

A Novel Parallel Timing Synchronization Scheme for High-Speed Receivers

Marco Morini¹, Alessandro Ugolini¹, Giulio Colavolpe¹,
Tommaso Foggi¹, and Armando Vannucci¹

Abstract—We address the parallel implementation of a closed-loop symbol timing synchronizer in digital receivers. Starting from a serial timing recovery loop, we propose a low-complexity parallel architecture which, unlike parallel schemes available in the literature, employs a single numerically controlled oscillator, and is practically suitable for high-speed receivers. Numerical simulations are carried out to compare the performance of serial and parallel implementations in terms of bit error rate. Results show that the proposed architecture achieves the same performance as the serial algorithm and is robust enough to ensure good performance also with high order modulations, which are critical for modern high throughput applications.

Index Terms—Timing synchronization, parallel implementation.

I. INTRODUCTION

THE design of parallel architectures in modern communication systems is becoming more and more important due to the rapidly increasing data rates, that are eventually limited by the processing speed of the hardware components [1]. In these architectures, the received signal is sampled by a high-rate analog-to-digital converter (ADC), followed by serial to parallel conversion, so that the samples can be processed in parallel at a lower clock rate.

In this letter, we consider the problem of high-speed parallel timing synchronization, which has been widely studied in the literature. In [2], the authors implemented a parallel Oerder and Meyr timing error detector (TED) whose architecture, however, requires an oversampling factor equal to 4, thus demanding considerable hardware resources. Moreover, the performance analysis was limited to 16-QAM, without investigating higher-order modulations. A parallel architecture, designed for optical coherent receivers, can be found in [3], where the algorithm is tested in the presence of sampling frequency offsets. Therein, only one error signal is exploited by the loop filter, thus increasing the convergence time. In [4], a parallel timing synchronization algorithm is proposed

for 16-QAM only, for which the parallel model shows an implementation loss compared to the serial case. Other related works are, for example [5], [6], and [7], where the authors specifically address FPGA implementations and high speed processing. All of the cited architectures, however, adopt multiple numerically-controlled oscillators (NCOs) to adjust the timing estimate, and the performance evaluation is limited to lower order constellations (at most 16-QAM).

Motivated thus by the complexity of hardware implementation for symbol timing synchronization at high baud rate, we present a novel parallel timing recovery architecture suitable for high-speed receivers and able to operate with different modulation formats. The novelties and most important aspects of our proposed architecture can be summarized as follows:

- 1) Unlike other works in the literature, our parallel scheme employs a single NCO, thus having the same complexity as the serial algorithm, since the components of the circuit are exactly the same. The only complexity increase is the replication of the branches in order to process samples in parallel, but everything is controlled by a single NCO, which is the main source of complexity in a hardware implementation, and which represents the main advantage and novelty of our solution with respect to other parallel architectures proposed in the literature.
- 2) Being the algorithm exactly the same as the serial one, there will be no performance losses, as we will demonstrate through numerical simulations.
- 3) We also address the problems arising from the hardware implementation of a division, by introducing an approximation which ensures very limited performance losses.
- 4) Finally, unlike what is done in the literature, we present results for high order modulations, which are critical to achieve high throughput in modern communications, showing that the proposed architecture is robust enough to ensure excellent performance.

This letter is organized as follows. In Section II, the system model is presented and in Section III the symbol timing recovery scheme with its blocks is described. In Section IV, the proposed parallel architecture is introduced, for which numerical simulations are reported in Section V, to compare the performance of serial and parallel algorithms. Finally, some conclusions are drawn in Section VI.

