}$ is the i -th data cache's control signal at c -th clock. As is shown in Figure 6, $r_{di}^{(c)}$ can be used as a switcher of the q -th cache. $r_{di}^{(c)} = 1$ indicates that the i -th switcher is on so that i -th data cache is allowed to read data at the c -th clock. And when $r_{di}^{(c)} = 0$, the switcher is off and there is no output from the i -th cache.

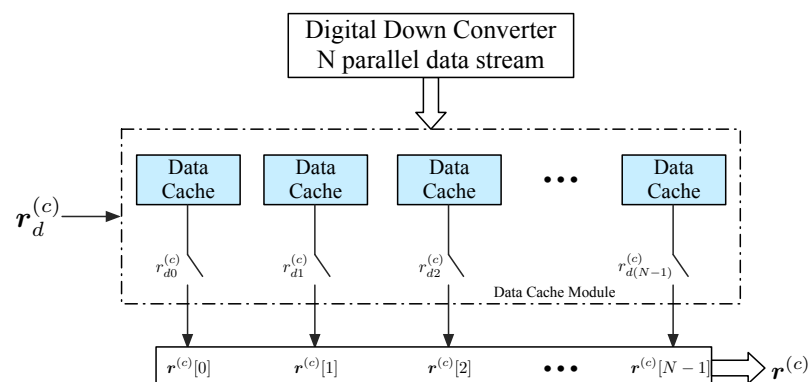


Figure 6. The control switch of the data cache.

The value of the control vector $\mathbf{r}_d^{(c)}$ is determined by the timing sampling point offset $\eta(c)$. When $\eta(c) = 0$, it means that there is no timing sampling point offset at the current moment and all data caches read data normally, that is, $\mathbf{r}_d^{(c)}$ is a full 1 vector. When $\eta(c) = -1$, the ADC sampling clock is slower than the true symbol rate and one sampling

point needs to be reserved. At this point, the q -th data cache does not read data and maintains the previous output,

$$(\mathbf{r}_d^{(c)} \leftarrow_q 0) = [r_{d0}^{(c)}, \dots, r_{d(q-1)}^{(c)}, 0, \dots, r_{d(N-1)}^{(c)}]^T, \quad (24)$$

where q is a decremental value with range $[0, N - 1]$, i.e., $q = q - 1$ when the next $\eta(c) = -1$ is coming.

When $\eta(c) = 1$, the ADC sampling clock is faster than the true symbol rate and one sampling point needs to be deleted. Firstly, reading the data that need to be deleted from p -th data cache which will be temporarily stored without entering the timing recovery loop. p is an incremental value with range $[0, N - 1]$, i.e., $p = p + 1$ when the next $\eta(c) = 1$ is coming. At this time, we introduce an enable signal $\mathcal{C}$ to control whether to discard the sequence $\mathbf{r}^{(c)}$ read from data caches. When $\mathcal{C} = 0$, the output sequence $\mathbf{r}^{(c)}$ is discarded, that means the current sequence $\mathbf{r}^{(c)}$ will not enter into the timing recovery loop. The system suspends the current operation and waits for the next clock to come. Then, all N parallel data caches read data normally.

The algorithm proposed in this paper to achieve the reservation or deletion of sampling points of data caches in the timing recovery loop can be described as Algorithm 2. In FPGA implementation, due to the high clock stability achievable in current industrial applications, timing synchronization failure caused by clock accumulation can be neglected.

Algorithm 2 Timing sampling point offset correction

Input: The timing sampling point offset indicator $\eta(c - 1)$.

Initialization: Initialize the vector $\mathbf{r}_d^{(c)} ([r_{di}^{(c)}]_{i=0}^{N-1} = 1)$, $q = N - 1$ and $p = 0$.

- 1: **if** $\eta(c - 1) = -1$ **then**
- 2: $q = \text{mod}(q - 1, N)$ and $[r_{di}^{(c)}]_{i=0}^{N-1} = 1$
- 3: $(\mathbf{r}_d^{(c)} \leftarrow_q 0)$ $\mathcal{C} = 1$
- 4: **else if** $\eta(c - 1) = 1$ **then**
- 5: $p = \text{mod}(p + 1, N)$ and $[r_{di}^{(c)}]_{i=0}^{N-1} = 0$
- 6: $(\mathbf{r}_d^{(c)} \leftarrow_p 1)$ $\mathcal{C} = 0$
- 7: **else**
- 8: $[r_{di}^{(c)}]_{i=0}^{N-1} = 1$
- 9: $\mathcal{C} = 1$
- 10: **end if**
- 11: Read data sequence $\mathbf{r}^{(c)}$ from data caches (DCs).
- 12: $\mathbf{r}^{(c)}[m] = \text{DCs}(\mathbf{r}_d^{(c)}[m])$.
- 13: Rearrange the output data which read from data caches to get the correct output sequence $\mathbf{r}^{(c)}$.
- 14: Determine whether the output data $\mathbf{r}^{(c)}$ is input to the timing recovery loop.

Output: The data sequence $\mathbf{r}^{(c)}$ that read from data cache module.

