Acquisition of modern GNSS signals using a modified parallel code-phase search architecture

Jérôme Leclère*, Cyril Botteron, Pierre-André Farine

Electronics and Signal Processing Laboratory (ESPLAB), École Polytechnique Fédérale de Lausanne (EPFL), Neuchâtel, Switzerland.

Abstract

The acquisition of global navigation satellite system signals can be performed using a fast Fourier transform (FFT). The FFT-based acquisition performs a circular correlation, and is thus sensitive to potential transitions between consecutive periods of the code. Such transitions are not occurring often for the GPS L1 C/A signal because of the low data rate, but very likely for the new GNSS signals having a secondary code. The straightforward solution consists in using two periods of the incoming primary code and using zero-padding for the local code to perform the correlation. However, this solution increases the complexity, and is moreover not efficient since half of the points calculated are discarded. This has led us to research for a more efficient algorithm, which discards less points by calculating several sub-correlations. It is applied to the GPS L5, Galileo E5a, E5b and E1 signals. Considering the radix-2 FFT, the proposed algorithm is more efficient for the L5, E5a and E5b signals, and possibly for the E1 signal. The theoretical number of operations can be reduced by 21 %, the processing time measured on a

*Corresponding author

Email address: jerome.leclere@epfl.ch (Jérôme Leclère)

software implementation is reduced by 39 %, and the memory resources are almost halved for an FPGA implementation.
Keywords: Acquisition, Correlation, FFT, FPGA, GNSS, Secondary code

1. Introduction

The satellites of the global navigation satellite systems (GNSSs), such as the American GPS or the European Galileo, send continuously signals in direction of the Earth. Those signals have mainly three components [1] : 1) Navigation data, which is a binary-coded message of +1 and -1 transmitted at low rate to provide the information necessary for the navigation, such as time and ephemeris; 2) A ranging code, which is a long known sequence of +1 and -1 specific to each satellite and transmitted at high rate. This code, also called pseudo-random noise (PRN) code, allows precise ranging and let the satellites to broadcast signals at the same frequencies (principle of the code division multiple access, CDMA). The values of the PRN codes are called chips instead of bits, to emphasize that they do not carry information, unlike bits of data; 3) A carrier, which is a sinusoidal signal whose frequency is in the L band.

After the antenna, and after the front-end, which downconverts the radio frequency signal to baseband by removing the carrier and performs the digitization, the first stage of a GNSS receiver is the acquisition [1]. Its purpose is threefold : 1) Detect the satellites in view; 2) Obtain a rough estimation of the received carrier frequency, because there is an uncertainty due to the Doppler effect and the receiver oscillator inaccuracy; and 3) Obtain a rough estimation of the phase of the PRN code transmitted, which is unknown

without time synchronization and a priori knowledge of the geometry.

The recently introduced GPS and Galileo signals bring new features compared to the initial civilian GPS L1 C/A signal, such as a higher power, longer codes for a better cross-correlation between satellites signals, pilot channels that do not carry data to facilitate long integrations and improve the sensitivity threshold, and secondary codes that are short PRN codes to make the data synchronization easier [2].

The secondary code is, as indicated by its name, a second code, which multiplies the primary code to form a longer code (called tiered code). The rate of the secondary code is lower than that of the primary code, since the length of one chip of the secondary code is equal to one period of the primary code, as shown in Fig. 1. The presence of a secondary code brings advantages and additional performance, but also makes the acquisition more difficult [2]. Exploiting the secondary code adds a third dimension to the acquisition search (besides the Doppler frequency and primary code dimensions), and implies the use of long coherent integration times, which impacts also the search in the Doppler frequency dimension. The secondary code is typically used to acquire very weak signals, such as in indoor or urban environment [3]. However, it is still possible to perform the acquisition using only the primary code, with the possibility to synchronize with the secondary code afterwards if the sensitivity is not the priority. This article focuses on this case.

Even in the cases where the secondary code is not exploited, the potential transitions between consecutive periods of the primary code prevent the direct use of the Parallel Code-phase Search (PCS) architecture [1], which uses

FFTs over one period of the primary code to perform correlations, because it can result in very high losses leading to the non-detection of the signal [4], as shown in Fig. 4.

There were different propositions to overcome this problem, well summarized in [5]. The straightforward solution consists in using two periods of the incoming primary code and one period of the primary code padded with zeros for the local code to perform the correlation [6], [4], as shown in Fig. 5. However, this solution increases the complexity and is clearly not efficient since half of the points calculated are unused. To tackle this problem, we propose a new algorithm that discards less points, by transforming the initial correlation into two or more sub-correlations. This algorithm, briefly introduced in [7], is not an approximation but an another way to compute the samples of interest, consequently there is no degradation of the sensitivity. The concept has some similarities with classical divide-and-conquer approaches such as the overlap-and-add or overlap-and-save method [8], although different.

The straightforward and proposed algorithms are first applied to the GPS L5, Galileo E5a and E5b signals, which are equivalent for the purpose of our problem because the length of their primary code is identical and they have the same modulation. Both algorithms are then applied to the Galileo E1 signal, considering the Binary Offset Carrier (BOC(1,1)) modulation and the Binary Phase Shift Keying (BPSK) modulation. The comparison of the algorithms is done for a hardware implementation in an Field Programmable Gate Array (FPGA), as well as for a software implementation using Matlab.

The paper is organized as follows. Section 2 provides the characteristics of some of the GNSS signals and a brief description of the acquisition. Sec-

tion 3 presents the PCS, the problem due to the secondary code, and the straightforward solution. Section 4 details the proposed algorithm, with a discussion on the particular case in which the FFT length is constrained to be a power of two. Section 5 includes the comparison of both algorithms for the GPS L5, Galileo E5a and E5b signals. Section 6 includes the comparison for the Galileo E1 signal processed as a BOC(1,1), and Section 7 for the Galileo E1 signal processed as a BPSK. Section 8 concludes on the results obtained and on the applicability to other GNSS signals and for other domains.

2. GNSS signals acquisition

2.1. New GNSS signals

The GPS L5, Galileo E5a and E5b signals consist each of two quadrature components (Quadrature PSK modulation), one including PRN codes and data (I channel) and the other including PRN codes only (Q channel). A detailed description of the GPS L5 signal can be found in [9] and [10], and in [11] and [12] for the Galileo E5 signal. The QPSK signal received from one satellite can be modelled as

$$\begin{aligned} s(t) = &\sqrt{P}c_{pi}(t-\tau)c_{si}(t-\tau)d(t-\tau)\cos(2\pi(f_L+f_d)t+\varphi_r)) \\ &+\sqrt{P}c_{pq}(t-\tau)c_{sq}(t-\tau)\sin(2\pi(f_L+f_d)t+\varphi_r)), \end{aligned} \tag{1}$$

where P is the total received power in Watt, c_{pi} and c_{pq} are the primary codes for the I and Q channels, c_{si} and c_{sq} are the secondary codes for the I and Q channels, d is the data, τ is a propagation delay in second, f_L is the carrier frequency in the L band in hertz, f_d is the Doppler frequency in hertz, φ_r is a phase in radian, and t is the time in second. After the front-end, which

Note : Eq. (1) contains a parenthesis in excess after the cos and sin functions.

downconverts the signal to an intermediate frequency and digitizes it, the signal is

$$\begin{aligned} s[n] = &\sqrt{P}c_{pi}[n-\tau/T_s]c_{si}[n-\tau/T_s]d[n-\tau/T_s]\cos(2\pi F_r n+\varphi)) \\ &+\sqrt{P}c_{pq}[n-\tau/T_s]c_{sq}[n-\tau/T_s]\sin(2\pi F_r n+\varphi)), \end{aligned} \tag{2}$$

where T_s is the sampling period in second, $F_r = (f_{if} + f_{offset} + f_d)T_s$ is the normalized received frequency that includes the intermediate frequency f_{if}, the offset due to the local oscillator f_{offset} and the Doppler frequency, and φ is a phase in radian (for simplicity, the noise is not included since it is not relevant for the derivations that follow). The Galileo E1 signal is slightly more complex because it includes subcarriers, causing the modulation to become a CBOC(6,1,1/11) [12]. However, the E1 signal can also be processed as a BOC(1,1) [13] or a BPSK signal [14] to reduce the receiver complexity.

The signals properties are given in Table 1, where it can be seen that the length of the primary codes is identical for the L5, E5a and E5b signals, this is why they are equivalent from the point of view of the acquisition problem discussed in this article. An illustration of the primary and secondary codes and data for the I channel of the GPS L5 signal is given Fig. 1. The three components are always synchronized, i.e. a data transition occurs only at the transition of the first chip of the secondary code, and a transition between two chips of the secondary code occurs only at the transition of the first chip of the primary code.

2.2. Acquisition

The goal of the acquisition is to detect the satellites in view and estimate the normalized Doppler frequency F_d and the code delay τ in Eq. 2. The

Note : Eq. (2) contains a parenthesis in excess after the cos and sin functions.

processing consists of three steps as shown in Fig. 2 : 1) Multiplication with a complex exponential of the same frequency as the Doppler frequency; 2) Multiplication by the primary code of the satellite searched with the same phase as the incoming code; and 3) Accumulation over one or several code periods, in order to increase the signal-to-noise ratio (SNR). Since F_d and τ are not known, different possibilities have to be tested until the signal is detected. In a serial search, the possibilities are tested one after each other. For the E1 signal processed as BOC(1,1), the local replica contains the code and the subcarrier, the rest is identical.

For a static user, the Doppler frequency is between ± 4.2 kHz for the GPS L1 C/A signal [15]. Making an adjustment for the carrier frequency (the Doppler being proportional to it) and the constellation (the Galileo satellites are slower as they have a higher altitude than the GPS satellites), the Doppler frequency is approximately ± 3.1 kHz for the L5 signal, ± 2.7 kHz for the E5a and E5b signals, and ± 3.6 kHz for the E1 signal. The offset due to the local oscillator is usually between 0 and few kHz, the limit depending on the oscillator accuracy [15]. The step between two frequencies to test depends on the integration time used. Two rules of thumb can be found in the literature, $1/(2T)$ and $2/(3T)$, where T is the integration time in second, which provides a maximum loss of 0.91 dB and 1.65 dB, respectively [15]. Regarding the primary codes, the range is their length, e.g. 10 230 chips for the L5, E5a and E5b signals, and the usual step between two phases tested is 1/2 chip for a BPSK modulation and 1/6 chip for a BOC(1,1) modulation, which provides a maximum and average loss of 2.50 dB and 1.16 dB, respectively [16].

For example, with the GPS L5 signal, using an integration time of 1

ms (which corresponds to one primary code period) and a frequency step of 500 Hz, there are 265 980 possibilities ($\lceil \frac{2\times 3100}{500} \rceil \times \frac{10\,230}{1/2}$) to test. Moreover, this process is repeated for each satellite searched (around 30 per constellation). This shows the complexity of the acquisition, and motivates the use of fast acquisition methods, such as the parallel code-phase search described in Section 3.