II. SYSTEM MODEL

We assume a linear modulation transmitted over an additive white Gaussian noise channel. The received signal before matched filtering (MF) can be expressed as

$$r(t) = \sum_i a_i p(t - iT - \tau) e^{j(2\pi i \Delta f T + \theta)} + w(t), \quad (1)$$

Manuscript received 21 May 2024; revised 4 July 2024; accepted 11 July 2024. Date of publication 15 July 2024; date of current version 12 September 2024. This work is funded in part by the European Space Agency, ESA-ESTEC, Noordwijk, The Netherlands, under contract n. 4000131060/20/NL/FE (“Multi-GBPS Broadband Terminal for Extremely High Throughput Satellite Systems”). This research was granted in part by University of Parma through the action Bando di Ateneo 2022 per la ricerca co-funded by MUR-Italian Ministry of Universities and Research - D.M. 737/2021 - PNR - PNRR - NextGenerationEU. The associate editor coordinating the review of this letter and approving it for publication was M. S. Bashir. (Corresponding author: Alessandro Ugolini.)

The authors are with the Department of Engineering and Architecture, University of Parma, 43124 Parma, Italy (e-mail: marco.morini@unipr.it; alessandro.ugolini@unipr.it; giulio.colavolpe@unipr.it; tommaso.foggi@unipr.it; armando.vannucci@unipr.it).

Digital Object Identifier 10.1109/LCOMM.2024.3428911

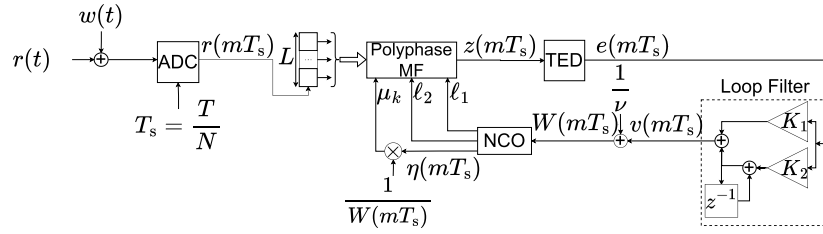


Fig. 1. Block diagram of the considered serial symbol timing recovery scheme.

where $\{a_i\}$ are the transmitted symbols belonging to an M -ary complex constellation, $p(t)$ is the shaping pulse, T is the symbol interval, θ is a phase offset, and $w(t)$ is a complex Gaussian noise process. Parameters τ and ΔfT identify the time offset and the normalized residual carrier frequency offset, respectively, where τ is assumed to be in the interval $-\frac{T}{2} \leq \tau \leq \frac{T}{2}$. The received signal is sampled at a fixed rate $1/T_s$ that is *asynchronous* with respect to the symbol rate $1/T$, i.e., the oversampling factor $\nu = T/T_s$ is in general non-integer (in our simulations, we assumed it to be $\nu = 2.25$). The *asynchronous* samples thus obtained are not aligned with the maximum eye opening. The role of symbol timing synchronization is thus to compute samples that are aligned with the optimum sampling instants.

III. SERIAL SYMBOL TIMING RECOVERY

The block diagram of the closed-loop symbol synchronizer is reported in Fig. 1 [8]. After ADC conversion, with rate $1/T_s$, a new sample enters a shift register with depth L . With the same rate, the entire content of the register is filtered by a polyphase filter, designed to efficiently combine MF and interpolation [9]. The *polyphase MF* outputs samples with rate $2/T$. This can be accomplished by using proper control signals, $\ell_1(mT_s)$ and $\ell_2(mT_s)$, whose dependence on T_s will be omitted for the sake of simplicity. These signals are generated by the NCO block, and their purpose is to label each sample at the output of the interpolator either as an *optimum* sample, when $(\ell_1, \ell_2) = (0, 1)$, i.e., it is computed at instant $kT + \hat{\tau}$, or as a *valid* sample, when $(\ell_1, \ell_2) = (1, 1)$, i.e., it is computed at instant $kT - T/2 + \hat{\tau}$.

Since we assumed a fractional oversampling factor, some samples should be discarded: this occurs when neither the sample at $kT + \hat{\tau}$ nor that at $kT - T/2 + \hat{\tau}$ are present in the interval $[mT_s, (m+1)T_s]$ and, as a consequence, the NCO generates the control signal $\ell_2 = 0$ to notify the *polyphase MF* block that no sample has to be generated at its output. Note that the control signals ℓ_1 and ℓ_2 should also be available to the TED block, which needs to know if the incoming sample is either *optimum* or *valid*.