4.3. Filter Oversampling Multiple Value

The value of K is important to the accuracy of timing phase error compensation, which can severely affect the performance of the loop. To analyze clearly, we introduce a value v_k ,

$$v_k = \frac{\lfloor K \times \mu \rfloor}{K} - \mu, \quad (25)$$

which represents the difference between the compensation value of the timing phase error through the MMF module and the actual timing phase output. It can be seen from (25) that the larger the value of K , the higher the timing recovery accuracy of the loop. However,

in actual communication system, $K \rightarrow \infty$ is impractical. According to the requirements of timing recovery performance, selecting an appropriate K value is very important. It should be noted that in FPGA implementation, the larger the the value of K , the larger the storage resources consumption. We use numerical simulation to determine how to select an appropriate K value, which can save resources to the greatest extent while ensuring the timing recovery performance.

5. Simulation and Hardware Implementation

This section aims to verify the feasibility of the proposed timing recovery loop. Sampling deviations are introduced in numerical simulations to emulate the timing errors encountered in practical communication systems, thereby evaluating the timing recovery performance of the proposed loop. To further validate the algorithm's feasibility under real-world operating conditions, a hardware platform for terahertz communication systems is established to conduct experimental tests on the proposed timing recovery loop.

5.1. Numerical Simulation Results

This section presents numerical simulations of the proposed timing recovery loop based on MMF compensation. The feasibility of the loop is evaluated using two key indicators: the clarity of the constellation diagram after timing recovery, and the proximity of the system's BER to the theoretical BER—with the deviation between the post-recovery BER and the theoretical BER required to fall within an acceptable range. As detailed in Section 4.3, a suitable value of parameter K is determined through loop performance simulation before FPGA implementation. Figure 7 illustrates the BER curves corresponding to different K values under a timing sampling deviation of $P = 0.99975$, where P denotes the timing sampling deviation, that is, resampling the transmitter sequence at P times of the original sampling rate.

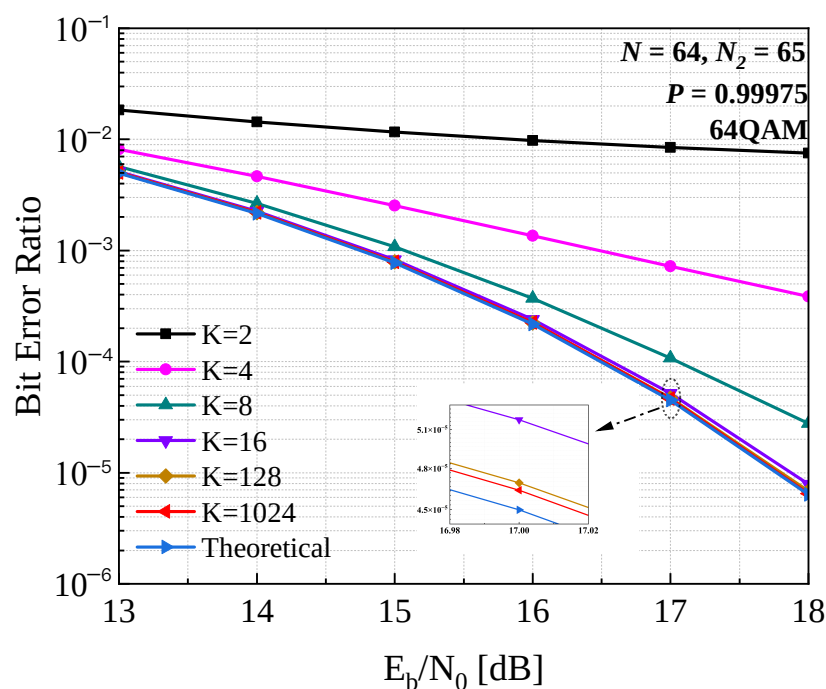


Figure 7. The BER curve for different K under $P = 0.99975$.

When $K \leq 4$, the BER curve decreases slowly, and the performance gap tends to widen as E_b/N_0 increases. When $K \geq 16$, the BER curve tends to align with the theoretical BER curve, and with the increase of K , the performance loss is less than 0.1 dB compared to the

theoretical BER curve. Thus, we choose $K = 32$ for both loop performance simulation and hardware implementation. The parameters of this architecture is shown in Table 1.

Table 1. Basic parameters of timing recovery loop.