3. Parallel Code-phase Search

As can be seen in Fig. 2, the second part of the acquisition processing corresponds to a correlation. It is well-known that the circular correlation between two signals can be computed efficiently using an FFT, as shown in Eq. 3, where $\mathbf{h}$ and $\mathbf{x}$ are the signals to correlate, the overbar denotes the complex conjugate, mod denotes the modulo operation, N is the length of the signals and also corresponds the FFT length here, and IFFT means Inverse FFT [8]. This means that using FFTs, it is possible to obtain the correlation results for all the code phases at the same time as shown in Fig. 3, hence the name of parallel code-phase search; where $\mathbf{h}$ corresponds to the local replica $\mathbf{c_{px}}$, $\mathbf{x}$ corresponds to the signal after the multiplication with the complex exponential, and $\mathbf{y}$ corresponds to the correlation result $\mathbf{r_{F_d}}$ for all the code phases. From now on, this notation is used, because it is the one usually used for digital filtering. Calculating the correlation with Eq. 3 corresponds to fixing $\mathbf{h}$ and circularly rotating $\mathbf{x}$ to the left, or equivalently

fixing $\mathbf{x}$ and circularly rotating $\mathbf{h}$ to the right.

$$y[n] = \sum_{k=0}^{N-1} \overline{h[k]} x[(n+k) \bmod N] \quad \text{with } n = 0, ..., N-1$$
$$\mathbf{y} = \mathrm{IFFT}\Big(\overline{\mathrm{FFT}(\mathbf{h})}\ \mathrm{FFT}(\mathbf{x})\Big) \tag{3}$$

3.1. FFT length

There exist several FFT algorithms, which have a complexity of $O(L \log L)$ instead of $O(L^2)$, L being the FFT length [17]. However, they have different performance depending on the value of L. The most common algorithm is the Cooley-Tukey algorithm, which requires that L be a composite number [18]. This algorithm is more efficient when L has small prime factors. The simplest and the most common form of this algorithm is the well-known radix-2 FFT, in which the only prime factor is two, i.e. L is a power of two. The FFT provided by the FPGA and digital signal processor (DSP) companies is highly optimized for their chip and requires a length that is a power of two. There are also algorithms that can perform the FFT when L is a prime number, less efficiently however [19], [20]. A state of the art of FFT algorithms can be found in [17].

In the PCS, the FFT is performed on one period of the primary code, the FFT length is thus $L = N = f_s \times T_p$, where f_s is the sampling frequency in hertz and T_p the length of one period of the primary code in second (1 ms for the L5, E5a and E5b signals, and 4 ms for the E1 signal). With the PCS, the code step corresponds to one sample, since the minimum shift is one sample. Thus, to have a code step of 1/2 chip, the sampling frequency must be twice

the chipping rate; the FFT length is then twice the number of chips in one period of the primary code, i.e. $f_s = 20.46$ MHz and $L = 20\,460$ for the L5, E5a and E5b signals, and $f_s = 2.046$ MHz and $L = 8184$ for the E1 signal processed as BPSK. To have a code step of 1/6 chip, the sampling frequency must be six times the chipping rate, thus $f_s = 6.138$ MHz and $L = 24\,552$ for the E1 signal processed as BOC(1,1).

It should be noted that these sampling frequencies are suitable for the acquisition, but not for the positioning. Indeed, it is well-known that the sampling frequency should not be a multiple of the chipping rate because a shifted version of a code can result in the same sequence, implying poor positioning resolution [21]. Taking this into account, the minimum value for L should be increased by at least one sample.

3.2. Impact of the secondary code

As shown in Fig. 1, the secondary code implies that a transition can occur between two consecutive primary code periods. When the acquisition is performed through a serial search as depicted in Fig. 2, the correlation is not circular and the integration of the signal always starts at the first chip of the primary code, therefore there is never a transition during the integration. However, in the PCS, the incoming code used for the circular correlation can start at any chip, i.e. the signal is usually composed of portion of two different primary code periods, as shown in Fig. 4 (b). When the local replica of the primary code is aligned with the incoming primary code, the magnitude of the correlation peak is maximum only if there is no transition (Fig. 4 (a)). Else, in case of transition, the correlation peak is reduced (Fig. 4 (b)), or even vanishes if the incoming primary code starts at the middle of the period

(Fig. 4 (c)). In fact, the problem is worst than a simple non-detection, because there may be a correlation peak detected at an incorrect frequency $\hat{F_d}$ [5], which means that the receiver will start tracking the signal incorrectly and waste time before performing again an acquisition.

3.3. Straightforward solution

There have been several propositions to overcome this problem [5]. The straightforward solution consists in using two consecutive periods of the incoming primary code and one period of primary code padded with zeros for the local replica to perform the correlation [6], [4]. In this way, there is always one period of the incoming code free of transition, and thus a maximum correlation peak, as illustrated in Fig. 5 (the sign of the peak is not important, only its magnitude is). It can be seen that there is a second peak, equal to the one of Fig. 4 (b). Indeed, since there are two periods of the incoming primary code, the local code is correctly aligned twice. However, the magnitude of the first peak is always maximum, whereas the second peak can be reduced or vanish due to the transition. Since the first peak always occurs in the first half of the correlation, the second half of the correlation is discarded. This, of course, is not computationally efficient. The next section explains the proposed algorithm to obtain the first half of the correlation while discarding fewer points.

Since now two periods of the primary code are used, the minimum value for N and L is 40 920 for the L5, E5a and E5b signals, 49 104 for the E1 signal processed as BOC(1,1), and 16 368 for the E1 signal processed as BPSK. If needed, a higher value for N and L can be used, by increasing the sampling frequency (which would reduce the code step and thus the associated loss)

or by using zero-padding on both signals.

4. Proposed Algorithm

The idea of the proposed algorithm is to transform the initial correlation into two or more sub-correlations of smaller size. The proposed algorithm exploits two facts for this : 1) Half of the points of one of the signals are zero; and 2) Half of the points of the correlation output are discarded. A third fact will be exploited for the gain in efficiency, the usage of zero-padding. The proposed algorithm is not just a decomposition of the correlation as it is done in [22], because here not all the points are computed, and the algorithm modifies the values of the samples that are discarded (which has no impact on the samples of interest). The algorithm is also different from techniques such as overlap-and-add or overlap-and-save, because the sub-correlations do not correspond to different portions of the correlation, but they are all involved in all the output samples.

4.1. Algorithm to obtain two sub-correlations

The simplest form of the proposed algorithm consists in transforming the initial correlation of N points into two sub-correlations of $3N/4$ points, as detailed step by step in Appendix A. The operation performed is given by Eq. 4. Using the linearity property of the IFFT, the proposed algorithm performs thus four FFTs and one IFFT of $3N/4$ points, whereas the traditional algorithm performs two FFTs and one IFFT of N points. The first $N/2$ points of the result are identical to those of the initial correlation, while the

other points are different, which is not important since they are discarded.

$$\begin{aligned}\mathbf{y_M} &= \mathrm{IFFT}\Big(\overline{\mathrm{FFT}(\mathbf{h_0})}\ \mathrm{FFT}(\mathbf{x_0})\Big) + \mathrm{IFFT}\Big(\overline{\mathrm{FFT}(\mathbf{h_1})}\ \mathrm{FFT}(\mathbf{x_1})\Big) \\ &= \mathrm{IFFT}\Big(\overline{\mathrm{FFT}(\mathbf{h_0})}\ \mathrm{FFT}(\mathbf{x_0}) + \overline{\mathrm{FFT}(\mathbf{h_1})}\ \mathrm{FFT}(\mathbf{x_1})\Big) \end{aligned} \tag{4}$$

with

$$\begin{aligned}\mathbf{h_0} &= \Big[\ h[0] \quad h[1] \quad \dots \quad h\left[\tfrac{N}{4}-1\right] \quad \overbrace{0 \quad \dots \quad 0}^{N/2}\Big] \\ \mathbf{h_1} &= \Big[h\left[\tfrac{N}{4}\right] \quad h\left[\tfrac{N}{4}+1\right] \quad \dots \quad h\left[\tfrac{N}{2}-1\right] \quad 0 \quad \dots \quad 0\Big] \\ \mathbf{x_0} &= \Big[\ x[0] \quad x[1] \quad \dots \quad x\left[\tfrac{3N}{4}-1\right]\Big] \\ \mathbf{x_1} &= \Big[x\left[\tfrac{N}{4}\right] \quad x\left[\tfrac{N}{4}+1\right] \quad \dots \quad x\left[N-1\right]\ \Big]\end{aligned}$$

If needed, the algorithm can also perform the sub-correlations on $3N/4-1$ points, by removing the last samples of $\mathbf{h_i}$ and $\mathbf{x_i}$ $(i = 0, 1)$, as explained in Appendix B. It can also perform the sub-correlations on $3N/4 + p$ points, by adding p zeros at the end of $\mathbf{h_i}$ and $\mathbf{x_i}$ $(i = 0, 1)$. The utility of this is shown in Section 4.3.

4.2. Algorithm complexity

Considering that an FFT of N points requires approximately $N\log(N)$ multiplications, the traditional algorithm requires approximately $3N\log(N)+N$ multiplications, while the proposed algorithm requires $5\frac{3N}{4}\log(\frac{3N}{4}) + 2\frac{3N}{4}$ multiplications. The approximate number of multiplications is thus greater for the proposed algorithm.

This means that while the initial aim of finding a new algorithm was to decrease the complexity of the traditional algorithm by computing less

points, the proposed algorithm in fact requires more operations when the same correlation length N is used. However, for the cases when N needs to be increased significantly using zero-padding to obtain a specific FFT length (especially for obtaining powers of two), the proposed algorithm can be very advantageous as shown in the next sections.

4.3. Use with the radix-2 FFT

The proposed algorithm performs FFTs on $3N/4$ points. If the use of radix-2 FFTs is desired, there is the constraint given by Eq. 5, with l a positive integer. Unfortunately, this equation has no integer solutions.

$$\frac{3N}{4} = 2^l \Leftrightarrow N = \frac{4}{3}2^l \tag{5}$$

However, it has been shown that the length of the signals could also be $3N/4 - 1$. In this case, the constraint is given by Eq. 6. This equation has integer solutions if l is odd, and the result for a range of suitable values is provided in Table 2.

$$\frac{3N}{4} - 1 = 2^l \Leftrightarrow N = \frac{4}{3}(2^l + 1) \tag{6}$$

It has been shown as well that the length of the signals could also be $3N/4 + p$. For $p = 1$, the constraint is given by Eq. 7. This equation has integer solutions if l is even, and the result for a range of suitable values is provided in Table 3. The interest in adding zeros to the signals is now clearer; it gives more flexibility regarding the length of the signals.

$$\frac{3N}{4} + 1 = 2^l \Leftrightarrow N = \frac{4}{3}(2^l - 1) \tag{7}$$

To make the link with the GNSS signals, the FFT length for the traditional and proposed algorithms in function of the sampling frequency is provided in Table 4, considering a code of 1 ms. It can be seen that there are two possibilities, alternatively either the FFT length for the proposed algorithm is half the FFT length of the traditional algorithm, or the FFT lengths are identical. For example, with a sampling frequency of 21 MHz, the traditional algorithm uses FFTs of 65 536 and the proposed algorithm uses FFTs of 32 768 points; indeed, two code periods correspond to 42 000 samples, consequently, the traditional algorithm needs zero-padding up to 65 536, while the proposed algorithm needs zero-padding up to 43 692 only, using FFTs of 32 768. However, with a sampling frequency of 24 MHz, the traditional algorithm still uses FFTs of 65 536 but the proposed algorithm now uses FFTs of 65 536 points. It is clear that when the FFT lengths are identical, the proposed algorithm is not interesting since it computes more FFTs. On the other case, the complexity is reduced since the proposed algorithm requires $5\frac{N}{2}\log(\frac{N}{2}) + 2\frac{N}{2}$ multiplications, which means a reduction of at least 20 %.

