

# Modified parallel code-phase search for acquisition in presence of sign transition

Jérôme Leclère, Cyril Botteron, Pierre-André Farine  
 Electronics and Signal processing Laboratory (ESPLAB)  
 École Polytechnique Fédérale de Lausanne (EPFL), Neuchâtel, Switzerland  
 Email: firstname.lastname@epfl.ch

**Abstract**—One of the method to have a fast acquisition of GNSS signals is the parallel code-phase search, which uses the fast Fourier transform (FFT) to perform the correlation. A problem with this method is the potential sign transition that can happen between two code periods due to data or secondary code and lead to a loss of sensitivity or to the non-detection of the signal. A known straightforward solution consists in using two code periods instead of one for the correlation. However, in addition to increasing the complexity, this solution is not efficient since half of the points calculated are discarded. This led us to look for a more efficient algorithm. The algorithm proposed in this article transforms the initial correlation into two smaller correlations. When the radix-2 FFT is used, the proposed algorithm is more efficient for half of the possible sampling frequencies. It is shown for example that the theoretical number of operations can be reduced by about 21 %, and that the memory resources for an FPGA implementation can be almost halved.

## I. INTRODUCTION

After the antenna and the front-end, the first step of a GNSS receiver is the acquisition [1]. With the evolution of the technology, the serial search has been replaced by parallel architectures, with a massive duplication of correlators or using FFTs [2]. A drawback of using the FFT is that a circular correlation is performed instead of a linear correlation, which can lead to a reduced signal-to-noise ratio due to potential sign transitions [3].

A well-known straightforward solution consists in doubling the length of the correlation and using zero-padding [4], [5]. In this case, the first half of the correlation always contains a correlation peak that is maximum, while the second half contains a peak that may be affected by a sign transition. Consequently, only the first half of the correlation is used.

This straightforward solution however increases the complexity, and is not computationally efficient since half of the points calculated are not used. This led us to look for a more efficient algorithm. Instead of computing the initial correlation, the algorithm proposed in this paper computes two smaller correlations. This algorithm may be more efficient if there is a constraint on the correlation length. For example, considering the use of radix-2 FFTs, for half of the possible sampling frequencies, the FFTs length is divided by two, leading to a better efficiency.

It is important to note that the proposed algorithm has no impact on the sensitivity since it is not an approximation but simply a different way to perform an operation, contrary

to algorithms performing a down-sampling in the frequency domain [6] or using a folding technique [7] for example.

Section II presents briefly the acquisition principle, and describes the use of the FFT, the problem due to the potential transitions, and the straightforward solution. Section III details the proposed algorithm. Section IV compares the straightforward and proposed algorithms for two particular cases before generalizing. Section V concludes on the applicability of the proposed algorithm.

## II. ACQUISITION OF GNSS SIGNALS

The acquisition consists in detecting the satellites in view and estimating their corresponding Doppler frequency and code phase. To do so, the incoming signal is multiplied by a complex exponential, then multiplied by a locally generated pseudo-random noise (PRN) code, and then accumulated to raise the signal out of the noise. To detect the signal, the frequency of the complex exponential should be as close as possible to the carrier frequency of the incoming signal, and the phase of the local PRN code should be as close as possible to the code phase of the incoming signal [8]. In a serial search, the possibilities are tested one after each other until the signal is detected, which implies a very long processing time due to the large number of possibilities. The FFT is thus often used to make the acquisition faster, by parallelizing the search in the code domain (method called parallel code-phase search (PCS)) or in the frequency domain (method called parallel frequency search (PFS)) [9], [10]. In this paper, we focus on the PCS.

### A. Correlation

The multiplication of the incoming signal by the local PRN code and the accumulation for different phases is equivalent to a correlation, as shown by (1), where  $\mathbf{h}$  and  $\mathbf{x}$  are the signals to correlate (they correspond to the local PRN code and to the incoming signal after the carrier removal, respectively), and  $N$  is the number of samples in one code period.

$$y[n] = \sum_{k=0}^{N-1} h[k]x[n+k] \quad \text{with } n = 0, \dots, N-1 \quad (1)$$

Calculating the correlation with (1) corresponds to fixing  $\mathbf{x}$  and shifting  $\mathbf{h}$  to the right, as illustrated in Fig. 1, where the position of  $\mathbf{h}$  relative to  $\mathbf{x}$  is shown for two different values of  $n$ .