Going into the details of the block diagram in Fig. 1, the main features of each block, i.e., the polyphase MF, the TED, the loop filter and the NCO, are described hereafter.

1) *Polyphase MF*: timing adjustment is performed by filtering and interpolating the asynchronous samples $r(mT_s)$ at the output of the ADC, to obtain the samples at the optimal sampling instants ($y(kT + \hat{\tau})$) and the valid samples ($y(kT - T/2 + \hat{\tau})$). The sequence at the output of the interpolator can

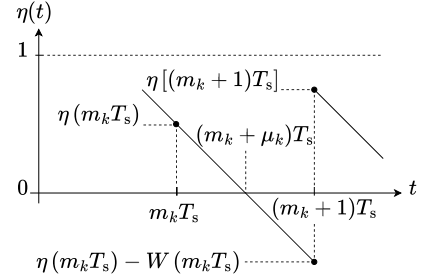


Fig. 2. Relation between the NCO register and the optimal sampling instant.

be expressed as

$$z(mT_s) = \ell_2 y \left[\left(k - \frac{\ell_1}{2} \right) T + \hat{\tau} \right]. \quad (2)$$

When the k -th filtered sample is within the interval $[mT_s, (m+1)T_s]$, the sample index m is called the k -th *basepoint index*, and is denoted by m_k . The optimal sampling instant exceeds $m_k T_s$ by some fraction of the symbol time that we denote as $\mu_k T_s$, where μ_k is the k -th *fractional index*. Therefore, the *basepoint index* m_k identifies the closest sample preceding the maximum eye opening instant, which can thus be expressed, using the *fractional index* μ_k , as $kT + \hat{\tau} = (m_k + \mu_k)T_s$. This situation is represented in Fig. 2.

2) *TED*: the filtered samples at the output of the interpolator can be used to generate an error signal according to a properly selected TED. Without loss of generality, we will consider the non-data-aided Gardner algorithm [10], which requires only two samples per symbol, corresponding to the rate available at the output of the *polyphase MF*. The error signal is given by [10]

$$e(mT_s) = \Re \{ y^* ((k-1/2)T + \hat{\tau}) \cdot [y((k-1)T + \hat{\tau}) - y(kT + \hat{\tau})] \} \quad (3)$$

if $(\ell_1, \ell_2) = (0, 1)$, while it is $e(mT_s) = 0$ otherwise.

3) *Loop Filter*: the error signal is filtered by a second order proportional-plus-integrator loop filter to track out the symbol clock frequency offset [11]. The noise equivalent bandwidth of the filter is a function of the loop filter constants [12]. The loop filter output, denoted by $v(mT_s)$, controls the amount by which the NCO decrements.

4) *NCO*: the role of the NCO is to select the *polyphase MF* depending on the k -th *fractional index* $\mu_k T_s$, and label the filtered samples as *optimum* or *valid*. It consists of a modulo-1 decrementing counter, called $\eta(mT_s)$, that is updated as [12]

$$\eta[(m+1)T_s] = \eta(mT_s) - W(mT_s) \mod 1, \quad (4)$$

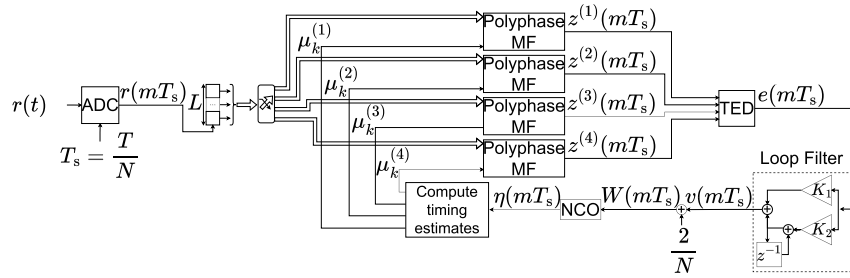


Fig. 3. Block diagram for parallel symbol timing recovery with $P = 4$.