4.4. Algorithm to obtain P sub-correlations

The proposed algorithm can be generalized to more than two sub-correlations, as follows. To obtain P sub-correlations, $\mathbf{h}$ and $\mathbf{x}$ must each be decomposed into P components, respectively $\mathbf{h_i}$ and $\mathbf{x_i}$ with $i = \{0, 1, ..., P-1\}$, of $\frac{P+1}{P}\frac{N}{2}$ points where :

- $\mathbf{h_i}$ contains $\frac{1}{P}\frac{N}{2}$ points of $\mathbf{h}$ and $\frac{N}{2}$ points of zeros.

- $\mathbf{x_i}$ contains $\frac{P+1}{P}\frac{N}{2}$ points of $\mathbf{x}$.
- $\mathbf{h_i}$ and $\mathbf{x_i}$ start at the samples $\frac{i}{P}\frac{N}{2}$ of $\mathbf{h}$ and $\mathbf{x}$, respectively.

The operation, defined by Eq. 8, requires $2P$ FFTs and one IFFT of $\frac{P+1}{P}\frac{N}{2}$ points.

$$\mathbf{y_{M,P}} = \mathrm{IFFT}\left(\sum_{i=0}^{P-1}\left(\overline{\mathrm{FFT}(\mathbf{h_i})}\ \mathrm{FFT}(\mathbf{x_i})\right)\right) \tag{8}$$

with

$$\mathbf{h_i} = \left[h\left[\tfrac{i}{P}\tfrac{N}{2}\right] \quad h\left[\tfrac{i}{P}\tfrac{N}{2}+1\right] \quad \ldots \quad h\left[\tfrac{i+1}{P}\tfrac{N}{2}-1\right] \quad \overbrace{0 \quad \ldots \quad 0}^{N/2}\right]$$

$$\mathbf{x_i} = \left[x\left[\tfrac{i}{P}\tfrac{N}{2}\right] \quad x\left[\tfrac{i}{P}\tfrac{N}{2}+1\right] \quad \ldots \quad x\left[\tfrac{i+1+P}{P}\tfrac{N}{2}-1\right]\right]$$

If needed, the number of points can be $\frac{P+1}{P}\frac{N}{2}-1$, requiring only to remove the last sample of $\mathbf{h_i}$ and $\mathbf{x_i}$ ($i = \{0, 1, ..., P-1\}$). The number of points can also be $\frac{P+1}{P}\frac{N}{2}+p$, requiring only to add p zeros samples at the end of $\mathbf{h_i}$ and $\mathbf{x_i}$ ($i = \{0, 1, ..., P-1\}$).

Note that as P increases, the number of output samples get closer to $N/2$, which is the number of samples of interest. However, the efficiency of the algorithm decreases because the number of FFTs increases linearly with P while the FFT length reduces only as $\frac{P+1}{P}$, consequently the number of sub-correlations should be as low as possible.

4.5. Use with the radix-2 FFT with three sub-correlations

The proposed algorithm with three sub-correlations performs FFTs on $2N/3$ points. If the use of radix-2 FFTs is desired, there is thus the constraint

given by Eq. 9, with l a positive integer. This equation has always integer solutions, and the result for a range of suitable values is provided in Table 5. A similar calculation can be done for any value of P.

$$\frac{2N}{3} = 2^l \Leftrightarrow N = \frac{3}{2}2^l = 3\ 2^{l-1} \tag{9}$$

5. Application for the acquisition of the L5, E5a and E5b signals

In this section, the traditional and proposed algorithms are compared for the acquisition of the GPS L5, Galileo E5a and E5b signals. First, we fix the correlation length (N) and the FFT length (L). Since the minimum length requires already relatively large FFTs, we will concentrate on values close to this minimum. Three cases are considered : 1) The use of the smallest FFT length (and thus correlation length), imposed by the length of the primary code and the modulation; 2) The use of the smallest FFT length that has 2 and 3 as prime factor only; this allows probably the use of a faster FFT algorithm; and 3) The use of the smallest FFT length that is a power of two, to use the fastest FFT algorithm and to check the conclusion obtained in Section 4.3.

5.1. Correlation and FFT lengths

As explained in Section 3.3, the smallest correlation length is $N = 40\,920$ for the GPS L5, Galileo E5a and E5b signals.

For the first case considered, the FFT length is 40 920 for the traditional algorithm, and 30 690 ($3/4 \times 40\,920$) for the proposed algorithm with two sub-correlations. Since the FFT lengths have relatively high prime factors

($40\,920 = 2^3 \times 3 \times 5 \times 11 \times 31$), it can be expected that the performance will be better for the next cases.

For the second case, the smallest number higher than 40 920 that has 2 and 3 as prime factors only is $41\,472 = 2^9 \times 3^4$. The FFT length is thus 41 472 for the traditional algorithm, and 31 104 ($2^7 \times 3^5$) for the proposed algorithm with two sub-correlations.

For the third case, the smallest power of two higher than 40 920 is 65 536, the FFT length is thus 65 536 for the traditional algorithm. For the proposed algorithm, according to Tables 2 and 3, the most suitable correlation length using two sub-correlations is 43 692 (suitable means the smallest correlation length (N) in the tables higher than 40 920), with an FFT length of 32 768. This means that the FFT length is divided by two compared to the traditional algorithm for this last case. These values can also be obtained using Table 4, the starting sampling frequency being 20.46 MHz.

The correlation and FFT lengths are summarized in Table 6, as well as the range of possible sampling frequencies.

5.2. Hardware implementation

In this section, the proposed and traditional algorithms are compared for an implementation inside an FPGA, the Stratix III from Altera [23]. The FFT used is the one provided by Altera, which requires a number of points that is a power of two [24]. All other FPGA companies also provide FFT with this restriction. Thus, only the third case is considered for the hardware implementation. To perform the comparison, the resources estimated by the Altera MegaWizard have been compared in terms of logic (counted as Adaptive Look-Up Table, ALUT), memory (counted as memory block of

9 Kibit, called M9K) and hardware multipliers (counted as Digital Signal Processing (DSP) elements).

The implementations of the traditional and proposed algorithms are provided in Figs. 6 and 7, respectively. The FFT followed by the conjugate operation has been replaced by an IFFT for the sake of simplicity, because the local replica is a real signal [22]. The corresponding estimated resources are provided in Table 7. It can be seen that the proposed algorithm requires more resources than the traditional one, however the processing time is divided by two since the FFT length is divided by two [22], leading to smaller acquisition time and time to first fix when the incoming signal is stored in a buffer for the acquisition [25]. Since the resources are increased by a factor lower than two (the memory is even reduced), the proposed algorithm is more efficient than the traditional one.

If an increase of the resources is not wanted, the proposed algorithm can be used with multiplexing as shown in Fig. 8. In this case, there is an additional memory, but this additional resource is largely compensated by the removal of two FFT blocks, as shown in Table 7. With multiplexing, the processing time is similar to the one of the traditional algorithm (in fact it is slightly lower due to the reduced latency of FFTs [22]).

The values provided in Table 7 are for a streaming implementation of the Altera FFT block [24], however, using the buffered or the burst implementation (which require less resources but increase the processing time) leads to similar results.

5.3. Software implementation

In this section, the proposed and traditional algorithms are compared on five different personal computers using Matlab. The FFT function of Matlab is based on the FFTW library, which has no restriction on the FFT length [26]. The average processing time over 1000 runs is shown in Fig. 9.

Focusing on the ranking of the algorithms, it can be seen that there are mainly two groups. The first, with the longest processing time, includes the traditional algorithm for a power of two length, and both algorithms for the minimum length case. This is coherent with the expectation. For $L = 65\,536$, the FFT length is far higher than the other cases, which explains a longer processing time. For the minimum FFT length case, since the length contain high prime factors, the FFT algorithm is less efficient than for the other cases. It can be noted that traditional algorithm is better than the proposed one for this case, as expected (cf. Section 4.2). The other group includes first the proposed algorithm for a power of two length, and then both algorithms for length that have small prime factors. The last two have equivalent performance, the proposed algorithm being slightly better on most of the computers, although not expected according to Section 4.2.

Considering the radix-2 FFT, the theoretical number of multiplications to perform an FFT is $\frac{L}{2}\log_2(L)$, with L the FFT length. The number of multiplications for the traditional algorithm is thus $3\frac{L}{2}\log_2(L)+L$, i.e. $1\,638\,400$ multiplications with $L = 65\,536$. The number of multiplications for the proposed algorithm is $5\frac{L}{2}\log_2(L) + 2L$, i.e. $1\,294\,336$ multiplications with $L = 32\,768$. The theoretical number of multiplications is thus reduced by 21 % (the reduction is similar for the number of additions). The result obtained

in the software implementation is thus better than foreseen by the theoretical complexity, since the proposed algorithm reduces the processing time by 39 % in average. It is difficult to explain the origin of this better performance, an hypothesis is that since the speed of a 65536-point FFT is lower than a 32-368 points FFT [26], this favors the proposed algorithm.

5.4. Case with pre-averaging

To use smaller FFTs, it is possible to perform a sum before the FFT in order to have one sample per chip [27], [28]. Since it is not possible to know where is the beginning of the chips, the process is repeated starting the sum at different phases. The proposed algorithm can be applied as well in addition to this technique. Applying this technique to the L5, E5a and E5b signals results in 10 230 points per code period. Considering the radix-2 FFT, the proposed algorithm will be thus more efficient since the FFT length will be 16 384 while the FFT length of the traditional algorithm will be 32 768 (cf Table 4, the equivalent sampling frequency after pre-averaging is 10.23 MHz).

6. Application to the acquisition of the E1 BOC(1,1) signal

In this section, a similar application study as in Section 5 is done, but applied to the E1 signal processed as BOC(1,1), which has a different correlation length.

6.1. Correlation and FFT lengths

As reported in Section 3.3, the smallest correlation length is $N = 49\,104$ for the Galileo E1 signal processed as BOC(1,1).

For the first case considered, the FFT length is 49 104 for the traditional algorithm, and 36 828 ($3/4 \times 49\,104$) for the proposed algorithm with two sub-correlations.

For the second case, the smallest number higher than 49 104 that has 2 and 3 as prime factors only is $49\,152 = 2^{14} \times 3$. The FFT length is thus 49 152 for the traditional algorithm, and 36 864 ($2^{12} \times 3^2$) for the proposed algorithm with two sub-correlations.

