An Improved Square Timing Error Detection Algorithm

Guo Pengfei, Li Jianguo, Liang Bizheng and Li Mucheng

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

June 8, 2018

An Improved Square Timing Error Detection Algorithm

Pengfei Guo, Jianguo Li, Bizheng Liang, and Mucheng Li

School of Information and Electronics, Beijing Institute of Technology,
Beijing, 100081, China

Email: {guopengfei,jianguoli,liangbizheng,limucheng}@bit.edu.cn

Abstract. Timing recovery is one of the most important issues that need to be solved for a receiver. The key of timing recovery is the extraction of timing errors. The timing recovery algorithm adopted by all-digital receivers is expected to be fast and have small computation burden. This paper first introduces the overall structure of the traditional square timing recovery loop, and then proposes an improved square timing error detection algorithm. By using the Goertzel algorithm, this improved algorithm optimizes the step-extracting spectral components at the symbol rate $1/T$. Finally, this paper simulates and analyzes the improved algorithm, as well as parameters' influence on the performance of the algorithm. It is proved that the improved square timing error detection algorithm can greatly reduce the amount of computation under the premise of guaranteeing performance.

Keywords: timing recovery; square timing error detection algorithm; reduce computation

1 Introduction

Timing recovery is essential for the receiver in digital communication system. It means to extract timing information from the received signal to make the receiver sample at the best position of the symbol[1], thereby reducing the bit error rate. In the all-digital receiver, the sampling clock with high stability and high accuracy is independent of the signal clock[2]. When such a sampling clock is used to sample the received signal directly, there will be some timing error because the sampling clock frequency is not synchronized with the symbol rate of the received symbol. And the timing recovery algorithm is to eliminate this timing error.

In general, the timing recovery loop first needs to extract the timing information from the received signal[3], estimate the timing error in combination with the local sampling clock, and then use the estimated value of the timing error to control the interpolator in real-time, including updating the filter's coefficient and determining the specific interpolation time. The output sequence of the interpolator is exactly synchronized with the received signal. That is the process of timing recovery[4]. The expected advantages of the timing recovery algorithm

are as follows: fast synchronization, low computation and easy for digital realization[5]. Based on the actual demand, this paper proposes an improved square timing error detection algorithm, simulates and analyzes the algorithm. The algorithm can largely reduce computation under the premise of good performance and is advantageous for digital realization.

2 Square Timing Recovery Loop

The square timing recovery loop is an efficient, stable and non-data-aided(NDA) timing recovery loop[6]. This loop does not require feedback of the signal and directly extracts the timing information from the received signal, so that the timing error can be quickly captured. And square timing error detection algorithm is quite independent of the carrier. The timing recovery would not be influenced by the frequency offset and phase offset of the carrier, so it could be prior to carrier synchronization. Thus, timing recovery has been widely used in all-digital receivers[7]. The square timing recovery loop mainly consists of match filter, sampling unit, squaring timing error detector, timing control unit and interpolating filter. Passing through the match filter and the sampling unit, the timing error in the received signal is extracted by the squaring timing error detector. Together with the sampling sequence, the timing error decides the best position for sampling in the interpolation filter to realize timing recovery[8].

Fig. 1. The Square Timing Recovery Loop

Fig. 1 shows the whole structure of the square timing recovery loop.

In this loop, the squaring timing error detector plays a key role. And it applies square timing error detection algorithm, whose principle is that the received baseband signal can generate baseband sampling sequence after passing match filter and sampling unit, then by modular squaring the sampling sequence, the frequency spectrum must include the spectral component at the symbol rate $1/T$. The timing error of the sampling point is expressed in the frequency domain as phase rotation of the spectral component. Martin Oerder proved that the normalized phase of that spectral component is an unbiased estimate of the normalized timing error in his paper[4]. The following part explains the specific

process of the loop. The signal received can be written as:

$$r(t) = \sum_{n=-\infty}^{\infty} a_n g_T(t - nT - \varepsilon(t)T) + n(t) = u(t) + n(t) \quad (1)$$