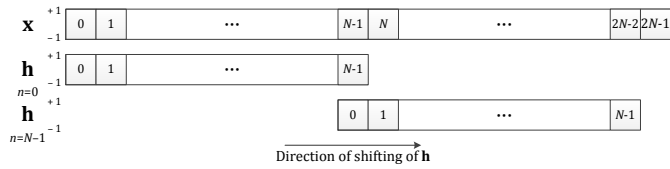


Fig. 1. Illustration of the correlation (Eq. (1)). The numbers in the boxes indicate the samples.

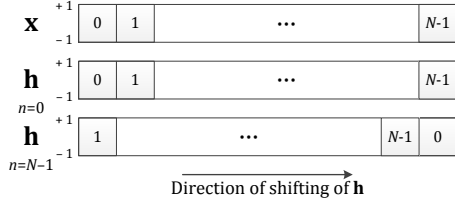


Fig. 2. Illustration of the circular correlation (Eq. (2)). The numbers in the boxes indicate the samples.

### B. Parallel code-phase search

It is well known that a circular correlation can be computed efficiently using FFTs [11]. Since the incoming code is periodic, doing a circular correlation instead of a correlation is not a problem. The FFT can thus be used for the acquisition [12]. The circular correlation between two signals,  $\mathbf{h}$  and  $\mathbf{x}$ , is given by (2), where the overbar denotes the complex conjugate, mod denotes the modulo operation, and IFFT means inverse FFT. To conjugate  $\mathbf{h}$  has no effect since the local code is real, but the conjugate on the FFT of  $\mathbf{h}$  must be kept, else the operation would become a convolution.

$$y[n] = \sum_{k=0}^{N-1} \overline{h[k]} x[(n+k) \bmod N] \quad \text{with } n = 0, \dots, N-1$$

$$\mathbf{y} = \text{IFFT}\left(\overline{\text{FFT}(\mathbf{h})} \text{FFT}(\mathbf{x})\right) \quad (2)$$

Calculating the circular correlation with (2) corresponds to fixing  $\mathbf{x}$  and circularly shifting  $\mathbf{h}$  to the right (or equivalently fixing  $\mathbf{h}$  and circularly shifting  $\mathbf{x}$  to the left), as illustrated in Fig. 2.

### C. Problem of potential sign transitions

With the PCS, the length of the incoming code used for the circular correlation corresponds to one period. However, the first sample of the incoming code can be any chip. This means that the incoming code is in fact composed of two portions of two different periods. Because of the data or the secondary code, a sign transition can thus be present and result in a reduction of the correlation peak, as shown in Fig. 3. Without data, at the correct alignment, the multiplication of the two codes results in a signal having only ones, and accumulating this signal gives the value 1023. However, with data, after the multiplication of the two codes, the first 623 points are equal to 1, and the last 400 points are equal to  $-1$ , leading to a value of 223 after accumulation.

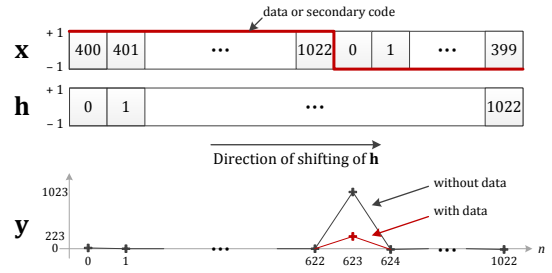


Fig. 3. Illustration of the problem due to data bit transition for a code of 1023 chips (e.g. code of the GPS L1 C/A signal). The numbers in the boxes indicate the chips.

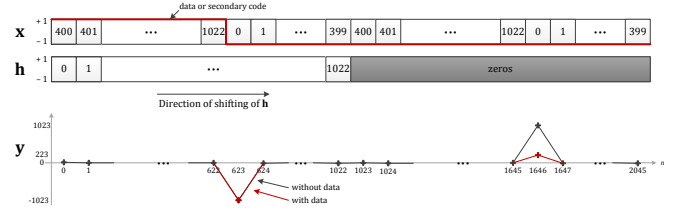


Fig. 4. Illustration of the straightforward solution to the data bit transition problem for a code of 1023 chips (Eq. (3)). The numbers in the boxes indicate the chips.

### D. Straightforward solution

A well-known straightforward solution consists in performing a circular correlation using two consecutive periods of the incoming code and one period of code padded with zeros for the local replica [4], [5]. In this way, there is always a complete period of the incoming code, and thus a maximum correlation peak, as shown in Fig. 4.