System Parameters	Variable	Value
Parallel paths	N	64
Modulation order	M -QAM	64-QAM
SRRC filter length	N_2	65
Up-sampling ratio	S_p	4
Symbol rate	R_s (Gsps)	2.5 Gsps
Timing sampling deviation	P	0.99975
TED	/	O&M
Filter oversampling multiple value	K	32

To comprehensively characterize the performance of the proposed timing recovery loop, numerical simulation results are presented in this part. Figure 8 illustrates a comparison of the constellation diagrams before and after the operation of the timing recovery loop: Figure 8a shows the constellation diagram prior to loop processing, while Figure 8b depicts the constellation diagram after loop optimization. A clearly distinguishable 64QAM constellation pattern is observed in Figure 8b, demonstrating the effective timing recovery capability of the proposed loop. And the EVM performance is shown in Figure 9, after timing recovery, the average EVM of the system is 3.86%, and most symbols have EVM values concentrated in the range of 0 to 4%. To further validate the loop's performance under noisy conditions, additive white Gaussian noise (AWGN) is introduced into the received sequence, and BER curves are obtained over an E_b/N_0 range of 13 dB to 18 dB is given. Figure 10 presents the BER curves corresponding to different TED algorithms. It can be observed that the performance of the proposed timing recovery loop outperforms the algorithms reported in [25,28]. Figure 11a illustrates the smoothed timing error estimate μ_l of the system, and Figure 11b depicts the NCO output μ . As observed from the figures, the system output μ_l oscillates around the actual timing error value. Combined with the analysis results presented in Figures 8–11, we can draw a conclusion that the proposed joint timing recovery loop exhibits efficient operational performance in practical scenarios.

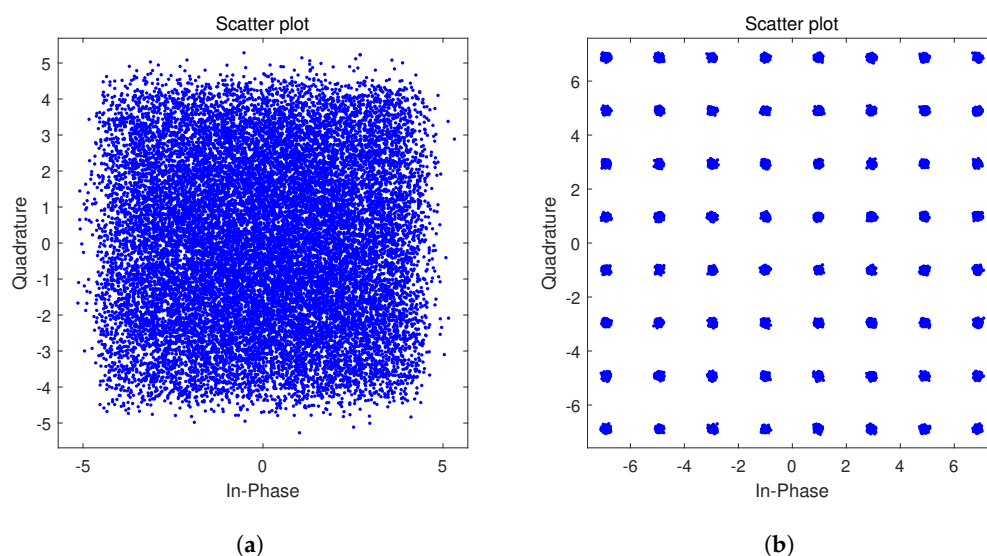


Figure 8. The constellation diagrams before (a) and after (b) timing recovery loop.

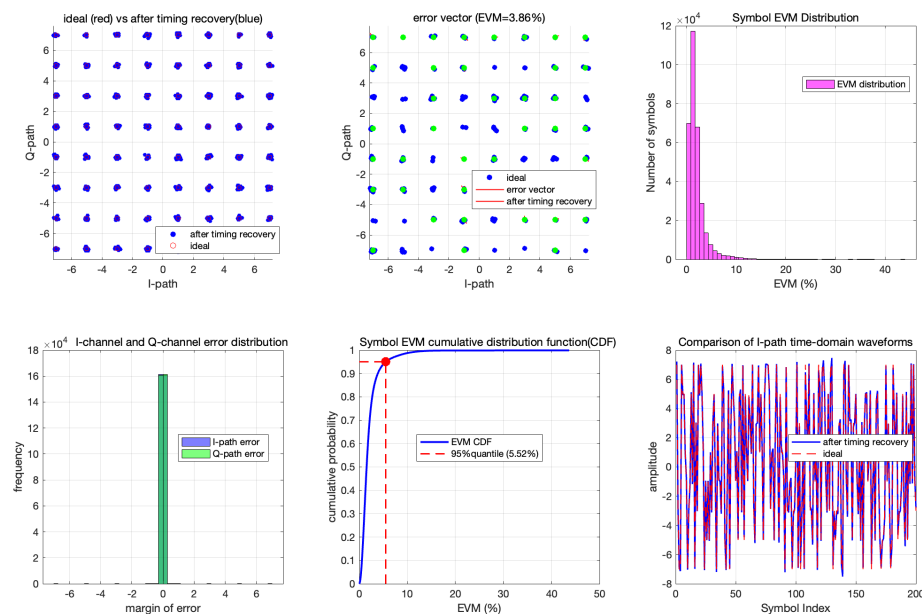


Figure 9. The EVM performance after loop.