Where a_n is the transmitted symbol sequence, $g_T(t)$ is the transmission signal pulse, $g_T(t)$ is the duration of the symbol, $n(t)$ is the channel noise which is assumed to be white and Gaussian distribution with power density N_0 , and $\varepsilon(t)$ is an unknown, slowly varying timing error. Timing recovery means, to some extent, an accurate estimate of $\varepsilon(t)$. Since $\varepsilon(t)$ varies very slowly, we can assume that $\varepsilon(t)$ will remain constant for a period of time, so in the digital realization we can group the data and then process the received signals section by section. The analysis followed assumes that the received signals are divided into groups and every group contains L symbols. After passing the matched filter (impulse response $g_R(t)$), $r(t)$ is sampled at the rate T/N , then we have samples

$$\tilde{r}_k = \tilde{r}(kT/N) = \sum_{n=-\infty}^{\infty} a_n g\left(\frac{kT}{N} - nT - \varepsilon T\right) + \tilde{n}\left(\frac{kT}{N}\right) \quad (2)$$

In the equation, $g(t) = g_T(t) * g_R(t)$.

The sequence is then sent to the square timing error detector, which applies square timing error detection algorithm. Fig. 2 shows how the algorithm works.

Fig. 2. Square Timing Error Detection Algorithm

First, the sampling sequence is modular squared, then we have

$$x_k = \left| \sum_{n=-\infty}^{\infty} a_n g\left(\frac{kT}{N} - nT - \varepsilon T\right) + \tilde{n}\left(\frac{kT}{N}\right) \right|^2 \quad (3)$$

In order to extract the spectral components at the symbol rate $1/T$, the DFT of x_k should be computed to get the complex Fourier coefficient at the symbol rate. With each group's length as L , and oversampling rate as N , the L -th sample in the LN -point DFT of x_k represents the symbol rate. Then the spectral component can be expressed as

$$X_m = \sum_{k=mLN}^{(m+1)LN-1} x_k e^{-j2\pi k/N} \quad (4)$$

To calculate the unbiased estimate of the normalized timing offset is to figure out the normalized phase of X_m , namely

$$\hat{\varepsilon}_m = -1/2\pi \arg(X_m) \quad (5)$$

The estimate of timing error $\hat{\varepsilon}_m$, along with the sampling sequence, will lead to timing recovery after passing the timing control unit and interpolation filter[9].

3 Square Timing Error Detection Algorithm

As mentioned above, the normalized phase of the spectral component at the symbol rate is an unbiased estimate of the normalized timing error. It is the spectral component at the symbol rate that matters; in another word, it is the L -th sample in the LN -point DFT of x_k that needs to be calculated. If FFT algorithm was used here like the traditional way, a lot of computation results were invalid. So, this paper proposes an improved square timing error detection algorithm which uses the Goertzel algorithm to detect the information at a certain frequency in the signal. That how Goertzel algorithm improves DFT will be analyzed in detail in this part[10]. Suppose that x_n is a sample sequence with a length of N , $n \in [0, N-1]$, and n is an integer. Then the DFT of $x[n]$ is defined as

$$X_k = \sum_{n=0}^{N-1} x_n w_N^{kn}, k = 0, 1, 2, \dots, N-1 \quad (6)$$

Wherein $W_N = e^{\frac{-j2\pi}{N}}$. The DFT summation can be written in another way as

$$X_k = \sum_{n=0}^{N-1} (w^{-k})^{N-n} \cdot x_n \quad (7)$$

It can be found that the powers of W in the DFT summation can be treated as the power terms in a polynomial, with $x[n]$ as the multiplier coefficients. Then this polynomial can be evaluated in the nested form:

$$X_k = (\dots(((W^{-k} \cdot x_0 + x_1) \cdot W^{-k} + x_2) \cdot W^{-k} + x_3) \dots + x_{N-1}) \cdot W^{-k} \quad (8)$$