where $W(mT_s) = 1/\nu + v(mT_s)$ is the control word at the output of the loop filter. The *fractional index* $\mu_k T_s$ is computed each time an underflow of the NCO occurs. When the NCO underflows, it also generates the control signals $(\ell_1, \ell_2) = (0, 1)$ to notify the *polyphase MF* that a new optimum sample should be computed. The sample index is the actual *basepoint index* m_k , while the value of the *fractional index* $\mu_k T_s$ can be computed from the content of the NCO register at instant $m_k T_s$, i.e., the time instant immediately preceding the interpolation instant $(m_k + \mu_k) T_s$, as can be seen from Fig. 2. From simple geometrical considerations, it follows that [12]

$$\mu_k T_s = \frac{\eta(mT_s)}{W(mT_s)}. \quad (5)$$

When in the interval $[mT_s, (m+1)T_s]$ the NCO does not underflow, it generates the control signals $(\ell_1, \ell_2) = (1, 1)$ to notify the *polyphase MF* that a valid sample should be computed. Finally, if the underflow does not occur neither in the interval $[mT_s, (m+1)T_s]$ nor in the following interval, the NCO generates the control signal $\ell_2 = 0$ to notify the *polyphase MF* that no samples should be computed.

As said, the entire digital timing recovery scheme operates with a clock rate $1/T_s$. However, for a proper computation of the error signal, at the output of the interpolator we need a new sample every $T/2$. Similarly, the TED needs to compute a new error signal $e(mT_s)$ every T , while the loop filter has to be activated every T_s .

IV. PARALLEL SYMBOL TIMING RECOVERY

The block diagram of the proposed architecture for the parallel symbol timing recovery algorithm is reported in Fig. 3, for a parallelization factor $P = 4$.

After ADC conversion, with rate $1/T_s$, a new sample enters a shift register with depth L . With the same rate, a selector extracts all the samples from the register, and provides them as input to one of the *polyphase MFs*. Then, with rate $1/(T_s \cdot P)$, the P *polyphase MFs* are activated to produce the corresponding filtered samples, which are then labeled either as *optimum* or *valid* by the NCO.

Note that the samples at the *polyphase MFs* outputs are characterized by a rate equal to $2/T$, since the Gardner TED requires two samples per symbol. This is possible thanks to the introduction of P sets of control signals, similarly to what has been described in Sec. III, denoted by $(\ell_1^{(p)}, \ell_2^{(p)})$

($p = 1, \dots, P$), not reported in Fig. 3 for the sake of visual clarity, each provided to the p -th *polyphase MF*.

The control signals are generated by the NCO, as described below, and their purpose is to label each sample at the output of the *polyphase MFs*, either as an *optimum* sample or a *valid* sample. Note that, in the parallel architecture, at the output of the bank of *polyphase MFs* we have P samples, each corresponding to a different time instant. In particular, *optimum* samples are associated either at instants $kT + \hat{\tau}$ or $(k-1)T + \hat{\tau}$, while *valid* samples are associated either at instants $(k - \frac{1}{2})T + \hat{\tau}$ or $(k - \frac{3}{2})T + \hat{\tau}$. Control signal $\ell_1^{(p)}$ identifies the instant corresponding to the sample at the output of the p -th *polyphase MF*. In order to consider all the possible time instants, signal $\ell_1^{(p)}$ can take discrete values in $[-2, 3]$.

Therefore, when $\ell_2^{(p)} = 1$ and $\ell_1^{(p)}$ is even, the sample is *optimum*, when $\ell_2^{(p)} = 1$ and $\ell_1^{(p)}$ is odd, the sample is *valid*, while when $\ell_2^{(p)} = 0$ the sample at the output of the p -th *polyphase MF* is not computed, regardless of the value of $\ell_1^{(p)}$. Note that, when $\ell_1^{(p)}$ is negative, the sample at the output of the p -th *polyphase MF* will be used in the computation of the error signal at the next clock cycle.

The functional blocks are conceptually similar to those in Fig. 1, while the operational differences are further detailed in the points that follow, except for the loop filter, which is exactly the same as in Sec. III.