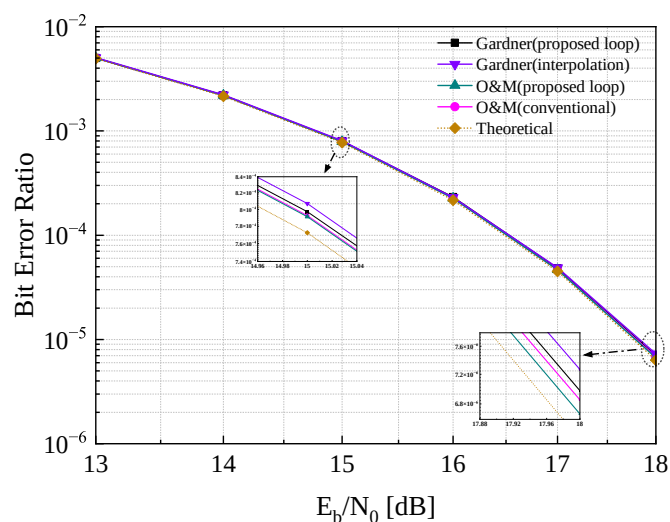


Figure 10. The BER curve of the loop.

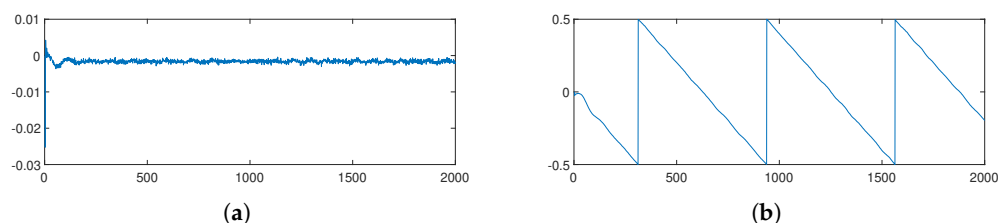


Figure 11. Parallel timing recovery system output parameters. (a) is the smoothed timing error estimated μ_l of the system. (b) is the the timing phase out μ .

5.2. FPGA Implementation and Experimental Results

To verify the feasibility and practicality of the proposed time-domain joint algorithm, a terahertz high-speed communication transceiver verification platform was established, and corresponding hardware tests were conducted. The main implementation parameters

are listed in Table 1, with the timing sampling deviation introduced by the actual communication system. Specifically, the filter length was selected as $N_2 = 33$, which balances loop performance and hardware resource consumption effectively.

The topology and physical photograph of the hardware verification platform are illustrated in Figure 12a,b respectively. Xilinx Zynq UltraScale+ RFSoc ZCU216 (ZU49DR) Evaluation Kit (marked as (1) in Figure 12b) was employed to generate a 64QAM-modulated signal with a data rate of 15 Gbps (2.5 Gbps) at an intermediate frequency (IF) of 2.5 GHz. The generated digital sequence from FPGA is converted into analog signal by one of DAC channels of XM655 16T16R Breakout Add-on Card and then mixed with radio frequency (RF) of 220 GHz [37–39] by subharmonic mixer (marked as (2) in Figure 12b), which is driven by 12.222 GHz reference clock from signal source. Finally, the transmitting signal propagates through a THz antenna.

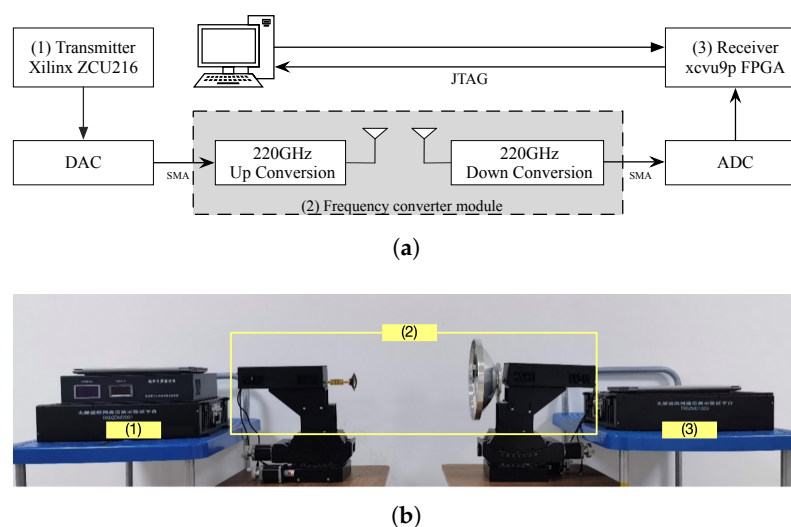


Figure 12. The hardware verification platform of the timing recovery loop. (a) is the topology of the platform. (b) is the photograph of platform.

At the receiver, the 220 GHz RF signal captured by the THz antenna was amplified by a low-noise amplifier (LNA) before being downconverted to an IF signal via another subharmonic mixer. The IF signal was then digitized by an ADC channel of the TI ADC12DJ5200RF EVM board at a sampling rate of 10 Gbps. A Xilinx UltraScale+ VCU118 (XCVU9P) Evaluation Kit (marked as (3) in Figure 12b) was utilized to implement the full receiver signal processing chain, including the proposed timing recovery loop, equalization, carrier synchronization, and 64QAM demodulation. The full-flow receiver data (from ADC input to demodulated output) was recorded to verify the end-to-end performance. The constellation diagram and BER after each processing stage were monitored to confirm the effectiveness of the proposed timing recovery algorithm in the actual receiver chain.