As the nested form is evaluated term by term, it produces a sequence of intermediate results y :

$$\begin{aligned} y_{-1} &= 0 \\ y_0 &= W^{-k} y_{-1} + x_0 \\ y_1 &= W^{-k} y_0 + x_1 \\ &\dots \\ y_{N-1} &= W^{-k} y_{N-2} + x_{N-1} \\ y_N &= W^{-k} y_{N-1} \end{aligned}$$

The final result y_N equals the desired DFT sum. The following iteration is a summary of the above process:

$$\begin{aligned} y_n &= W^{-k} y_{n-1} + x_n, n = 0, \dots, N-1 \\ y_N &= W^{-k} y_{N-1} \end{aligned}$$

The iteration equation in the discrete domain is equivalent to the following transfer function:

$$\frac{Y}{X} = \frac{1}{1 - W^{-k} z^{-1}} \quad (9)$$

So the transfer function of the Goertzel filter is:

$$H_k[z] = \frac{1}{1 - W_N^{-k} z^{-1}} = \frac{1 - W_N^k z^{-1}}{1 - 2 \cos[2\pi k/N] z^{-1} + z^{-2}} \quad (10)$$

Multiply W_N^{-k} will lead to the final result. The block diagram of the Goertzel algorithm is as Fig. 3 The realization of the filter can be expressed as

$$s_k[n] = x[n] + 2 \cos[2\pi k/N] s_k[n-1] - s_k[n-2] \quad (11)$$

$$X[k] = W_N^{-k} \cdot y_k[N-1] = W_N^{-k} \cdot (s_k[N-1] - W_N^k s_k[N-2]) \quad (12)$$

And $s_k[-1] = s_k[-2] = 0$. The filter is composed of two parts; the formula (11) means the first part will be recursively calculated for N times, and the formula (12) means the latter part only calculates once in the end. $X[k]$ will be figured out.

Fig. 3. The Block Diagram of Goertzel Algorithm

Using Goertzel algorithm, the improved square timing error detection algorithm is advantageous in the following aspects:

1. Reducing computation. It uses a recursive real arithmetic structure to avoid intricate complex operations;
2. Saving storage space. The required storage space is small, and only the corresponding coefficients and some intermediate results are stored;
3. Improving real-time performance. The Goertzel algorithm can immediately process every sampling result, improving the calculation efficiency while the FFT needs all the values of the data block before processing;
4. There is no limit on the number of sampling points N , which can be arbitrarily selected.

The key advantage of the improved square timing error detection algorithm lies in the reduction of computation. Here is a detailed analysis of the computation. From the realization of the filter, we can see that for each sample value, the formula (11) calculates once recursively, and calculates for N times all together. And the formula (12) only calculates once in the end after recursive calculation. Every operation of the formula (11) needs two real additions and one real multiplication, and one of the multipliers in real multiplication is constant $2 \cos[2\pi k/N]$ (this process also saves storage space). Every operation of the formula (12) needs one real addition and two real multiplications. Therefore, the Goertzel algorithm requires a total of $2N+1$ real additions and $N+2$ real multiplications to finish one operation. The computation of Goertzel algorithm is far less than that of FFT since it requires a large amount of unnecessary operations. So we only compare the computation of Goertzel algorithm with that of DFT. When it comes to DFT, one operation needs real multiplying complex for N times and complex addition for $N-1$ times, namely $2N$ real multiplications and $2N-2$ real additions. Table 1 shows the reduction of computation.

Table 1. Comparison Between the Computation of Goertzel Algorithm and DFT

	Real Addition	Real Multiplication
Goertzel Algorithm	$2N+1$	$N+2$
DFT	$2N-2$	$2N$

The table shows that Goertzel Algorithm reduces real multiplication by almost a half and greatly improved the performance of square timing error detection algorithm.