1) *Polyphase Filter*: interpolation is performed by a bank of P parallel *polyphase MFs*, designed as described in Sec. III, which, every $T_s \cdot P$, output the corresponding samples, that are then labeled either as *optimum* or *valid* by the NCO. The total number of filtered samples, and therefore of timing estimates, that are computed every $T_s \cdot P$, is equal to the number of *optimum* and *valid* samples. The sample at the output of the p -th *polyphase MF* can be expressed as

$$z^{(p)}(mT_s) = \ell_2^{(p)} y \left[\left(k - \frac{\ell_1^{(p)}}{2} \right) T + \hat{\tau} \right], \quad (6)$$

which is identical to Eq. (2) when $P = 1$.

We define by $\mathcal{I}_p = [(m + (p-1))T_s, (m+p)T_s]$, with $p = 1, \dots, P$, the set of P intervals in which a new sample can be computed. The sample in the p -th interval will be provided by the p -th *polyphase MF*.

When the k -th filtered sample is within the p -th interval of \mathcal{I}_p , the sample index $m_k^{(p)} = [m + (p-1)]$ is called the k -th *basepoint index*. The optimal sampling instant exceeds $m_k^{(p)} T_s$ by some fraction of the symbol time, $\mu_k^{(p)} T_s$, where $\mu_k^{(p)}$ is

called the k -th *fractional index*. Therefore, the *basepoint index* $m_k^{(p)}$ identifies the closest sample preceding the maximum eye opening instant, which can thus be expressed, using the *fractional index* $\mu_k^{(p)}$, as $[k + (p-1)T + \hat{\tau}] = (m_k^{(p)} + \mu_k^{(p)})T_s$.

2) *TED*: the expression of the error signal depends on the number of *optimum* samples at the output of the *polyphase MFs*. It turns out that we can have either 1 or 2 *optimum* samples. In the former case, the error signal is computed as in the serial algorithm, according to Eq. (3). In the latter case, instead, the error signal is computed as

$$\begin{aligned} e(mT_s) = & \{ \Re \{ y^* ((k-1/2)T + \hat{\tau}) \\ & \cdot [y((k-1)T + \hat{\tau}) - y(kT + \hat{\tau})] \} \\ & + \Re \{ y^* ((k-3/2)T + \hat{\tau}) \\ & \cdot [y((k-2)T + \hat{\tau}) - y((k-1)T + \hat{\tau})] \} \} / 2. \end{aligned} \quad (7)$$

3) *NCO*: the NCO consists in a modulo-1 decrementing counter that is recursively updated, every $T_s \cdot P$, according to

$$\eta[(m+P)T_s] = \eta(mT_s) - W(mT_s) \pmod{1}. \quad (8)$$

Its role is to provide the p -th *polyphase MF* with the k -th *basepoint index* $m_k^{(p)}$ and the k -th *fractional index* $\mu_k^{(p)}$, and label the corresponding filtered sample either as *optimum* or *valid*. This can be accomplished according to these steps:

- 1) Define the following vectors of ordered horizontal threshold values: $\mathbf{I}_O \triangleq [0.5, 0, -0.5]$, $\mathbf{I}_V \triangleq [0.75, 0, 25, -0.25, -0.75]$, and $\mathbf{I}_t \triangleq 1 - 0.25 \cdot t$ ($t = 1, \dots, 7$). Odd and even values of t refer to elements in \mathbf{I}_V and \mathbf{I}_O , respectively.
- 2) Determine the number of intersections between the NCO register and the horizontal thresholds in \mathbf{I}_t . Define $\mathbf{I}_{t'}$ as the vector containing the elements of \mathbf{I}_t that intersect the content of the NCO. If an intersection occurs in the p -th interval of \mathcal{I}_p , the NCO generates a control signal $\ell_2^{(p)} = 1$ to notify the p -th *polyphase MF* that a new sample should be computed. Otherwise, if in the p -th interval there are no intersections, the NCO will generate a control signal $\ell_2^{(p)} = 0$ to notify the p -th *polyphase MF* that the corresponding sample should not be computed. Intersections with thresholds in \mathbf{I}_O and \mathbf{I}_V are associated to *optimum* and *valid* samples, respectively. The last element of $\mathbf{I}_{t'}$ belonging to \mathbf{I}_O is associated with the sample computed at instant $kT + \hat{\tau}$.
- 3) For each intersection found at the previous step, compute the timing estimates $\mu_k^{(p)}$. From geometrical considerations, it can be seen, for example by looking at Fig. 4 (a), that $W(mT_s)/P = [\eta(mT_s) - \mathbf{I}_{t'}] / \mu_k^{(p)}$. Thus, the p -th timing estimate can be computed as