Firstly, a system direct-connection test was conducted, where the only timing deviation introduced into the system originated from the ADC sampling instants. This test scenario focuses on verifying the core performance of the proposed timing recovery loop under a simplified yet practical non-ideal condition. The constellation diagrams after timing recovery were displayed in ChipScope software on a PC via the JTAG interface. Figure 13a presents the constellation diagram of the received signal before timing recovery, while Figure 13b shows the constellation diagram after the proposed timing recovery loop is enabled. It can be observed that the proposed loop achieves excellent performance in mitigating ADC-induced sampling deviations, as evidenced by the fully distinguishable 64QAM constellation.

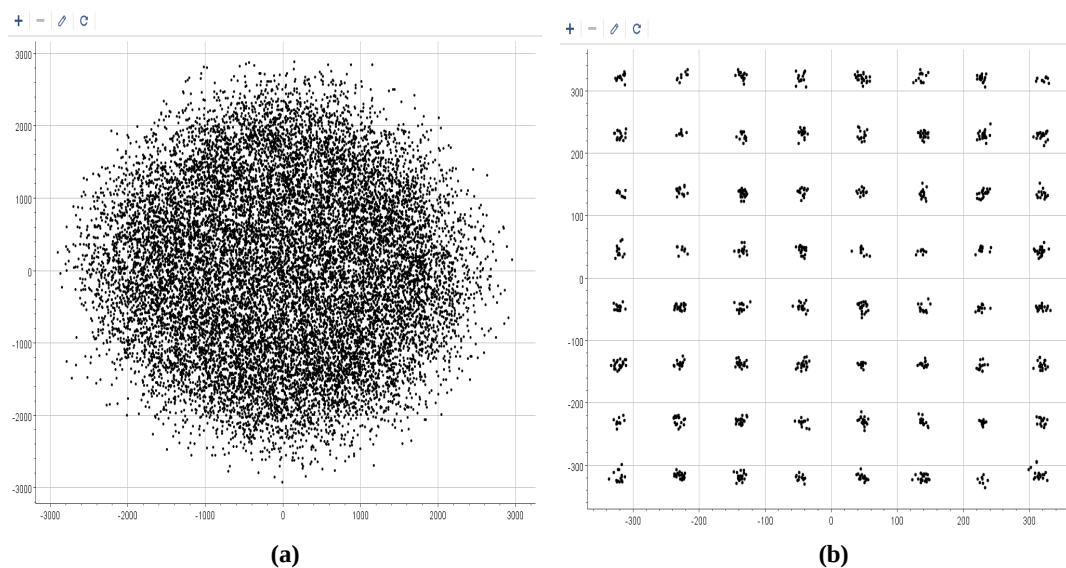


Figure 13. The constellation diagram before (a) and after (b) the timing recovery process of the received signal.

Subsequently, a long-distance THz communication field test was carried out to further validate the loop's performance in practical application scenarios. In this test, the proposed timing recovery loop was integrated as a key stage in the receiver signal processing chain to jointly complete system demodulation. The received constellation diagrams and BER of the system were statistically analyzed through three independent test runs, with the measured results presented in Figure 14. Figure 14a shows the physical photograph of the receiver hardware platform deployed in the field test, while Figure 14b depicts the three sets of statistical BER results from the independent tests. The average BER across the three measurements is 3.75×10^{-8} , and the post-demodulation constellation diagram exhibits clear clustering of 64QAM symbol points. These results fully demonstrate the superior performance, excellent stability, and reliable practicality of the proposed timing recovery loop in real long-distance THz communication systems.

Table 2 presents the FPGA resource utilization results and comparison with other studies under the same environment with geometric mapping. It can be seen our architecture shows a consistent advantage in LUT and BRAM utilization. Compared with existing timing recovery loops, this loop avoids a large number of FFT/IFFT operations or interpolation, reduces the computational complexity of the system. The advantage of the proposed loop is even more significant as the number of parallel paths N increases.

Table 2. Resource Occupation Comparison.

Logic Utilization	Proposed	In [5] Proposed	In [28] Proposed
DSPs	2241	1060	837
Slice LUTs	67,806	81,000	74,172
Slice Registers	66,061	46,900	104,600
BRAM	16	153	380

Table exists mapping estimation.