The corresponding equation is given by (3), with  $N$  still the number of samples in one code period, and  $h[k] = 0$  for  $N \leq k \leq 2N - 1$ .

$$y[n] = \sum_{k=0}^{2N-1} \overline{h[k]} x[(n+k) \bmod 2N] \quad \text{with } n = 0, \dots, 2N-1$$

$$\mathbf{y} = \text{IFFT}\left(\overline{\text{FFT}(\mathbf{h})} \text{FFT}(\mathbf{x})\right) \quad (3)$$

Since the length of the incoming code corresponds to two periods, there are two peaks in the circular correlation. However, the peak appearing in the first half of the circular correlation is always maximum (the sign of the peak is not important, only its magnitude is), while the peak appearing in the second half of the circular correlation can be reduced due to a sign transition. Consequently, only the first half of the circular correlation is used and the second half is discarded, i.e. only  $y[n]$  for  $n = 0, \dots, N - 1$  is desired. It can be noted that the last sample of  $\mathbf{x}$  is not necessary anymore, and thus the correlation can be computed on  $2N - 1$  points if wanted. Of course, this solution is not computationally efficient. The next section describes the proposed algorithm to obtain the first half of the circular correlation while discarding fewer points.

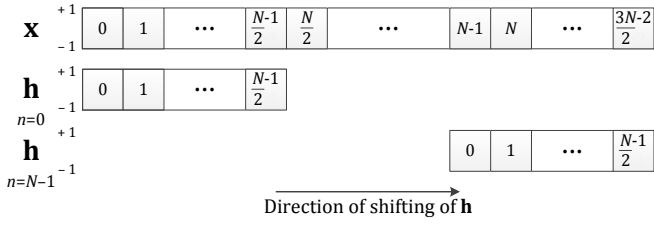


Fig. 5. Illustration of the first sum in (6). The numbers in the boxes indicate the samples.

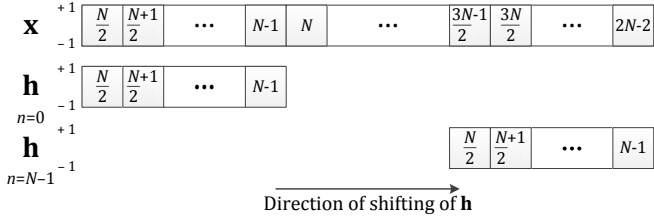


Fig. 6. Illustration of the second sum in (6). The numbers in the boxes indicate the samples.

### III. PROPOSED ALGORITHM

#### A. Derivation

Firstly, starting from (3), we use the fact that  $h[k] = 0$  for  $N \leq k \leq 2N - 1$  to obtain (4).

$$y[n] = \sum_{k=0}^{N-1} \overline{h[k]} x[(n+k) \bmod 2N] \quad \text{with } n = 0, \dots, 2N-1 \quad (4)$$

Secondly, we use the fact that we are not interested in computing the second half of  $\mathbf{y}$  to limit  $n$  to  $N - 1$ . This implies that  $n+k \leq 2N - 2$ , which allows the removal of the modulo operation to obtain (5), which is equivalent to (1).

$$\begin{aligned} y[n] &= \sum_{k=0}^{N-1} \overline{h[k]} x[(n+k) \bmod 2N] \quad \text{with } n = 0, \dots, N-1 \\ &= \sum_{k=0}^{N-1} \overline{h[k]} x[n+k] \quad \text{with } n = 0, \dots, N-1 \end{aligned} \quad (5)$$

Thirdly, we separate the sum in (5) in two sums to obtain (6). For the first sum,  $0 \leq n+k \leq 3N/2 - 2$ , therefore the last  $N/2$  samples of  $\mathbf{x}$  are not used, as illustrated in Fig. 5. For the second sum,  $N/2 \leq n+k \leq 2N - 2$ , therefore the first  $N/2$  samples of  $\mathbf{x}$  are not used, as illustrated in Fig. 6.

$$\begin{aligned} y[n] &= \sum_{k=0}^{N/2-1} \overline{h[k]} x[n+k] + \sum_{k=N/2}^{N-1} \overline{h[k]} x[n+k] \\ &\quad \text{with } n = 0, \dots, N-1 \end{aligned} \quad (6)$$

Fourthly, we replace the sums in (6) by circular correlations by computing  $N/2 - 1$  extra samples, i.e.  $0 \leq n \leq 3N/2 - 2$ , and including a modulo operation to obtain (7), which is illustrated in Figs. 7 and 8.