$$\mu_k^{(p)} = \frac{\eta(mT_s) - \mathbf{I}_{t'}}{W(mT_s)} \cdot P. \quad (9)$$

In Fig. 4 we report a graphical representation of how to compute the timing estimates $\mu_k^{(p)}$ and how to assign labels to the samples, for two different scenarios.

In Fig. 4 (a), there are 2 intersections between the NCO and the vector \mathbf{I}_O , i.e., 2 *optimum* samples, and 2 intersections between the NCO and the vector \mathbf{I}_V , i.e., 2

valid samples. Therefore, $\mathbf{I}_{t'} = [0.5, 0.25, 0, -0.25]$ and $(\ell_1^{(1)}, \ell_1^{(2)}, \ell_1^{(3)}, \ell_1^{(4)}) = (2, 1, 0, -1)$.

In Fig. 4 (b), instead, there are 2 intersections with vector \mathbf{I}_O and 1 intersection with vector \mathbf{I}_V , i.e., 2 *optimum* samples and 1 *valid* sample. Therefore, $\mathbf{I}_{t'} = [0.5, 0.25, 0]$ and $(\ell_1^{(1)}, \ell_1^{(3)}, \ell_1^{(4)}) = (2, 1, 0)$. We can see that in the interval $[(m+1)T_s, (m+2)T_s]$ there are no intersections between the NCO and the thresholds in \mathbf{I}_t . In this case the NCO will generate a control signal $\ell_2^{(2)} = 0$, so that the second sample will not be computed.

We note that the computation of the timing estimates $\mu_k^{(p)}$ requires a run time division by $W(mT_s)$, as can be seen from (9). The control word $W(mT_s)$ is a time varying quantity, since it depends on the output of the loop filter $v(mT_s)$. Since the division by a time varying quantity can create problems in the hardware implementation, we propose a different method to compute the values of $\mu_k^{(p)}$. Since $v(mT_s)$ is the filtered error signal, it will tend to zero once the loop has converged. Therefore, to the purpose of computing (9), we replace $W(mT_s)$ with the constant value $2/\nu$, i.e. we assume that $W(mT_s) = 2/\nu + v(mT_s) \approx 2/\nu$.

Indicating with $\tilde{\mu}_k^{(p)}$ the new *fractional index*, Eq. (9) becomes

$$\tilde{\mu}_k^{(p)} = \frac{\eta(mT_s) - \mathbf{I}_{t'}}{2/\nu} \cdot P. \quad (10)$$

The relation between $\mu_k^{(p)}$ and $\tilde{\mu}_k^{(p)}$ can be derived as follows:

$$\begin{aligned} \mu_k^{(p)} &= \frac{\eta(mT_s) - \mathbf{I}_{t'}}{W(mT_s)} \cdot P = \frac{\eta(mT_s) - \mathbf{I}_{t'}}{\frac{2}{\nu} + v(mT_s)} \cdot P \\ &= \tilde{\mu}_k^{(p)} \cdot \frac{1}{1 + v(mT_s)\frac{\nu}{2}} \approx \tilde{\mu}_k^{(p)} \cdot \left(1 - \frac{\nu}{2} \cdot v(mT_s)\right), \end{aligned} \quad (11)$$

where the last approximation exploits a first order Taylor expansion.

V. NUMERICAL RESULTS

We evaluated the performance of the proposed symbol timing synchronization algorithm in terms of bit error rate (BER), both for the serial and for the parallel implementation. Performance was compared with that achievable by a receiver operating in the absence of timing offset, i.e., $\tau = 0$, denoted as *Ideal* in the figures that follow.