For the third case, the smallest power of two higher than 49 104 is 65 536. The FFT length is thus 65 536 for the traditional algorithm, as with the L5, E5a and E5b signals acquisition. For the proposed algorithm, according to Tables 2 and 3, the most suitable correlation length using two sub-correlations is 87 380 with an FFT length of 65 536. This means that the FFT length is identical to the FFT length of the traditional algorithm while there are more FFTs, this case is thus not interesting. For the proposed algorithm using three sub-correlations, according to Table 5, the most suitable correlation length is 49 152 with an FFT length of 32 768 points. This means that the FFT length is half the FFT length of the traditional algorithm, however the algorithm requires six FFTs, consequently the proposed algorithm is less efficient for the E1 BOC(1,1) signal than for the L5, E5a and E5b signals (where only four FFTs were used). The correlation and FFT lengths are summarized in Table 8.

6.2. Hardware implementation

The implementation of the traditional algorithm is identical to the one for the L5, E5a and E5b signals, and the implementation of the proposed algorithm with three sub-correlations is provided in Fig. 10. The corresponding

resources are provided in Table 9, where it can be seen that the proposed algorithm requires far more resources than the traditional one, more than a factor 2 (except for the memory). Consequently, the proposed algorithm is not more efficient than the traditional algorithm in this case.

6.3. Software implementation

The proposed and traditional algorithms are compared on five different personal computers using Matlab as in Section 5.3. The average processing time over 1000 runs is shown in Fig. 11.

The results are more heterogeneous than for the L5, E5a and E5b signals. It can be seen that however, the least three performing algorithms are the same as previously. The proposed algorithm for a power of two length is less efficient than previously, which is expected since there are more FFTs performed. Finally, the best two options are for an FFT length that has small prime factors, with an advantage for the traditional algorithm this time (which is expected according to Section 4.2).

In summary, it can be seen that the most efficient way to perform the correlation is to use lengths with small prime factors and limited zero-padding. Here, the small prime factors are 2 and 3, but 5 and 7 can be also considered. With this wide variety of length available, it is rarely possible to reduce the zero-padding with the proposed algorithm to lead to better performance (although possible in some specific cases, but the gain will be marginal).

Considering the radix-2 FFT, the theoretical number of multiplications for the traditional algorithm with $L = 65\,536$ is $1\,638\,400$, as in Section 5. The number of multiplications for the proposed algorithm using three sub-correlations is $7\frac{L}{2}\log_2(L) + 3L$, i.e. $1\,818\,624$ multiplications with $L =$

32 768. The theoretical number of multiplications is thus increased by 11 %. The result obtained in the software implementation is thus better than foreseen by the theoretical complexity, since the proposed algorithm reduces the processing time by 15 % in average.

6.4. Modifying the correlation length

The minimum correlation length was fixed according to the code length and the code step. The code length is fixed, however, the code step can be modified, impacting at the same time the code alignment loss.

The code step used for the E1 BOC(1,1) signal was 1/6 chip. By increasing the code step in order to obtain a correlation length of $N = 43\,692$ instead of 49 104, the comparison between the traditional and proposed algorithm would be the same as for the L5, E5a and E5b signals. The code step corresponding to this length is about 1/5.34 chip ($2 \times 4092/43\,692$), resulting in a maximum and average loss of 2.87 dB and 1.31 dB, respectively. This is very close to the loss using a code step of 1/6 chip, namely 2.50 dB at maximum and 1.16 dB in average.

Consequently, it is possible to use a slightly larger step for the Galileo E1 BOC(1,1) signal, in order to be able to use the proposed algorithm with two sub-correlations and smaller FFTs, as for the L5, E5a and E5b signals.

7. Application to the acquisition of the E1 BPSK signal

As reported in Section 3.3, the smallest correlation length is $N = 16\,368$ for the Galileo E1 signal processed as BPSK.

For the first case considered, the FFT length is 16 368 for the traditional algorithm, and 12 276 ($3/4 \times 16\,368$) for the proposed algorithm with two

sub-correlations.

For the second case, the smallest number higher than 16 368 that has 2 and 3 as prime factors only is $16\,384 = 2^{14}$, which in fact has just 2 as prime factor. The FFT length is thus 16 384 for the traditional algorithm, and 12 288 ($2^{12} \times 3$) for the proposed algorithm with two sub-correlations. Since the FFT length for the traditional algorithm is a power of two, it is expected that the proposed algorithm will be less efficient for this case.

For the third case, the smallest power of two higher than 16 368 is 16 384. The FFT length is thus 16 384 for the traditional algorithm, as with the second case. For the proposed algorithm, according to Tables 2 and 3, the most suitable correlation length using two sub-correlations is 21 844, with an FFT length of 16 384. This means that the FFT length is identical to the FFT length of the traditional algorithm while there are more FFTs, this case is thus not interesting. For the proposed algorithm using three sub-correlations, according to Table 5, the most suitable correlation length is 24 576 still with an FFT length of 16 384 points. In fact, to halve the FFT length, it would require to have at least 1023 sub-correlations, which is clearly not efficient. The proposed algorithm is thus less efficient for this case also.

Consequently, the proposed algorithm is not efficient for the acquisition of the E1 signal processed as BPSK considering the minimum sampling frequency, 2.046 MHz. However, such a low sampling frequency has an impact on the positioning accuracy, it is thus common to use a higher frequency. In this case, the choice for the best algorithm with the radix-2 FFT can be found using Table 4.

8. Conclusions

In this article, we discussed the problem of decreasing the complexity of the acquisition of the new GPS and Galileo signals that have a secondary code that is not exploited. The straightforward solution of doubling the size of the correlation and discarding half of the points calculated is not satisfying, which led us to look for an algorithm that discards less points. The proposed algorithm transforms the initial correlations into two or more smaller sub-correlations, without loss of sensitivity since the samples of interest are computed exactly.

The proposed algorithm is more efficient than the traditional algorithm when the latter requires significant zero-padding. The zero-padding depends on the sampling frequency and the type of FFT used, and can be significant mainly when the radix-2 FFT is used. Considering then the radix-2 FFT, the proposed algorithm is more efficient for half of the possible sampling frequencies, which is interesting for hardware and DSP based receivers.

For the GPS L5, Galileo E5a and E5b signals, the minimum sampling frequency, 20.46 MHz, is in a range where the proposed algorithm is more efficient. It has been shown that the theoretical number of operations is reduced by 21 %, the memory requirement is almost halved for an FPGA implementation, and the processing time is reduced by 39 % in average on personal computers.

For the Galileo E1 signal processed as BOC(1,1), the minimum sampling frequency considered, 6.138 MHz, is in a range where the proposed algorithm is not more efficient. However, if the sampling frequency is decreased to 5.4615 MHz at the expense of an average loss 0.15 dB, this sampling frequency

is in a range where the proposed algorithm is more efficient, with the same reduction as for the L5, E5a and E5b signals.

For the Galileo E1 signal processed as BPSK, the minimum sampling frequency considered, 2.046 MHz, is in a range where the proposed algorithm is not more efficient. For higher sampling frequencies, it is easy to check if the proposed algorithm is more efficient or not using the tables provided in this article.

Using similar analysis as done in this article, it is easy to determine if the proposed algorithm is suitable for other and future GNSS signals. For example, the E6 CS signal is a good candidate since the minimum correlation length is half the one for the L5, E5a and E5B signals. The proposed algorithm may be also interesting for the E6 PRS signal, the GPS M signal or the future GLONASS CDMA and BeiDou signals.

The problem discussed in this article does not restrict to GNSS and to the Parallel Code-phase Search. Any system that performs a circular correlation (or a convolution) between two signals where one of them has half of zeros and where half of the output is discarded can use the proposed algorithm. For example, this problem is also present in the Double Block Zero Padding acquisition method [29].

Acknowledgment

The authors thanks Myriam Foucras from ABBIA GNSS Technologies / ENAC, and Sara Grassi and Youssef Tawk from the ESPLAB of the EPFL for their comments and suggestions that greatly improved this article.

Appendix A. Proposed algorithm to obtain two sub-correlations

The circular correlation between two sequences, $\mathbf{h}$ and $\mathbf{x}$, can be defined by Eq. A.1, where the overbar denotes the complex conjugate, mod denotes the modulo operation, N is the length of the sequences and the FFT length.

$$y[n] = \sum_{k=0}^{N-1} \overline{h[k]} x[(n+k) \bmod N] \quad \text{with } n = 0, ..., N-1$$
$$\mathbf{y} = \mathrm{IFFT}\Big(\overline{\mathrm{FFT}(\mathbf{h})}\ \mathrm{FFT}(\mathbf{x})\Big) \tag{A.1}$$

The circular correlation can also be written using matrix notation, as shown in Eq. A.2, where $\mathbf{y}$ is a vector of N points, $\overline{\mathbf{H}}$ is a circular matrix of $N \times N$, and $\mathbf{x}$ is a vector of N points.

$$\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[N-2] \\ y[N-1] \end{bmatrix} = \begin{bmatrix} \overline{h[0]} & \overline{h[1]} & \cdots & \overline{h[N-2]} & \overline{h[N-1]} \\ \overline{h[N-1]} & \overline{h[0]} & \cdots & \overline{h[N-3]} & \overline{h[N-2]} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \overline{h[2]} & \overline{h[3]} & \dots & \overline{h[0]} & \overline{h[1]} \\ \overline{h[1]} & \overline{h[2]} & \dots & \overline{h[N-1]} & \overline{h[0]} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-2] \\ x[N-1] \end{bmatrix}$$
$$\mathbf{y} = \overline{\mathbf{H}}\ \mathbf{x} \tag{A.2}$$

The operation explained in Section 3.3, consists in performing a circular correlation where half of one of the signal is zeros. It can be then formulated by Eq. A.3, which is similar to Eq. A.2, with $h[n] = 0$ for $N/2 < n < N-1$.

$$\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[\frac{N}{2}-1] \\ y[\frac{N}{2}] \\ \vdots \\ y[N-2] \\ y[N-1] \end{bmatrix} = \begin{bmatrix} \overline{h[0]} & \overline{h[1]} & \cdots & \overline{h[\frac{N}{2}-1]} & 0 & \cdots & 0 & 0 \\ 0 & \overline{h[0]} & \cdots & \overline{h[\frac{N}{2}-2]} & \overline{h[\frac{N}{2}-1]} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & \overline{h[0]} & \overline{h[1]} & \ldots & \overline{h[\frac{N}{2}-1]} & 0 \\ 0 & 0 & \ldots & 0 & \overline{h[0]} & \ldots & \overline{h[\frac{N}{2}-2]} & \overline{h[\frac{N}{2}-1]} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \overline{h[2]} & \overline{h[3]} & \ldots & 0 & 0 & \ldots & \overline{h[0]} & \overline{h[1]} \\ \overline{h[1]} & \overline{h[2]} & \ldots & 0 & 0 & \ldots & 0 & \overline{h[0]} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[\frac{N}{2}-1] \\ x[\frac{N}{2}] \\ \vdots \\ x[N-2] \\ x[N-1] \end{bmatrix}$$

$$\mathbf{y} = \overline{\mathbf{H}}\,\mathbf{x} \tag{A.3}$$

First step : Use of the first half of the output only

As explained in Section 3.3, the second half of $\mathbf{y}$ is not used in our acquisition problem. Consequently, the first step consists in removing the unnecessary rows in the vector $\mathbf{y}$ and the matrix $\overline{\mathbf{H}}$, which gives Eq. A.4. At this stage, $\mathbf{y_T}$ (T for truncated) is a vector of $N/2$ points, $\overline{\mathbf{H_T}}$ is a matrix of $N/2 \times N$, and $\mathbf{x}$ is a vector of N points.