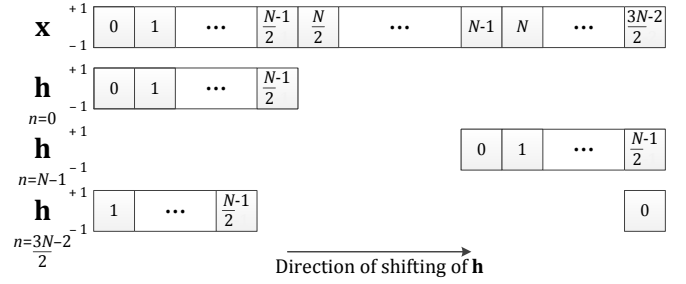


Fig. 7. Illustration of the first sum in (7). The numbers in the boxes indicate the samples.

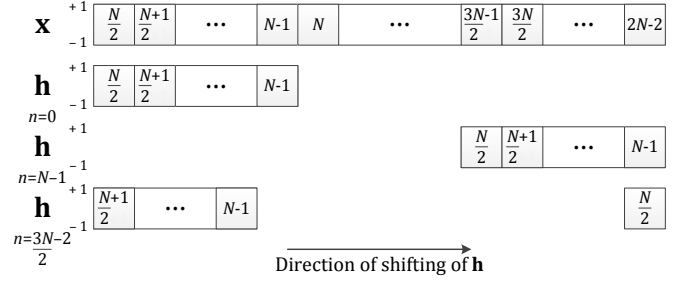


Fig. 8. Illustration of the second sum in (7). The numbers in the boxes indicate the samples.

$$\begin{aligned} y_M[n] &= \sum_{k=0}^{N/2-1} \overline{h[k]} x[(n+k) \bmod (3N/2-1)] \\ &\quad + \sum_{k=0}^{N/2-1} \overline{h\left[\frac{N}{2}+k\right]} x\left[\frac{N}{2}+(n+k) \bmod (3N/2-1)\right] \\ &\quad \text{with } n = 0, \dots, 3N/2-2 \end{aligned} \quad (7)$$

Coming back with the FFT notation, the operation performed in (7) can be computed using FFTs as shown in (8).

$$\begin{aligned} \mathbf{y}_M &= \text{IFFT}\left(\overline{\text{FFT}(\mathbf{h}_0)} \text{FFT}(\mathbf{x}_0)\right) \\ &\quad + \text{IFFT}\left(\overline{\text{FFT}(\mathbf{h}_1)} \text{FFT}(\mathbf{x}_1)\right) \\ &= \text{IFFT}\left(\overline{\text{FFT}(\mathbf{h}_0)} \text{FFT}(\mathbf{x}_0) + \overline{\text{FFT}(\mathbf{h}_1)} \text{FFT}(\mathbf{x}_1)\right) \end{aligned} \quad (8)$$

with

$$\begin{aligned} \mathbf{h}_0 &= \left[ h[0] \quad h[1] \quad \dots \quad h\left[\frac{N}{2}-1\right] \quad \overbrace{0 \quad \dots \quad 0}^{N-1} \right] \\ \mathbf{h}_1 &= \left[ h\left[\frac{N}{2}\right] \quad h\left[\frac{N}{2}+1\right] \quad \dots \quad h[N-1] \quad 0 \quad \dots \quad 0 \right] \\ \mathbf{x}_0 &= \left[ x[0] \quad x[1] \quad \dots \quad x\left[\frac{3N}{2}-2\right] \right] \\ \mathbf{x}_1 &= \left[ x\left[\frac{N}{2}\right] \quad x\left[\frac{N}{2}+1\right] \quad \dots \quad x[2N-2] \right] \end{aligned}$$

The resulting signal  $\mathbf{y}_M$  (M for modified) is a vector of  $3N/2 - 1$  samples. Its first  $N$  samples are equal to the first  $N$  samples of  $\mathbf{y}$ , i.e. the proposed algorithm computes correctly the first half of  $\mathbf{y}$ , where there is the maximum correlation

peak. The last  $N/2 - 1$  samples of  $\mathbf{y}_M$  are different than those of  $\mathbf{y}$ , without consequences since these samples are discarded.

The proposed algorithm computes thus five FFTs (FFT and IFFT are equivalent from computational point of view) of  $3N/2 - 1$  points and discards  $N/2 - 1$  points, while the straightforward algorithm computes three FFTs of  $2N$  points and discards  $N$  points (or  $2N - 1$  and  $N - 1$  if the last sample of  $\mathbf{x}$  is not used).