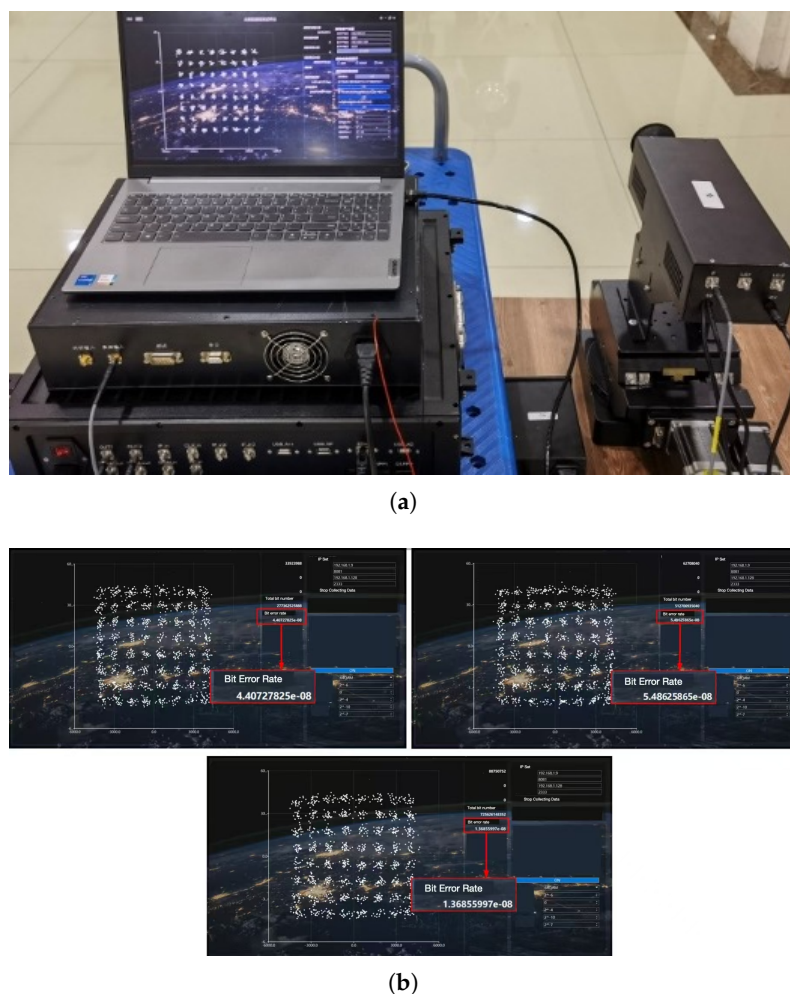


Figure 14. (a) is the topology of the platform. (b) is the photograph of platform.

6. Conclusions

This paper proposes a joint parallel timing recovery loop based on a time-domain feedback architecture, with its performance comprehensively validated through numerical simulations and hardware experiments. Under the condition of an AWGN channel with a timing sampling deviation factor $P = 0.99975$ introduced, the proposed loop was compared with conventional parallel loops, and the simulation results demonstrate that its BER aligns well with the theoretical curve, incurring a performance loss of less than 0.1 dB with 64QAM modulation. For hardware validation, the loop was successfully deployed on the Xilinx UltraScale+ VCU118 (XCVU9P) Evaluation Kit and integrated into a 220 GHz communication system. System-level tests confirm its capability to support real-time demodulation of 15 Gbps 64QAM signals. As a key stage in the receiver, it enables the system to achieve an average BER of 3.75×10^{-8} , verifying the loop's excellent stability and practicality in supporting end-to-end system performance. Notably, the architecture avoids extensive FFT/IFFT operations and interpolation, significantly reducing hardware resources. However, higher synchronization precision requires storing more MMF coefficients, and its performance is achieved at the cost of ROM resources. Combined with its inherent high parallelism, the proposed loop effectively addresses the resource constraints of on-board systems and meets the evolving demands of next-generation large-bandwidth, high-speed THz communication.

Author Contributions: Conceptualization, F.W. and W.W.; methodology, F.W.; software, F.W. and X.L.; validation, F.W., L.X. and W.W.; formal analysis, X.L.; investigation, H.Z.; resources, F.W.;

data curation, W.W.; writing—original draft preparation, X.L.; writing—review and editing, F.W.; visualization, L.X.; supervision, H.Z.; project administration, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors would like to express their gratitude to colleagues who provided valuable insights and technical assistance during the course of this research.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

THz	Terahertz
MMF	Modified matched filter
TED	Timing error detector
DCs	Data caches
BER	Bit error rate
FPGA	Field-programmable gate array
IF	Intermediate frequency
ADC	Analog to Digital Converter
NCO	Numerical control oscillator
SRRC	Square-root-raised-cosine
LF	Loop filter