$$\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[\frac{N}{2}-1] \end{bmatrix} = \begin{bmatrix} \overline{h[0]} & \overline{h[1]} & \cdots & \overline{h[\frac{N}{2}-1]} & 0 & \cdots & 0 & 0 \\ 0 & \overline{h[0]} & \cdots & \overline{h[\frac{N}{2}-2]} & \overline{h[\frac{N}{2}-1]} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \overline{h[0]} & \overline{h[1]} & \cdots & \overline{h[\frac{N}{2}-1]} & 0 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[\frac{N}{2}-1] \\ x[\frac{N}{2}] \\ \vdots \\ x[N-1] \end{bmatrix}$$

$$\mathbf{y_T} = \overline{\mathbf{H_T}}\,\mathbf{x} \tag{A.4}$$

Note : In Eq. (A.4), the sample $x[N\text{-}2]$ is missing in the vector $\mathbf{x}$.

Second step : Separation of the matrix

The second step consists in separating the matrix $\overline{\mathbf{H_T}}$ in two matrices, $\overline{\mathbf{H'_{T0}}}$ and $\overline{\mathbf{H'_{T1}}}$, where $\overline{\mathbf{H'_{T0}}}$ is $\overline{\mathbf{H_T}}$ with $h[n] = 0$ for $N/4 < n < N/2 - 1$, and $\overline{\mathbf{H'_{T1}}}$ is $\overline{\mathbf{H_T}}$ with $h[n] = 0$ for $0 < n < N/4 - 1$, such that $\overline{\mathbf{H_T}}$ is the sum of $\overline{\mathbf{H'_{T0}}}$ and $\overline{\mathbf{H'_{T1}}}$, as shown in Eq. A.5.

$$
\begin{aligned}
\mathbf{y_T} &= \overline{\mathbf{H_T}}\ \mathbf{x} \\
&= (\overline{\mathbf{H'_{T0}}} + \overline{\mathbf{H'_{T1}}})\ \mathbf{x} \\
&= \left(\begin{bmatrix}
\overline{h[0]} & \overline{h[1]} & \cdots & 0 & 0 & \cdots & 0 & 0 \\
0 & \overline{h[0]} & \cdots & 0 & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & \overline{h[0]} & \overline{h[1]} & \cdots & 0 & 0
\end{bmatrix} \right. \\
&\quad \left. + \begin{bmatrix}
0 & 0 & \cdots & \overline{h[\frac{N}{2}-1]} & 0 & \cdots & 0 & 0 \\
0 & 0 & \cdots & \overline{h[\frac{N}{2}-2]} & \overline{h[\frac{N}{2}-1]} & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & 0 & \cdots & \overline{h[\frac{N}{2}-1]} & 0
\end{bmatrix} \right)
\begin{bmatrix}
x[0] \\ x[1] \\ \vdots \\ x[\frac{N}{2}-1] \\ x[\frac{N}{2}] \\ \vdots \\ x[N-2] \\ x[N-1]
\end{bmatrix}
\end{aligned}
\tag{A.5}
$$

These two matrices contain thus $N/4$ columns of zeros, without counting the last column that was already present in $\overline{\mathbf{H_T}}$. It is thus possible to remove these columns from $\overline{\mathbf{H'_{T0}}}$ and $\overline{\mathbf{H'_{T1}}}$ to obtain the matrices $\overline{\mathbf{H_{T0}}}$ and $\overline{\mathbf{H_{T1}}}$, and to remove the corresponding rows of $\mathbf{x}$ to obtain the vectors $\mathbf{x_0}$ and $\mathbf{x_1}$, as

shown in Eq. A.6. At this stage, $\mathbf{y_T}$ is a vector of $N/2$ points, $\overline{\mathbf{H_{T0}}}$ and $\overline{\mathbf{H_{T1}}}$ are matrices of $N/2 \times 3N/4$, and $\mathbf{x_0}$ and $\mathbf{x_1}$ are vectors of $3N/4$ points.

$$
\begin{aligned}
\mathbf{y_T} &= (\overline{\mathbf{H'_{T0}}} + \overline{\mathbf{H'_{T1}}})\ \mathbf{x} \\
&= \overline{\mathbf{H_{T0}}}\ \mathbf{x_0} + \overline{\mathbf{H_{T1}}}\ \mathbf{x_1} \\
&= \begin{bmatrix}
\overline{h[0]} & \overline{h[1]} & \cdots & \overline{h[\frac{N}{4}-1]} & 0 & \cdots & 0 & 0 \\
0 & \overline{h[0]} & \cdots & \overline{h[\frac{N}{4}-2]} & \overline{h[\frac{N}{4}-1]} & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & 0 & 0 & \ldots & \overline{h[\frac{N}{4}-1]} & 0
\end{bmatrix}
\begin{bmatrix}
x[0] \\ x[1] \\ \vdots \\ x[\frac{N}{4}-1] \\ x[\frac{N}{4}] \\ \vdots \\ x[\frac{3N}{4}-2] \\ x[\frac{3N}{4}-1]
\end{bmatrix} \\
&+ \begin{bmatrix}
\overline{h[\frac{N}{4}]} & \overline{h[\frac{N}{4}+1]} & \cdots & \overline{h[\frac{N}{2}-1]} & 0 & \cdots & 0 & 0 \\
0 & \overline{h[\frac{N}{4}]} & \cdots & \overline{h[\frac{N}{2}-2]} & \overline{h[\frac{N}{2}-1]} & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & 0 & 0 & \ldots & \overline{h[\frac{N}{2}-1]} & 0
\end{bmatrix}
\begin{bmatrix}
x[\frac{N}{4}] \\ x[\frac{N}{4}+1] \\ \vdots \\ x[\frac{N}{2}-1] \\ x[\frac{N}{2}] \\ \vdots \\ x[N-2] \\ x[N-1]
\end{bmatrix}
\end{aligned}
\tag{A.6}
$$

Third step : Making the matrices circular

From Eq. A.6, it can be seen that the matrices $\overline{\mathbf{H_{T0}}}$ and $\overline{\mathbf{H_{T1}}}$ have a circular pattern. It is thus possible to transform them to the circular matrices

$\overline{\mathbf{H_0}}$ and $\overline{\mathbf{H_1}}$ by adding $N/4$ rows, as shown in Eqs. A.7 and A.8.

$$\overline{\mathbf{H_0}} = \begin{bmatrix} \overline{h[0]} & \overline{h[1]} & \cdots & \overline{h[\frac{N}{4}-1]} & 0 & \cdots & 0 & 0 \\ 0 & \overline{h[0]} & \cdots & \overline{h[\frac{N}{4}-2]} & \overline{h[\frac{N}{4}-1]} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & 0 & \ldots & \overline{h[\frac{N}{4}-1]} & 0 \\ 0 & 0 & \ldots & 0 & 0 & \ldots & \overline{h[\frac{N}{4}-2]} & \overline{h[\frac{N}{4}-1]} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \overline{h[2]} & \overline{h[3]} & \cdots & 0 & 0 & \cdots & \overline{h[0]} & \overline{h[1]} \\ \overline{h[1]} & \overline{h[2]} & \cdots & 0 & 0 & \cdots & 0 & \overline{h[0]} \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}a\\a\\a\\a\\a\end{matrix}}\right\} N/2 \text{ rows} \\ \left.\vphantom{\begin{matrix}a\\a\\a\end{matrix}}\right\} N/4 \text{ rows} \end{matrix} \tag{A.7}$$

$$\overline{\mathbf{H_1}} = \begin{bmatrix} \overline{h[\frac{N}{4}]} & \overline{h[\frac{N}{4}+1]} & \cdots & \overline{h[\frac{N}{2}-1]} & 0 & \cdots & 0 & 0 \\ 0 & \overline{h[\frac{N}{4}]} & \cdots & \overline{h[\frac{N}{2}-2]} & \overline{h[\frac{N}{2}-1]} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & 0 & \ldots & \overline{h[\frac{N}{2}-1]} & 0 \\ 0 & 0 & \ldots & 0 & 0 & \ldots & \overline{h[\frac{N}{2}-2]} & \overline{h[\frac{N}{2}-1]} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \overline{h[\frac{N}{4}+2]} & \overline{h[\frac{N}{4}+3]} & \cdots & 0 & 0 & \cdots & \overline{h[\frac{N}{4}]} & \overline{h[\frac{N}{4}+1]} \\ \overline{h[\frac{N}{4}+1]} & \overline{h[\frac{N}{4}+2]} & \cdots & 0 & 0 & \cdots & 0 & \overline{h[\frac{N}{4}]} \end{bmatrix} \tag{A.8}$$

Eq. A.6 can then be modified to obtain Eq. A.9. At this stage, $\mathbf{y_M}$ (M for modified) is a vector of $3N/4$ points, $\overline{\mathbf{H_0}}$ and $\overline{\mathbf{H_1}}$ are circular matrices of $3N/4 \times 3N/4$, and $\mathbf{x_0}$ and $\mathbf{x_1}$ are vectors of $3N/4$ points. The vector $\mathbf{y_M}$ is composed of the initial vector $\mathbf{y_T}$ of $N/2$ points, and of the vector $\mathbf{y_D}$ (D for discarded) of $N/4$ points.