Sometimes, it is assumed that the FFT for the local code are pre-generated in order to save operations during the acquisition. In this case, the proposed algorithm computes 3 FFTs while the straightforward algorithm computes 2 FFTs.

### B. Use with radix-2 FFT

Since the proposed algorithm performs FFTs of  $3N/2 - 1$  points, if the use of radix-2 FFTs is desired, there is the constraint given by (9), with  $l$  a positive integer [13], [14]. This equation has integer solutions only if  $l$  is odd, and the result for a range of suitable values is provided in Table I.

$$\frac{3N}{2} - 1 = 2^l \Leftrightarrow N = \frac{2}{3}(2^l + 1) \quad (9)$$

However, it is possible to add zeros at the end of the signals to increase the length without impacting the first  $N$  samples of  $\mathbf{y}_M$ , as illustrated in Fig. 9 where one zero is added.

By adding one zero, the proposed algorithm performs FFTs of  $3N/2$  points, and the constraint to use radix-2 FFTs is given by (10), which has no integer solution.

$$\frac{3N}{2} = 2^l \Leftrightarrow N = \frac{2}{3}2^l \quad (10)$$

By adding two zeros, the proposed algorithm performs FFTs of  $3N/2 + 1$  points, and the constraint to use radix-2 FFTs is given by (11), which has integer solutions only if  $l$  is even. The result for a range of suitable values is provided in Table I.

$$\frac{3N}{2} + 1 = 2^l \Leftrightarrow N = \frac{2}{3}(2^l - 1) \quad (11)$$

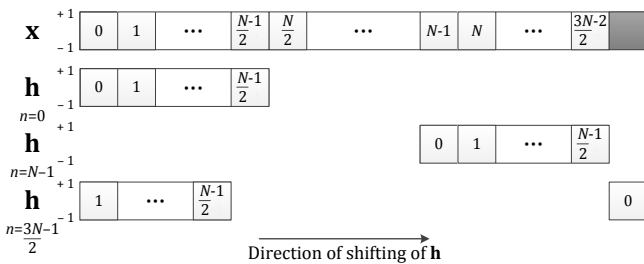


Fig. 9. Illustration of the first sum in (7) with the padding of one zero. The numbers in the boxes indicate the samples.

$l$	11	12	13	14	15	16
$L = 2^l$	2048	4096	8192	16384	32768	65536
$2N$	2732	5460	10924	21844	43692	87380

TABLE I

POSSIBLE CORRELATION LENGTH ( $2N$ ) AND FFT LENGTH ( $L$ ) FOR THE PROPOSED ALGORITHM CONSIDERING THE RADIX-2 FFT.

## IV. APPLICATION

In this section, the straightforward and proposed algorithms are compared considering a PRN code of 1 ms. The comparison is done in terms of number of operations, FPGA resources and processing time on personal computers. Two particular cases are discussed before presenting the general case.

### A. Sampling frequency of 5 MHz

For the first case, a sampling frequency of 5 MHz is considered. There are thus  $N = 5000$  samples in one code period. To tackle the sign transition problem, two periods are used, i.e.  $2N = 10000$  samples.

For the straightforward algorithm, in order to use radix-2 FFTs, the signals must be padded with zeros until the closest power of two, i.e. 16384. The corresponding implementation is given in Fig. 10, where  $\mathbf{x}$  contains 10000 signal samples plus 6384 zeros, and  $\mathbf{h}$  contains 5000 signal samples plus 11384 zeros. Note that since the local code is a real signal, the FFT followed by the conjugate operation has been replaced by an IFFT for the sake of simplicity [15].

For the proposed algorithm,  $\mathbf{x}_0$  and  $\mathbf{x}_1$  contain  $3N/2 - 1 = 7499$  signal samples, and must be padded with 693 zeros to reach the closest power of two, 8192. The proposed algorithm will thus compute the first 8192 samples of a circular correlation on 10924 points (cf Table I). The corresponding implementation is given in Fig. 11 where  $\mathbf{x}_0$  contains the first 7499 samples of  $\mathbf{x}$  plus 693 zeros,  $\mathbf{x}_1$  contains the last 7499 samples of  $\mathbf{x}$  plus 693 zeros,  $\mathbf{h}_0$  contains the first  $N/2 = 2500$  samples of  $\mathbf{h}$  plus 5692 zeros, and  $\mathbf{h}_1$  contains the last 2500 samples of  $\mathbf{h}$  plus 5692 zeros.