4 Simulation Results And Analysis

First, the BER performance of the improved square timing error detection algorithm is simulated and analyzed. The simulation conditions are set as follows: timing error is $0.25T$; every group's length L is 64; all the received symbols are divided into 2000 groups; QPSK modulation; cubic interpolation is used in interpolation filter; oversampling rate is 4. And comparing theoretical BER curve with improved square timing recovery loop BER curve when E_b/N_0 is from -2 to 7 leads to the simulation results seen in Fig.4 .

Fig. 4. BER Performance of the Improved Square Timing Recovery Loop

It can be drawn from the Fig.4 that after timing recovery performed by the improved square timing error detection algorithm, the BER performance is extremely close to the theoretical BER performance, proving that improved square timing error detection algorithm can realize timing recovery. Second, L is an important parameter in the improved square timing error detection algorithm. Based on the analysis above, the value of L does not influence the overall computation. Therefore, the following analysis focuses on the impact on the variance of timing estimate caused by different value of L . Suppose that L is 4, 8, 16, 32, 64, 128, 256, and 512 while other conditions remain the same, and explore the connection between L and variance of timing estimate.

Fig. 5. Connection Between L and Variance of Timing Estimate

Fig. 5 shows that as L increases, the variance of timing estimate reduces significantly. When L is big enough, the timing error stemmed from the extracted spectral components at the symbol rate $1/T$ will be accurate and stable. But when L is too large, too many symbols are needed before processing, resulting in long-time delay. So, L should be in a reasonable range, for example 64, to meet the requirements of stability and real-time performance.

5 Conclusion

The timing recovery algorithm adopted by all-digital receivers is expected to be fast and have small computation burden. After analyzing the square timing recovery loop, this paper proposes an improved square timing error detection algorithm. The algorithm can accurately realize timing recovery while greatly reducing computation at the same time.

References

1. D'Andrea, A.N., Luise, M.: Optimization of symbol timing recovery for QAM data demodulators. *IEEE Trans. Commun.* 44, 399–406 (1996)
2. Xia, J., Li, Y., Xu, T., Chen, J., Song, Y.: Timing Recovery Algorithm Selection for Duo-Binary Signal Used in Next Generation 25G-EPON. *IEEE Photonics J.* 9, 1-7 (2017)
3. Gardner, F.M.: A BPSK/QPSK timing-error detector for sampled receivers. *IEEE Trans. Commun.* 34, 423–429 (1986)
4. Oerder, ., Meyr, H.: Digital filter and square timing recovery. *IEEE Trans. Commun.* 36, 605–612 (1988)
5. Scardoni, F., Magarini, M., Spalvieri, A.: Impact of Self Noise on Tracking Performance of Non-Data-Aided Digital Timing. *J Lightwave Technol.* 33, 3755–3762 (2015)
6. Ayat, M., Mirzakuchaki, S., Beheshti-Shirazi, A.A.: Design of new digital blind feedforward timing recovery algorithm for linear modulations in digital communication receivers. *IET Commun.* 10, 1736–1743 (2016)
7. Gardner, F.M.: Interpolation in digital modems. I. Fundamentals. *IEEE Trans. Commun.* 41, 501–507 (1993)
8. Johansson, H., Gustafsson, O.: Linear-phase FIR interpolation, decimation, and mth-band filters utilizing the farrow structure. *IEEE T CIRCUIT-I.* 52, 2197–2207 (2005)
9. Lamb, D., Chamon, F.O., Nascimento, V.H.: Efficient filtering structure for spline interpolation and decimation. *Electron Lett.* 52, 39-41 (2016)
10. Ghouse, M.A.: Modular Realizations of the DFT From the Goertzel DFT Modules (of Smaller Dimension). *IEEE Trans. Signal Process.* 41, 427–(1993)