$$\mathbf{y_M} = \begin{bmatrix} \mathbf{y_T} \\ \mathbf{y_D} \end{bmatrix} = \overline{\mathbf{H_0}}\ \mathbf{x_0} + \overline{\mathbf{H_1}}\ \mathbf{x_1}$$

$$\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[\frac{N}{2}-1] \\ y_D[0] \\ \vdots \\ y_D[\frac{N}{4}-2] \\ y_D[\frac{N}{4}-1] \end{bmatrix} = \begin{bmatrix} \overline{h[0]} & \overline{h[1]} & \cdots & \overline{h[\frac{N}{4}-1]} & 0 & \cdots & 0 & 0 \\ 0 & \overline{h[0]} & \cdots & \overline{h[\frac{N}{4}-2]} & \overline{h[\frac{N}{4}-1]} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & \overline{h[\frac{N}{4}-1]} & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & \overline{h[\frac{N}{4}-2]} & \overline{h[\frac{N}{4}-1]} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \overline{h[2]} & \overline{h[3]} & \cdots & 0 & 0 & \cdots & \overline{h[0]} & \overline{h[1]} \\ \overline{h[1]} & \overline{h[2]} & \cdots & 0 & 0 & \cdots & 0 & \overline{h[0]} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[\frac{N}{4}-1] \\ x[\frac{N}{4}] \\ \vdots \\ x[\frac{3N}{4}-2] \\ x[\frac{3N}{4}-1] \end{bmatrix}$$

$$+ \begin{bmatrix} \overline{h[\frac{N}{4}]} & \overline{h[\frac{N}{4}+1]} & \cdots & \overline{h[\frac{N}{2}-1]} & 0 & \cdots & 0 & 0 \\ 0 & \overline{h[\frac{N}{4}]} & \cdots & \overline{h[\frac{N}{2}-2]} & \overline{h[\frac{N}{2}-1]} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & \overline{h[\frac{N}{2}-1]} & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & \overline{h[\frac{N}{2}-2]} & \overline{h[\frac{N}{2}-1]} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \overline{h[\frac{N}{4}+2]} & \overline{h[\frac{N}{4}+3]} & \cdots & 0 & 0 & \cdots & \overline{h[\frac{N}{4}]} & \overline{h[\frac{N}{4+1}]} \\ \overline{h[\frac{N}{4}+1]} & \overline{h[\frac{N}{4}+2]} & \cdots & 0 & 0 & \cdots & 0 & \overline{h[\frac{N}{4}]} \end{bmatrix} \begin{bmatrix} x[\frac{N}{4}] \\ x[\frac{N}{4}+1] \\ \vdots \\ x[\frac{N}{2}-1] \\ x[\frac{N}{2}] \\ \vdots \\ x[N-2] \\ x[N-1] \end{bmatrix} \tag{A.9}$$

Coming back with the initial notation, the operation performed in Eq. A.9 can be computed using FFTs as shown by Eq. A.10

$$\begin{aligned} \mathbf{y_M} &= \mathrm{IFFT}\Big(\overline{\mathrm{FFT}(\mathbf{h_0})}\ \mathrm{FFT}(\mathbf{x_0})\Big) + \mathrm{IFFT}\Big(\overline{\mathrm{FFT}(\mathbf{h_1})}\ \mathrm{FFT}(\mathbf{x_1})\Big) \\ &= \mathrm{IFFT}\Big(\overline{\mathrm{FFT}(\mathbf{h_0})}\ \mathrm{FFT}(\mathbf{x_0}) + \overline{\mathrm{FFT}(\mathbf{h_1})}\ \mathrm{FFT}(\mathbf{x_1})\Big) \end{aligned} \tag{A.10}$$

with

$$\mathbf{h_0} = \Big[\; h[0] \qquad h[1] \qquad \dots \quad h\left[\tfrac{N}{4}-1\right] \quad \overbrace{0 \quad \dots \quad 0}^{N/2} \Big]$$

$$\mathbf{h_1} = \Big[h\left[\tfrac{N}{4}\right] \quad h\left[\tfrac{N}{4}+1\right] \quad \dots \quad h\left[\tfrac{N}{2}-1\right] \quad 0 \quad \dots \quad 0 \Big]$$

$$\mathbf{x_0} = \Big[\; x[0] \qquad x[1] \qquad \dots \quad x\left[\tfrac{3N}{4}-1\right] \Big]$$

$$\mathbf{x_1} = \Big[x\left[\tfrac{N}{4}\right] \quad x\left[\tfrac{N}{4}+1\right] \quad \dots \quad x\left[N-1\right] \Big]$$

This means that instead of discarding half of the points calculated ($\frac{N/2}{N}$), the proposed algorithm discards only one third ($\frac{N/4}{3N/4}$).

Appendix B. Options with the signals length

Going back to Eq. A.6, it can be seen that the last column of $\overline{\mathbf{H_{T0}}}$ and $\overline{\mathbf{H_{T1}}}$ contains only zeros, consequently, this column and the last point of $\mathbf{x_0}$ and $\mathbf{x_1}$ can be removed without modifying the result. In this case, $\overline{\mathbf{H_0}}$ and $\overline{\mathbf{H_1}}$ are circular matrices of $3N/4-1 \times 3N/4-1$, $\mathbf{x_0}$ and $\mathbf{x_1}$ are vectors of $3N/4-1$ points, and $\mathbf{y_M}$ is a vector of $3N/4-1$ points, without impacting the wanted result $\mathbf{y_T}$, modifying only the discarded part $\mathbf{y_D}$, which now contains $N/4-1$ points.

Conversely, it is also possible to add columns of zeros to $\overline{\mathbf{H_{T0}}}$ and $\overline{\mathbf{H_{T1}}}$ after their last column, and add p points of any value at the end of $\mathbf{x_0}$ and $\mathbf{x_1}$. By adding p columns, the length of the signals for the sub-correlations becomes $3N/4+p$. The utility of this is shown in Section 4.3.

References

[1] K. Borre, D. Akos, N. Bertelsen, P. Rinder, S. Jensen, A software-defined GPS and Galileo receiver. Single-frequency approach, Birkhäuser Boston, 2007.

[2] D. Borio, M-sequence and secondary code constraints for GNSS signal acquisition, IEEE Transactions on Aerospace and Electronic Systems 47 (2) (2011) 928–945.

[3] A. Ayaz, T. Pany, B. Eissfeller, Assessment of GNSS signal acquisition sensitivities for indoor and urban scenarios, in: IEEE 21st International Symposium on Personal, Indoor and Mobile Radio Communications Workshops (PIMRC Workshops), 2010, pp. 223–227.

[4] D. Borio, M. Fantino, L. Lo Presti, The impact of the Galileo signal in space in the acquisition system, in: E. Re, M. Ruggieri (Eds.), Satellite Communications and Navigation Systems, Signals and Communication Technology, Springer US, 2008, pp. 151–167.

[5] L. Lo Presti, X. Zhu, M. Fantino, P. Mulassano, GNSS signal acquisition in the presence of sign transition, IEEE Journal of Selected Topics in Signal Processing 3 (4) (2009) 557–570.

[6] C. Yang, C. Hegarty, M. Tran, Acquisition of the GPS L5 signal using coherent combining of I5 and Q5, in: Proceedings of the 17th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2004), 2004, pp. 2184–2195.

[7] J. Leclère, C. Botteron, P.-A. Farine, Modified parallel code-phase search for acquisition in presence of sign transition, in: International Conference on Localization and GNSS (ICL-GNSS), 2013, pp. 1–6.

[8] J. G. Proakis, D. G. Manolakis, Digital signal processing: Principles, algorithms, and applications, 4th edition, Prentice Hall, 2006.

[9] J. J. J. Spilker, A. J. van Dierendonck, Proposed new L5 civil GPS codes, NAVIGATION, Journal of The Institute of Navigation 48 (3) (2001) 135–144.

[10] SAIC, Navstar GPS space segment / user segment L5 interfaces, Interface specification IS-GPS-705 Revision A (June 2010).

[11] N. C. Shivaramaiah, A. G. Dempster, The Galileo E5 AltBOC: Understanding the signal structure, in: Proceedings of the International Global Navigation Satellite Systems Society (IGNSS 2009), 2009.

[12] EU, European GNSS (Galileo) open service, signal in space interface control document, Issue 1.1 (September 2010).

[13] O. Julien, C. Macabiau, J.-L. Issler, L. Ries, Galileo E1 OS/SoL acquisition, tracking and data demodulation performances for civil aviation, in: Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC), 2010 5th ESA Workshop on, 2010, pp. 1–8.

[14] N. Martin, V. Leblond, G. Guillotel, V. Heiries, BOC(x,y) signal acquisition techniques and performances, in: Proceedings of the 16th Inter-

national Technical Meeting of the Satellite Division of the Institute of Navigation (ION GPS/GNSS 2003), 2013, pp. 188 – 198.

[15] F. van Diggelen, A-GPS: Assisted GPS, GNSS, and SBAS, GNSS Technology and Applications Series, Artech House, 2009.

[16] W. De Wilde, J.-M. Sleewaegen, A. Simsky, J. Van Hees, C. Vandewiele, E. Peeters, J. Grauwen, F. Boon, Fast signal acquisition technology for new GPS/Galileo receivers, in: IEEE/ION Position, Location, And Navigation Symposium (PLANS), 2006, pp. 1074–1079.

[17] P. Duhamel, M. Vetterli, Fast Fourier transforms: A tutorial review and a state of the art, Signal Processing 19 (4) (1990) 259–299.

[18] J. W. Cooley, J. W. Tukey, An algorithm for the machine calculation of complex Fourier series, Math. Comp. 19 (1965) 297–301.

[19] C. M. Rader, Discrete Fourier transforms when the number of data samples is prime, Proceedings of the IEEE 56 (6) (1968) 1107–1108.

[20] L. Bluestein, A linear filtering approach to the computation of discrete Fourier transform, IEEE Transactions on Audio and Electroacoustics 18 (4) (1970) 451–455.

[21] D. M. Akos, M. Pini, Effect of sampling frequency on GNSS receiver performance, NAVIGATION, Journal of The Institute of Navigation 53 (2) (2006) 85–96.

[22] J. Leclère, C. Botteron, 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: Proceedings of the 25th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2012), 2012, pp. 1406–1416.

[23] Altera, Stratix III device handbook (March 2011).

[24] Altera, FFT MegaCore function user guide (November 2012).

[25] J. Leclère, C. Botteron, P.-A. Farine, Comparison framework of FPGA-based GNSS signals acquisition architectures, IEEE Transactions on Aerospace and Electronic Systems 49 (3).

[26] M. Frigo, S. Johnson, FFTW: An adaptive software architecture for the FFT, in: Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 3, 1998, pp. 1381–1384.

[27] J. Starzyk, Z. Zhu, Averaging correlation for C/A code acquisition and tracking in frequency domain, in: Proceedings of the 44th IEEE Midwest Symposium on Circuits and Systems (MWSCAS 2001), Vol. 2, 2001, pp. 905–908.

[28] C. Hegarty, M. Tran, A. van Dierendonck, Acquisition algorithms for the GPS L5 signal, in: Proceedings of the 16th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GPS/GNSS 2003), 2003, pp. 165–177.

[29] M. Foucras, O. Julien, C. Macabiau, B. Ekambi, A novel computationally efficient Galileo E1 OS acquisition method for GNSS software receiver, in: Proceedings of the 25th International Technical Meeting of

the Satellite Division of the Institute of Navigation (ION GNSS 2012), 2012, pp. 365–383.