The complexity of both algorithms can be evaluated by the number of operations they perform. Considering that an FFT of  $L$  points requires  $L/2 \log_2(L)$  multiplications, the straightforward algorithm performs  $3L/2 \log_2(L) + L$  multiplications, i.e. 360448 with  $L = 16384$ . The proposed algorithm performs  $5L/2 \log_2(L) + 2L$  multiplications, i.e. 282624 with  $L = 8192$ . The theoretical number of multiplications is thus reduced by 21.6% (the reduction is similar for the number of additions). If we consider that the FFTs for the local code are pre-generated, the theoretical number of multiplications would be reduced by 28.3%, which is even better.

The complexity can also be evaluated by the resources required with a hardware implementation. Considering an Altera Stratix III FPGA, a basic comparison of the algorithms can be done using the estimation provided by the Altera MegaWizard. Table II provides the resources in terms of Adaptive Look-up table (ALUT), memory of 9 Kibit (M9K) and digital signal

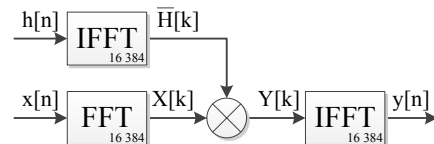


Fig. 10. Implementation of the straightforward algorithm for a sampling frequency of 5 MHz.

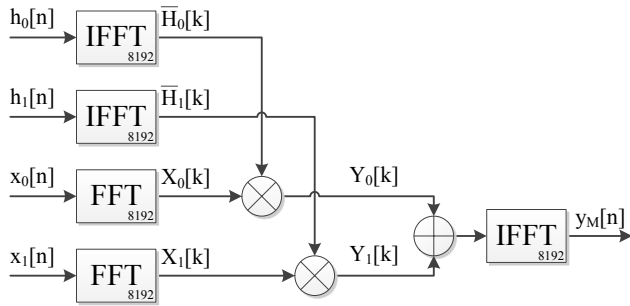


Fig. 11. Implementation of the proposed algorithm for a sampling frequency of 5 MHz.

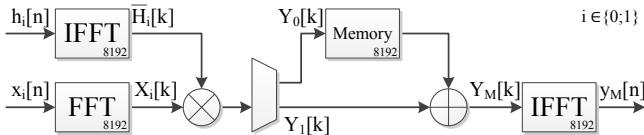


Fig. 12. Implementation of the proposed algorithm using multiplexing for a sampling frequency of 5 MHz.

processing (DSP) elements for both algorithms, considering the streaming implementation of the FFT with a resolution of 18 bits [16]. It can be seen that the proposed algorithm uses more resources, except for the memory, however the processing time is divided by two since the FFT length is divided by two [15]. Since the resources are increased by a factor less than two, there is thus a gain in efficiency.

If an increase of the resources is not desired, the proposed algorithm can be used with multiplexing as depicted in Fig. 12. The multiplexing requires an additional memory, which is compensated by the removal of two FFT blocks, as shown in Table II. Therefore, with the implementation of Fig. 12, the processing time is about the same as with the implementation of Fig. 10 (in fact it is slightly reduced due to the reduced latency of the FFTs [15]), and the resources are globally reduced (by almost of a factor two for the memory).

The complexity can also be evaluated by the processing time required with a software implementation. The average processing time of the algorithms has been measured on four different personal computers. With the proposed algorithm, the processing time was reduced by 24, 18, 12 and 11 %, respectively, which gives an average reduction of 16.2 %.

Algorithm	Logic (ALUT)	Memory (M9K)	Multipliers (DSP element)
Straightforward	22 431	912	76
Proposed	35 286	760	128
Proposed with multiplexing	21 150	488	76

TABLE II

FPGA RESOURCES FOR THE STRAIGHTFORWARD ( $L = 16\,384$ ) AND PROPOSED ( $L = 8192$ ) ALGORITHMS.

### B. Sampling frequency of 4 MHz

For the second case, a sampling frequency of 4 MHz is considered. There are thus  $N = 4000$  samples in one code period. To tackle the sign transition problem, two periods are used, i.e.  $2N = 8000$  samples.

For the straightforward algorithm, in order to use radix-2 FFTs, the signals have to be padded with zeros until the closest power of two, i.e. 8192.

For the proposed algorithm,  $x_0$  and  $x_1$  contain  $3N/2 - 1 = 5999$  signal samples, and must be padded with 2193 zeros to reach the closest power of two, 8192. This means that the FFT length is the same as for the straightforward algorithm, while the proposed algorithm performs 5 FFTs instead of 3 for the straightforward algorithm. Consequently, in this case, the proposed algorithm is not more efficient.