We considered three ModCods foreseen in the DVB-S2X standard [13] (64APSK with rate 128/180, 128APSK with rate 140/180, and 256APSK with rate 135/180) using low-density parity-check (LDPC) codes.

We set $\Delta f T = 0.01$, as the maximum uncompensated residual offset after coarse frequency synchronization, a sampling clock offset (SCO) of 100 parts-per-million (ppm) and a shaping pulse having root-raised cosine spectrum with roll-off factor $\alpha = 0.2$. We also assumed perfect carrier frequency offset and phase offset compensation before the LDPC decoder.

Fig. 5 (a) shows the BER in a case where only a time offset τ is present, i.e., perfect sampling clock, carrier frequency, and phase recovery are assumed. Both serial and parallel algorithms achieve close to ideal performance, for all the modulation formats. More importantly, we can see that the

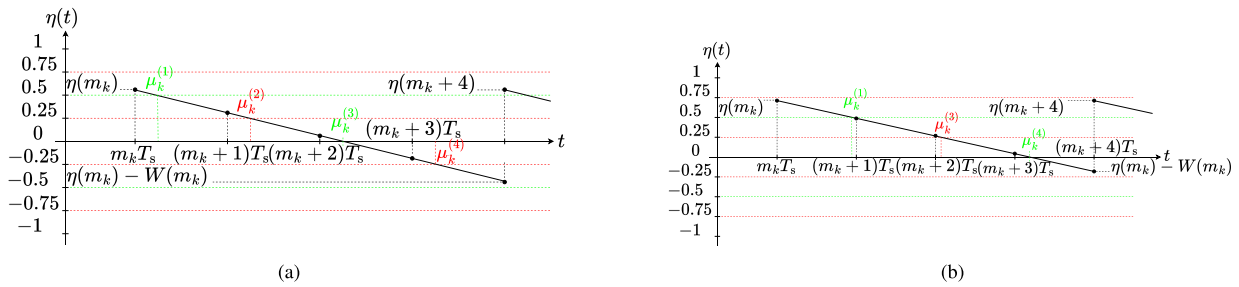


Fig. 4. Timing estimates computation: (a) 4 intersections corresponding to 2 optimum samples (green) and 2 valid samples (red). (b) 3 intersections corresponding to 2 optimum samples (green) and 1 valid samples (red).

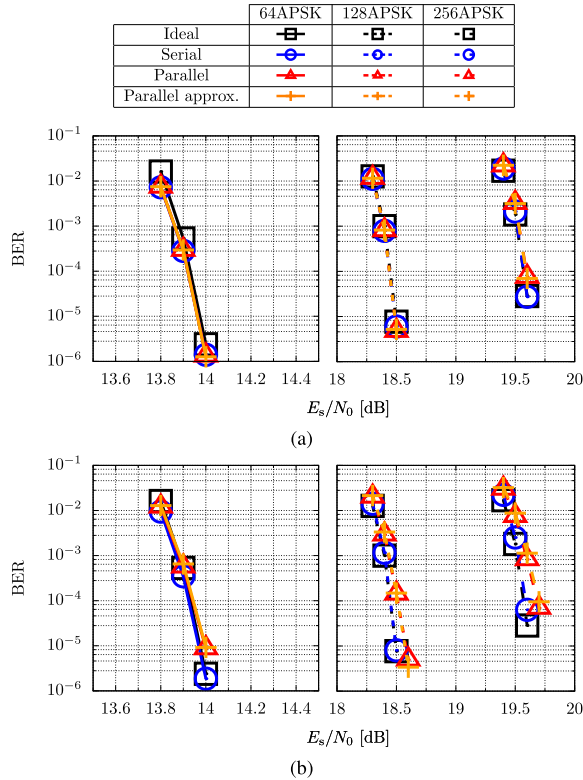


Fig. 5. BER for serial and parallel algorithms, (a) $\Delta f T = 0$, $SCO = 0$ ppm and (b) $\Delta f T = 0.01$, $SCO = 100$ ppm.

parallel algorithm shows no performance losses with respect to the serial architecture.