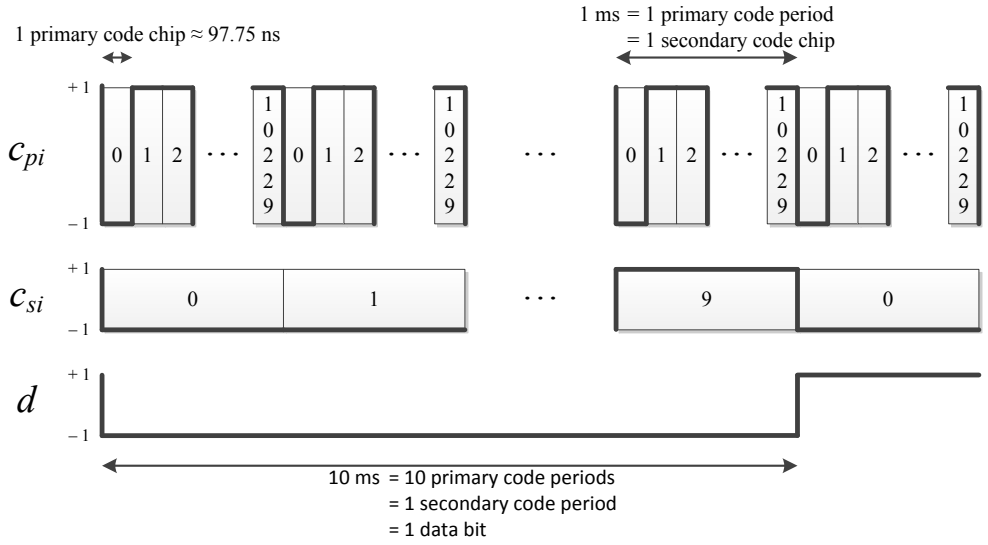

Figure 1: Illustration of the primary code, secondary code and data for the GPS L5 signal (I channel)

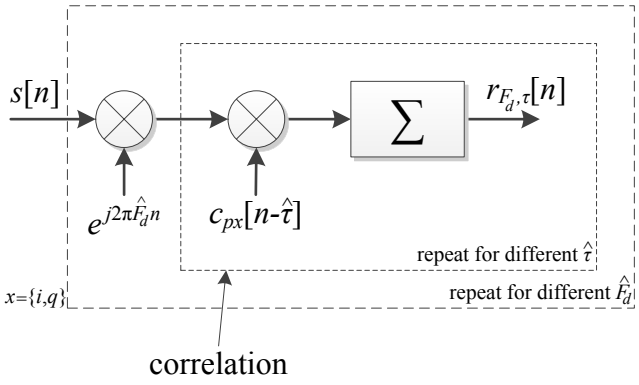

Figure 2: Serial search acquisition

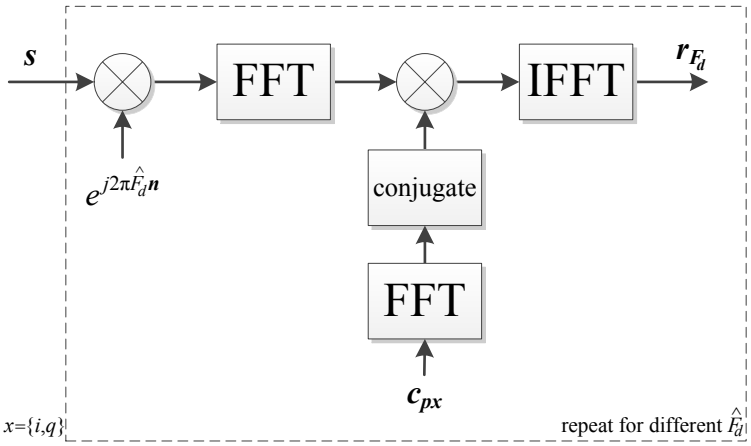

Figure 3: Parallel code-phase search acquisition

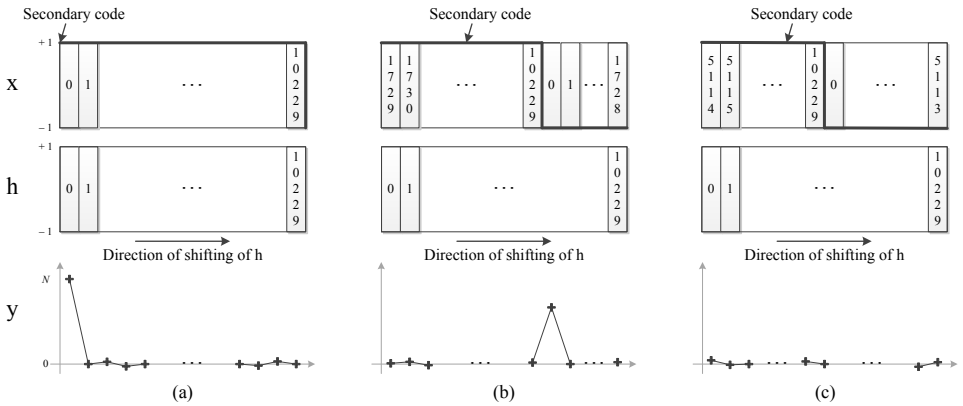

Figure 4: Illustration of the problem due to the secondary code transition. (a) By chance, the incoming primary code starts with the first chip, the correlation at the correct alignment is maximum. (b) The incoming primary code does not start at the first chip (usual case), the correlation at the correct alignment is reduced. (c) In the worst case, the incoming primary code starts at the middle of a period, the correlation at the correct alignment is 0.

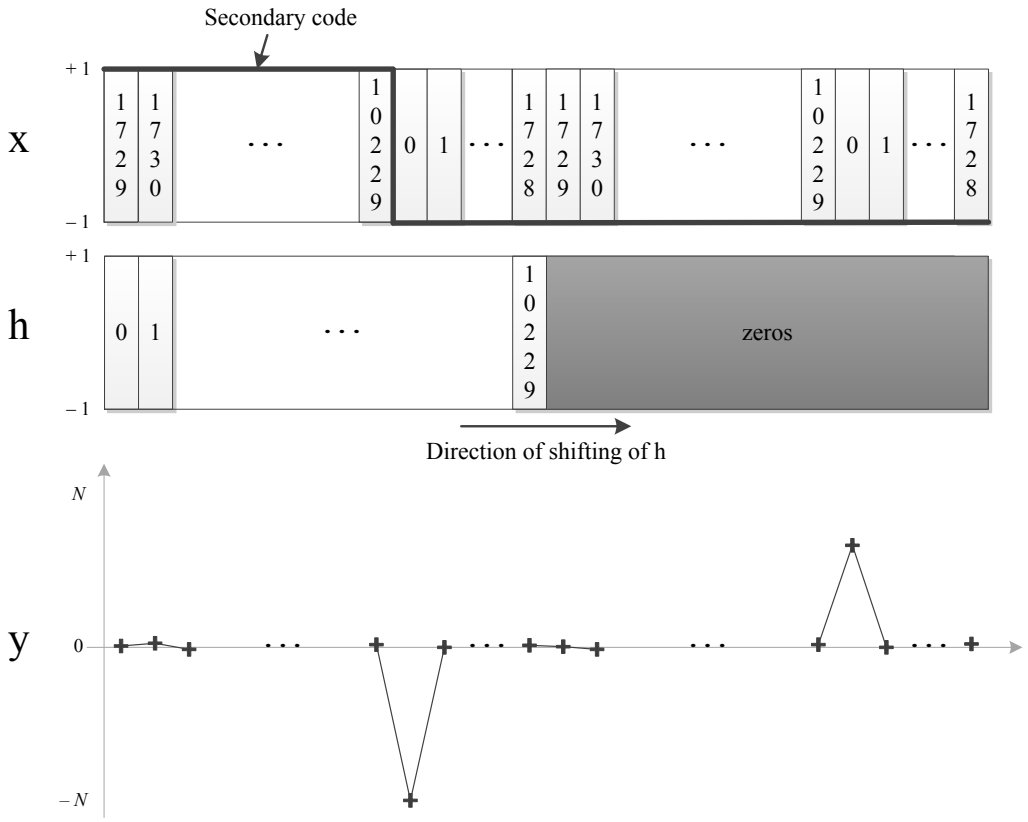

Figure 5: Straightforward solution to the secondary code transition problem. The magnitude of the first peak is always maximum, whereas the second peak can be reduced due to transition.

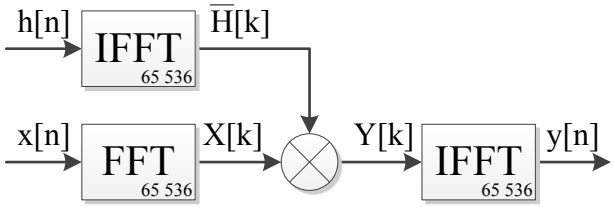

Figure 6: Implementation of the traditional algorithm for the GPS L5, Galileo E5a and E5b signals

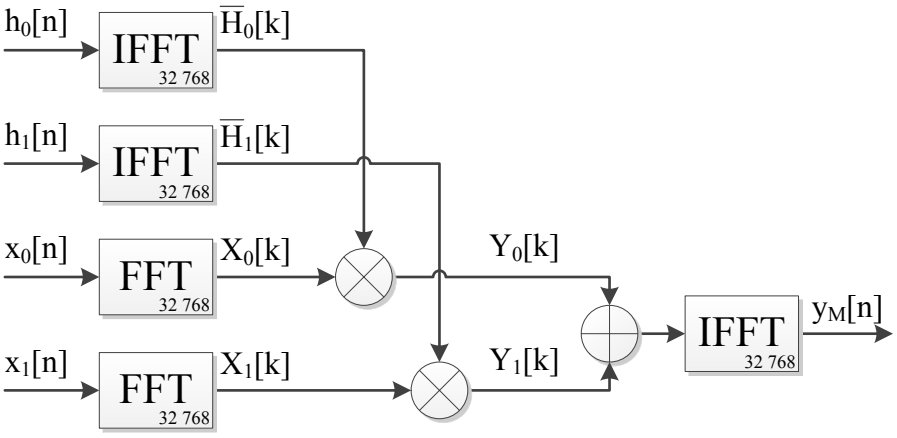

Figure 7: Implementation of the proposed algorithm for the GPS L5, Galileo E5a and E5b signals

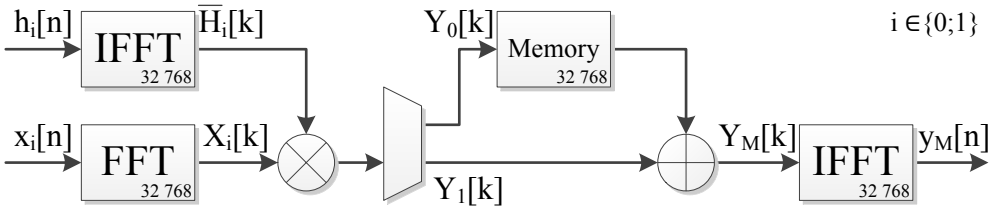

Figure 8: Implementation of the proposed algorithm with multiplexing for the GPS L5, Galileo E5a and E5b signals