### C. General case

The two particular cases discussed have resulted in different conclusions. In the first case, the proposed algorithm was more efficient than the straightforward algorithm because the FFT length was halved, while in the second case the proposed algorithm was not more efficient since the FFT length was unchanged and more FFTs were computed. In fact, the efficiency of the proposed algorithm depends on the initial correlation length, i.e. on the sampling frequency, as shown in Fig. 13.

If the initial number of points is slightly below or equal to a power of two, the straightforward algorithm is more efficient because the zero-padding needed to reach the desired length is moderate (e.g. for the second case discussed, the length has been increased from 8000 to 8192, i.e. only 2.4 % of increase).

If the initial number of points is slightly above a power of two, the straightforward algorithm requires important zero-padding (e.g. for the first case discussed the length has been increased from 10 000 to 16 384, i.e. 63.84 % of increase), while the proposed algorithm requires moderate zero-padding (e.g. for the first case discussed, the length has been increased from 10 000 to 10 924, i.e. 9.24 % of increase), which explains the better efficiency of the proposed algorithm in this case.

The best choice between the straightforward and proposed algorithms according to the sampling frequency can be found using Fig. 13. It can be seen that the proposed algorithm is more efficient than the straightforward algorithm for half of the sampling frequencies. For the GPS L1 C/A signal, which has a PRN code of 1 ms, the minimum sampling frequency is 2.046 MHz. If the sampling frequency is close to this minimum, more specifically between 2.046 and 2.048 MHz, the straightforward algorithm is more efficient. However, such low sampling frequency result in poor resolution and a higher frequency is often used. For the GPS L5, Galileo E5a and E5b signals, which have a PRN code of 1 ms, the minimum sampling frequency is 20.46 MHz. If the sampling frequency is close to this minimum, more specifically between 20.46 and 21.846 MHz, the proposed algorithm is more efficient (as also demonstrated in [17]).

It exists methods that perform a compression or averaging before the correlation to reduce the number of points. For example, a method performs an average in order to obtain one sample per chip [18]. The proposed algorithm can be used in addition to such method. For example, with the GPS L1 C/A signal, the number of points would be 2046 after averaging (two periods of 1023 points); in this case the straightforward algorithm is more efficient (the equivalent sampling frequency is 1.023 MHz). With the GPS L5, Galileo E5a and E5b signals, the number of points would be 20 460 after averaging (two periods of 10 230 points); in this case the proposed algorithm is more efficient (the equivalent sampling frequency is 10.23 MHz).

## V. CONCLUSION

This paper focuses on the problem of the acquisition of GNSS signals with the parallel code-phase search approach when sign transitions due to data or secondary code can happen.

A straightforward solution available in the literature double the length of the correlation and discards half of the points calculated, which is inefficient. With this in mind, we try to find a more efficient algorithm. The algorithm we propose transforms the initial correlation into two smaller correlations, implying more FFTs, but of smaller length.

Without constraints on the length of the FFT, the proposed algorithm does not present advantage. However, when using the common and widespread radix-2 FFT, which requires lengths that are a power of two, the proposed algorithm can be more efficient. Indeed, for some sampling frequencies, the zero-padding to reach the required length is less important for the proposed algorithm, which enables to halve the length of the FFTs and leads to a better efficiency. The proposed algorithm is more efficient for half of the possible sampling frequencies. For the other half, the length of the FFT is not reduced, and the proposed algorithm is thus not more efficient.

The proposed algorithm can be applied to any GNSS signal, and is particularly well-suited for the GPS L5, Galileo E5a and E5b signals. It can potentially be applied to other domains, as soon as a system performs a circular correlation where half of an input is zero, and half of the output is discarded.