Fig. 5 (b) reports the BER assuming a residual carrier frequency offset and a non ideal sampling clock frequency. In this case, a slight performance degradation can be noticed for higher-order modulations, for both serial and parallel algorithms. In particular, the parallel algorithm shows a gap of approximately 0.1 dB, at $BER = 10^{-6}$, from the ideal case. The serial and parallel algorithms achieve almost the same performance, with a difference below 0.1 dB.

VI. CONCLUSION

We proposed a parallel algorithm for symbol timing synchronization for high data rates receivers. We evaluated the algorithm with modulations and codes foreseen in DVB-S2X standard. The proposed architecture is derived from the serial

structure, and uses a single NCO to update the timing estimate provided to a bank of parallel interpolators. We evaluated the performance in terms of BER, in the presence of both a residual carrier frequency offset and a sampling clock offset. For all the considered scenarios, the proposed parallel algorithm achieves the same performance as the serial implementation.

ACKNOWLEDGMENT

The view expressed herein can in no way be taken to reflect the official opinion of the European Space Agency.

REFERENCES

- [1] L. Lang, J. Wang, Y. Wang, Z. Zhao, B. Tang, and Q. Xu, "Terahertz high speed parallel signal processing structure," in *Proc. 13th Int. Conf. Commun. Softw. Netw. (ICCSN)*, Jun. 2021, pp. 388–392.
- [2] C. Lin, J. Zhang, and B. Shao, "A high speed parallel timing recovery algorithm and its FPGA implementation," in *Proc. 2nd Int. Symp. Intell. Inf. Process. Trusted Comput.*, Oct. 2011, pp. 63–66.
- [3] X. Zhou and X. Chen, "Parallel implementation of all-digital timing recovery for high-speed and real-time optical coherent receivers," *Opt. Exp.*, vol. 19, no. 10, p. 9282, May 2011.
- [4] H. Li, Z.-G. Wang, and H.-J. Wang, "A high speed parallel timing synchronization algorithm for 16QAM," in *Proc. 13th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process. (ICCWAMTIP)*, Dec. 2016, pp. 403–407.
- [5] X. Hao, C. Lin, and Q. Wu, "A parallel timing synchronization structure in real-time high transmission capacity wireless communication systems," *Electronics*, vol. 9, no. 4, p. 652, Apr. 2020.
- [6] J. Hu, L. Zhu, and J. Wang, "The implementation of high speed parallel timing synchronization algorithm based on FPGA," in *Proc. 10th Int. Conf. Commun. Softw. Netw. (ICCSN)*, Jul. 2018, pp. 484–487.
- [7] X. Hao, Q. Wu, Z. Wang, and C. Lin, "Parallel timing synchronization algorithm and its implementation in high speed wireless communication systems," in *Proc. Int. Conf. Electron., Inf., Commun. (ICEIC)*, Jan. 2019, pp. 1–6.
- [8] U. Mengali and A. D'Andrea, *Synchronization Techniques for Digital Receivers*. New York, NY, USA: Springer, 1997.
- [9] F. J. Harris and M. Rice, "Multirate digital filters for symbol timing synchronization in software defined radios," *IEEE Sel. Areas Commun.*, vol. 19, no. 12, pp. 2346–2357, Dec. 2001.
- [10] F. Gardner, "A BPSK/QPSK timing-error detector for sampled receivers," *IEEE Trans. Commun.*, vol. COM-34, no. 5, pp. 423–429, May 1986.
- [11] F. M. Gardner, "Phaselock techniques," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-14, no. 1, pp. 170–171, Jan. 1984. [Online]. Available: <https://ieeexplore.ieee.org/document/6313286>
- [12] M. Rice, *Digital Communications: A Discrete-Time Approach*. London, U.K.: Pearson Education, 2009.
- [13] *Second Generation Framing Structure, Channel Coding and Modulation Systems for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications, Part II: S2-Extensions (DVB-S2X)*, Standard ETSI EN 302 307-2, Digital Video Broadcasting (DVB), 2014. [Online]. Available: <http://www.etsi.org>