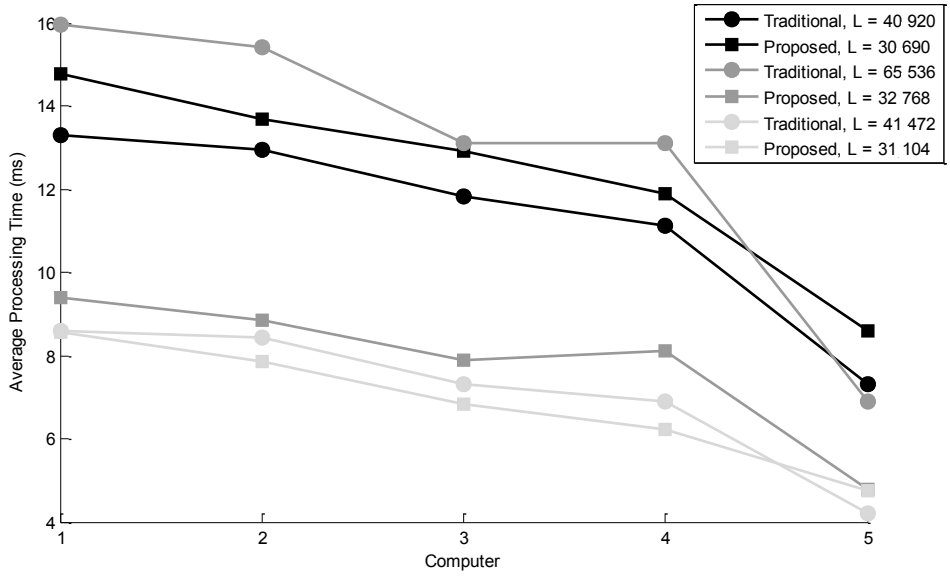

Figure 9: Average processing time of the algorithms on different computers for the GPS L5, Galileo E5a and E5b signals (L is the FFT length). The CPU of the computers are respectively : Core 2 Duo E4600 @ 2.40 GHz, QuadCore Xeon E5430 @ 2.66 GHz, Core 2 Duo E6700 @ 2.66 GHz, QuadCore Xeon E5430 2.66 GHz, Mobile Dual Core i7-2620M @ 3.2 GHz.

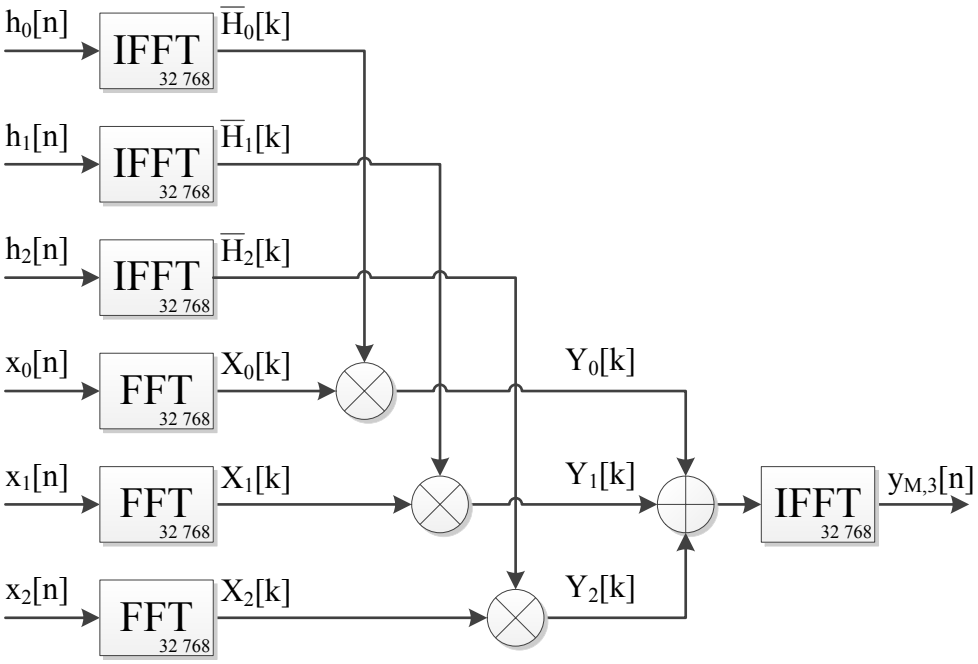

Figure 10: Implementation of the proposed algorithm for the E1 BOC(1,1) signal

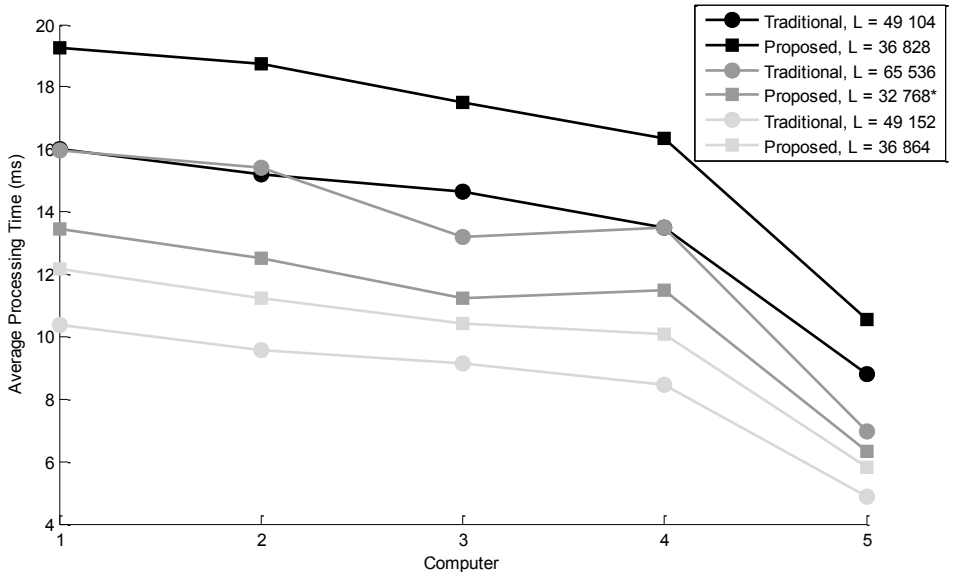

Figure 11: Average processing time of the algorithms on different computers for the Galileo E1 BOC(1,1) signal. (**use three sub-correlations*)

Signal	GPS L5		Galileo E5a		Galileo E5b		Galileo E1	
	I	Q	I	Q	I	Q	Q	Q
Carrier frequency (MHz)	1176.45		1176.45		1207.14		1575.42	
Primary code chipping rate (Mchip/s)	10.23		10.23		10.23		1.023	
Primary code length (chip)	10 230		10 230		10 230		4092	
Secondary code chipping rate (chip/s)	1000		1000		1000		250	
Secondary code length (chip)	10	20	20	100	4	100	-	25
Data rate (bit/s)	100	-	50	-	250	-	250	-

Table 1: Properties of the GPS L5 and Galileo E5a, E5b and E1 signals

l	5	7	9	11	13	15	17	19
$L = 2^l$	32	128	512	2048	8192	32 768	131 072	524 288
N	44	172	684	2732	10 924	43 692	174 764	699 052

Table 2: Possible correlation length (N) and radix-2 FFT length (L) of the proposed algorithm with two sub-correlations for l odd.

l	6	8	10	12	14	16	18	20
$L = 2^l$	64	256	1024	4096	16 384	65 536	262 144	1 048 576
N	84	340	1364	5460	21 844	87 380	349 524	1 398 100

Table 3: Possible correlation length (N) and radix-2 FFT length (L) of the proposed algorithm with two sub-correlations for l even.

Sampling frequency (MHz)	Range (MHz)	FFT length for the traditional algorithm	FFT length for the proposed algorithm
$1.023 - 1.024$	0.001	2048	2048
$1.025 - 1.366$	0.341	4096	2048
$1.367 - 2.048$	0.681	4096	4096
$2.049 - 2.730$	0.681	8192	4096
$2.731 - 4.096$	1.365	8192	8192
$4.097 - 5.462$	1.365	16 384	8192
$5.463 - 8.192$	2.729	16 384	16 384
$8.193 - 10.922$	2.729	32 768	16 384
$10.923 - 16.384$	5.461	32 768	32 768
$16.385 - 21.846$	5.461	65 536	32 768
$21.847 - 32.768$	10.921	65 536	65 536
$32.769 - 43.690$	10.921	131 072	65 536

Table 4: Radix-2 FFT length for the traditional and proposed algorithms with two sub-correlations in function of the sampling frequency considering a 1-ms code. For a 4-ms code, multiply the actual sampling frequency by 4.

l	7	8	9	10	11	12	13
$L = 2^l$	128	256	512	1024	2048	4096	8192
N	192	384	768	1536	3072	6144	12 288

l	14	15	16	17	18	19	20
$L = 2^l$	16 384	32 768	65 536	131 072	262 144	524 288	1 048 576
N	24 576	49 152	98 304	196 608	393 216	786 432	1 572 864

Table 5: Possible correlation length (N) and radix-2 FFT length (L) of the proposed algorithm with three sub-correlations.

Algorithm	Minimum FFT length	Minimum FFT length with small prime factors	Power of two FFT length
Traditional	$N = 40\,920$ $L = 40\,920$ $f_s = 20.46$	$N = 41\,472$ $L = 41\,472$ $f_s \in [20.46 - 20.736]$	$N = 65\,536$ $L = 65\,536$ $f_s \in [20.46 - 32.768]$
Proposed	$N = 40\,920$ $L = 30\,690$ $f_s = 20.46$	$N = 41\,472$ $L = 31\,104$ $f_s \in [20.46 - 20.736]$	$N = 43\,692$ $L = 32\,768$ $f_s \in [20.46 - 21.846]$

Table 6: Correlation length (N), FFT length (L) and sampling frequency (f_s, in MHz) for the acquisition of the GPS L5, Galileo E5a and E5b signals

Algorithm	Logic usage (ALUT)	Memory usage (M9K)	Multipliers usage (DSP element)
Traditional	22 881	3648	76
Proposed	36 006	3040	128
Proposed with multiplexing	21 618	1952	76

Table 7: FPGA resources for the GPS L5, Galileo E5a and E5b signals with the traditional ($N = L = 65\,536$) and proposed ($N = 43\,692$ and $L = 32\,768$) algorithms.

Algorithm	Minimum FFT length	Minimum FFT length with small prime factors	Power of two FFT length
Traditional	$N = 49\,104$	$N = 49\,152$	$N = 65\,536$
	$L = 49\,104$	$L = 49\,152$	$L = 65\,536$
	$f_s = 6.138$	$f_s \in [6.138 - 6.144]$	$f_s \in [6.138 - 8.192]$
Proposed	$N = 49\,104$	$N = 49\,152$	$N = 49\,152$*
	$L = 36\,828$	$L = 36\,864$	$L = 32\,768$*
	$f_s = 6.138$	$f_s \in [6.138 - 6.144]$	$f_s \in [6.138 - 6.144]$

Table 8: Correlation length (N), FFT length (L) and sampling frequency (f_s, in MHz) for the acquisition of the E1 BOC(1,1) signal. (**use three sub-correlations*)

Algorithm	Logic usage (ALUT)	Memory usage (M9K)	Multipliers usage (DSP element)
Traditional	22 881	3648	76
Proposed	50 430	4256	180

Table 9: FPGA resources for the traditional ($N = L = 65\,536$) and proposed algorithms ($N = 49\,152$ and $L = 32\,768$ with three sub-correlations) for the E1 BOC(1,1) signal