References

1. Wang, Y.; Qin, Z.; Ma, Y.; Qi, Y.; Wu, N. A Parallel Carrier Recovery Scheme for an 8Gbps Terahertz Communication System. In Proceedings of the 2020 IEEE/CIC International Conference on Communications in China (ICCC), Chongqing, China, 9–11 August 2020; pp. 1308–1313.
2. Heffernan, B.M.; Kawamoto, Y.; Maekawa, K.; Greenberg, J.; Amin, R.; Hori, T.; Tanigawa, T.; Nagatsuma, T.; Rolland, A. 60 Gbps real-time wireless communications at 300 GHz carrier using a Kerr microcomb-based source. *APL Photonics* **2023**, *8*, 066106. [\[CrossRef\]](#)
3. Cao, H.; Wang, Z.; Wang, W.; Zhang, Z.; Li, W.; Zhou, T.; Zhou, H.; Dong, Y.; Wang, H.; Yang, Q.; et al. 8K Real-Time Video-Transmission Using PDM and FDM in 220-GHz Band Based on Polarization Crosstalk Model and Optimization Strategy. *IEEE Commun. Mag.* **2024**, *62*, 86–93. [\[CrossRef\]](#)
4. Zhang, T.; Zhang, H.; Zhu, H.; Lin, J.; Huang, X.; Du, J.; Yang, Y. A 50 Gbps Real-Time Wireless Communication System at 252 GHz Using FPGA Baseband Modem. *IEEE Trans. Terahertz Sci. Technol.* **2025**, *15*, 1055–1067. [\[CrossRef\]](#)
5. Zhang, N.; Feng, J.; Zhang, P.; Gong, F. Parallel doubly fed symbol timing recovery algorithm and fpga implementation for burst broadband satellite access. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2023**, *31*, 2102–2111. [\[CrossRef\]](#)
6. Qu, F.; Li, Z.; Zhang, M.; Tu, X.; Wei, Y. A Carrier-Based Gardner Timing Synchronization Algorithm for BPSK Signal in Maritime Communication. *J. Mar. Sci. Eng.* **2023**, *11*, 829. [\[CrossRef\]](#)
7. Dyadyuk, V.; Bunton, J.D.; Pathikulangara, J.; Kendall, R.; Sevimli, O.; Stokes, L.; Abbott, D.A. A multigigabit millimeter-wave communication system with improved spectral efficiency. *IEEE Trans. Microw. Theory Tech.* **2007**, *55*, 2813–2821. [\[CrossRef\]](#)
8. Wang, C.; Lin, C.; Chen, Q.; Lu, B.; Deng, X.; Zhang, J. A 10-Gbit/s wireless communication link using 16-QAM modulation in 140-GHz band. *IEEE Trans. Microw. Theory Tech.* **2013**, *61*, 2737–2746. [\[CrossRef\]](#)
9. Mueller, K.; Muller, M. Timing recovery in digital synchronous data receivers. *IEEE Trans. Commun.* **1976**, *24*, 516–531. [\[CrossRef\]](#)
10. Franks, L. Carrier and bit synchronization in data communication—a tutorial review. *IEEE Trans. Commun.* **1980**, *28*, 1107–1121. [\[CrossRef\]](#)
11. Gardner, F. A BPSK/QPSK timing-error detector for sampled receivers. *IEEE Trans. Commun.* **1986**, *34*, 423–429. [\[CrossRef\]](#)
12. Oerder, M.; Meyr, H. Digital filter and square timing recovery. *IEEE Trans. Commun.* **1988**, *36*, 605–612. [\[CrossRef\]](#)