## REFERENCES

- [1] L. Weill, "Differences between signal acquisition and tracking," *Inside GNSS*, vol. 6, no. 1, pp. 22–27, 2011.
- [2] L. Kurz, T. Kappen, T. Coenen, and T. Noll, "Comparison of massive-parallel and FFT based acquisition architectures for GNSS-receivers," in *ION GNSS 2010*, September 2010, pp. 2874–2883.
- [3] L. Lo Presti, X. Zhu, M. Fantino, and P. Mulassano, "GNSS signal acquisition in the presence of sign transition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 4, pp. 557–570, August 2009.
- [4] D. Borio, M. Fantino, and L. Lo Presti, "The impact of the Galileo signal in space in the acquisition system," in *Satellite Communications and Navigation Systems*, ser. Signals and Communication Technology, E. Re and M. Ruggieri, Eds. Springer US, 2008, pp. 151–167.
- [5] C. Yang, C. Hegarty, and M. Tran, "Acquisition of the GPS L5 signal using coherent combining of L5 and Q5," in *ION GNSS 2004*, September 2004, pp. 2184–2195.
- [6] F. Xu and G. Yang, "A new GNSS acquisition method with signal down sampling in frequency domain," in *IEEE VTC*, April 2009, pp. 1–5.
- [7] P.-H. Lee, D.-H. Liu, W.-L. Mao, H. Tsao, F. Chang, H. Yeh, and K. Chen, "A novel low-complexity acquisition method for next generation GNSS signals," in *ION GNSS 2008*, September 2008, pp. 1507–1513.
- [8] F. van Diggelen, *A-GPS: Assisted GPS, GNSS, and SBAS*, ser. GNSS Technology and Applications Series. Artech House, 2009.
- [9] K. Borre, D. Akos, N. Bertelsen, P. Rinder, and S. Jensen, *A Software-Defined GPS and Galileo Receiver*, ser. Applied and Numerical Harmonic Analysis. Birkhäuser, 2007.
- [10] J. Leclère, C. Botteron, and P.-A. Farine, "Comparison framework of FPGA-based GNSS signals acquisition architectures," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 3, July 2013.
- [11] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications, 4/E*. Prentice Hall, 2006.
- [12] D. van Nee and A. Coenen, "New fast GPS code-acquisition technique using FFT," *Electronics Letters*, vol. 27, no. 2, pp. 158–160, 1991.
- [13] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comp.*, vol. 19, pp. 297–301, 1965.
- [14] P. Duhamel and M. Vetterli, "Fast Fourier transforms: A tutorial review and a state of the art," *Signal Processing*, vol. 19, no. 4, pp. 259–299, 1990.
- [15] J. Leclère, C. Botteron, and P.-A. Farine, "Improving the performance of the FFT-based parallel code-phase search acquisition of GNSS signals by decomposition of the circular correlation," in *ION GNSS 2012*, September 2012, pp. 1406–1416.
- [16] Altera, *FFT MegaCore Function User Guide*, November 2012.
- [17] J. Leclère, C. Botteron, and P.-A. Farine, "Acquisition of GNSS signals having a secondary code using a modified parallel code-phase search architecture," *Signal Processing*, submitted for publication.
- [18] J. Starzyk and Z. Zhu, "Averaging correlation for C/A code acquisition and tracking in frequency domain," in *IEEE MWSCAS 2001*, vol. 2, August 2001, pp. 905–908.

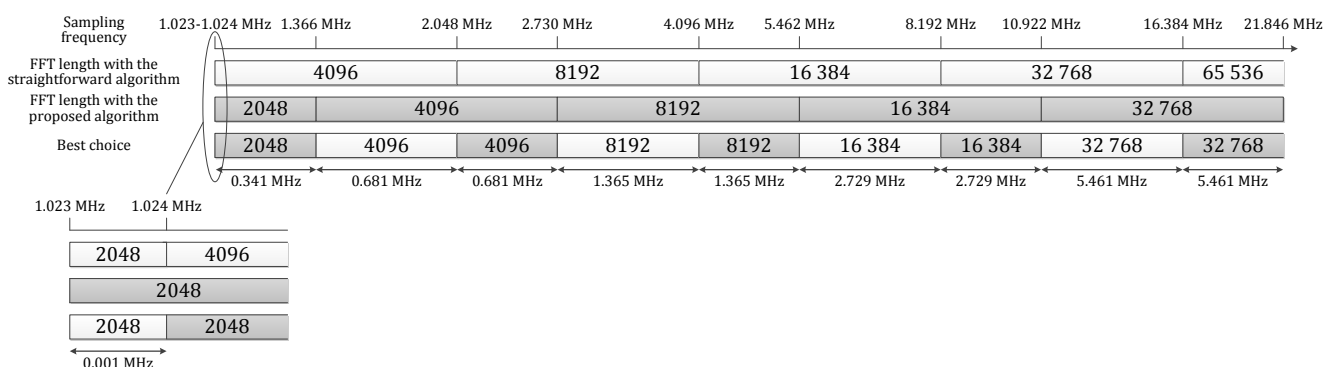


Fig. 13. FFT length for the straightforward and proposed algorithms for a 1-ms code according to the sampling frequency. The axis is logarithmic. For a 4-ms code (as Galileo E1), multiply the actual sampling frequency by 4.