13. Gardner, F.M. Interpolation in digital modems. I. Fundamentals. *IEEE Trans. Commun.* **1993**, *41*, 501–507. [\[CrossRef\]](#)
14. Farrow, C.W. A continuously variable digital delay element. In Proceedings of the 1988 IEEE International Symposium on Circuits and Systems, Espoo, Finland, 7–9 June 1988; pp. 2641–2645.
15. Teng, F.; Yang, W.; Yan, J.; Ma, H.; Jiao, Y.; Gao, Z. A Parallel Solution of Timing Synchronization in High-Speed Remote Sensing Data Transmission. *Remote Sens.* **2023**, *15*, 3347. [\[CrossRef\]](#)
16. Raghavan, S.; Thapar, H. On feed-forward and feedback timing recovery for digital magnetic recording systems. *IEEE Trans. Magn.* **1991**, *27*, 4810–4812. [\[CrossRef\]](#)
17. Tuukkanen, V.; Vesma, J.; Renfors, M. Combined interpolation and maximum likelihood symbol timing recovery in digital receivers. In Proceedings of the ICUPC 97—6th International Conference on Universal Personal Communications, San Diego, CA, USA, 12–16 October 1997; Volume 2, pp. 698–702.
18. Zhu, W.P.; Yan, Y.; Ahmad, M.O.; Swamy, M.S. Feedforward symbol timing recovery technique using two samples per symbol. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2005**, *52*, 2490–2500.
19. Shi, K.; Serpedin, E. Fast timing recovery for linearly and nonlinearly modulated systems. *IEEE Trans. Veh. Technol.* **2005**, *54*, 2017–2023. [\[CrossRef\]](#)
20. Pelet, E.R.; Salt, J.E. Timing jitter analysis of optimum non-data-aided symbol synchronizer for QAM. *IEEE Trans. Commun.* **2010**, *58*, 1247–1255. [\[CrossRef\]](#)
21. Barletta, L.; Magarini, M.; Scardoni, F.; Spalvieri, A. Impact of loop delay on the performance of Gardner timing recovery. *IEEE Photonics Technol. Lett.* **2013**, *25*, 1797–1800. [\[CrossRef\]](#)
22. Schmidt, D.; Lankl, B. Parallel architecture of an all digital timing recovery scheme for high speed receivers. In Proceedings of the 2010 7th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP 2010), Newcastle Upon Tyne, UK, 21–23 July 2010; pp. 31–34.
23. Schmidt, D.; Lankl, B.; Fischer, J.K.; Hilt, J.; Schubert, C. Real-time implementation of a parallelized feedforward timing recovery scheme for receivers in optical access networks. In Proceedings of the 2012 38th European Conference and Exhibition on Optical Communications, Amsterdam, The Netherlands, 16–20 September 2012; pp. 1–3.
24. Hao, X.; Wu, Q.; Wang, Z.; Lin, C. Parallel timing synchronization algorithm and its implementation in high speed wireless communication systems. In Proceedings of the 2019 International Conference on Electronics, Information, and Communication (ICEIC), Auckland, New Zealand, 22–25 January 2019; pp. 1–6.
25. Hu, J.; Zhu, L.; Wang, J. The Implementation of High Speed Parallel Timing Synchronization Algorithm Based on FPGA. In Proceedings of the 2018 10th International Conference on Communication Software and Networks (ICCSN), Chengdu, China, 6–9 July 2018; pp. 484–487.
26. Hao, X.; Lin, C.; Wu, Q. A parallel timing synchronization structure in real-time high transmission capacity wireless communication systems. *Electronics* **2020**, *9*, 652. [\[CrossRef\]](#)
27. Wang, Y.; Zhang, Y.; Yang, Z. Multiple-symbol parallel interpolation in all-digital receiver. In Proceedings of the 2009 5th International Conference on Wireless Communications, Networking and Mobile Computing, Beijing, China, 24–26 September 2009; pp. 1–6.
28. Lin, C.; Zhang, J.; Shao, B. A high speed parallel timing recovery algorithm and its FPGA implementation. In Proceedings of the 2011 2nd International Symposium on Intelligence Information Processing and Trusted Computing, Wuhan, China, 22–23 October 2011; pp. 63–66.
29. Srinivasan, M.; Chen, C.C.; Grebowsky, G.; Gray, A. An all-digital, high data-rate parallel receiver. *JPL TDA Prog. Rep.* **1997**, *42*, 131.
30. Wang, Y.; Ding, J.; Zhu, M.; Zhang, L.; Meng, J.; Tong, W.; Xin, Z.; Cai, Y.; Zhang, J.; Hua, B.; et al. 105-Gb/s Photonics-Aided THz Communication with FPGA-Based Low-Complexity Resampling and Equalization. *IEEE Trans. Microw. Theory Tech.* **2025**, *73*, 10700–10710. [\[CrossRef\]](#)
31. Zhang, Z.; Li, Y.; Nima, B. Digital finite impulse response equalizer for nonlinear frequency response compensation in wireless communication. *Electronics* **2023**, *12*, 2010. [\[CrossRef\]](#)
32. Qi, Y.; Li, B.; Qin, Z.; Ma, Y.; Wang, Y.; Wu, N. Low-complexity implementation of parallel receiver for terahertz communication system. In Proceedings of the IET International Radar Conference (IET IRC 2020), Online, 4–6 November 2020; Volume 2020, pp. 84–89.
33. Hedin, G.; Holmes, J.; Lindsey, W.; Woo, K. Theory of false lock in Costas loops. *IEEE Trans. Commun.* **1978**, *26*, 1–12. [\[CrossRef\]](#)
34. Leng, W.; Zhang, Y.; Yang, Z. A modified gardner detector for multilevel PAM/QAM system. In Proceedings of the 2008 International Conference Communications, Circuits and Systems, Xiamen, China, 25–27 May 2008; pp. 891–895.
35. William, H.T.; Shanmugan, K.; Rappaport, T.; Kosbar, K.L. *Principles of Communication Systems Simulation with Wireless Applications*; Prentice Hall: Upper Saddle River, NJ, USA, 2004; Volume 7458.
36. Cui, J.; Song, Y.; Jiang, H.; Wang, C.; Zhang, M.; Liu, G.; Li, D.; Zhao, J.; Liu, J.; Su, Y.; et al. Measurement and Modeling on Terahertz Channel Propagation Through Vegetation. *IEEE Trans. Terahertz Sci. Technol.* **2025**, *15*, 582–595. [\[CrossRef\]](#)

37. Niu, Z.; Zhang, B.; Dai, B.; Zhang, J.; Shen, F.; Hu, Y.; Fan, Y.; Zhang, Y. 220 GHz multi circuit integrated front end based on solid-state circuits for high speed communication system. *Chin. J. Electron.* **2022**, *31*, 569–580. [[CrossRef](#)]
38. Cui, J.; Zhang, Y.; Xia, D.; Xu, Y.; Xiao, F.; Yan, B.; Xu, R. A 220 GHz broadband sub-harmonic mixer based on global design method. *IEEE Access* **2019**, *7*, 30067–30078. [[CrossRef](#)]
39. Lee, S.; Kim, J.; Lee, K.; Song, H.J. 248-GHz subharmonic mixer last transmitter with I/Q imbalance and LO feedthrough calibration. *IEEE Solid-State Circuits Lett.* **2024**, *7*, 159–162. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.