Implementation Strategies for a Software-Compensated FFT-based Generic Acquisition Architecture with Minimal FPGA Resources

MARC-ANTOINE FORTIN, FRANCIS BOURDEAU and RENÉ JR. LANDRY
École de Technologie Supérieure (Éts), Montréal, Canada

Received December 2014; Revised February 2015;

ABSTRACT: A generic 2048-point FFT acquisition architecture is proposed to address L1 civil signals from all four GNSS constellations. After emphasizing hardware design criteria and their resulting design limitations, software compensation approaches are compared. A detailed validation methodology, involving a successive 1000-step Monte-Carlo study, was defined to optimally configure the acquisition channel, with new metrics to establish formal signal detection. The integration thereof results in a novel, minimalistic, yet generic, acquisition channel implementation, as well as a thorough validation method. Execution time of one acquisition iteration is approximately 5 ms, in line with VHDL simulations and foreseen channel management overhead. Coarse/fine search increments and thresholds are based on extensive experimentation. A 41 dB-Hz acquisition sensitivity threshold was established to achieve >95% detection rates for 1 ms integrations, while 15 ms non-coherent integrations are required for signal strengths down to 37 dB-Hz. These thresholds account for known implementation losses. Copyright © 2015 Institute of Navigation

1 INTRODUCTION

The acquisition of a Global Navigation Satellite System (GNSS) signal consists of searching for a given satellite signal buried in noise. For a receiver to synchronize itself onto such a Pseudo-Random Noise (PRN) signal, it must approximate the spreading code time offset due to its propagation, as well as its Doppler frequency shift induced by the satellite Line Of Sight (LOS) variations, i.e., the relative movement of the satellite with respect to the receiver. During cold start, without a priori knowledge about a given satellite orbit, this two-dimensional (2-D) search may become a heavier processing burden as the code length increases. Indeed, one typically searches the best code delay with a half-chip resolution (for BPSK modulation) and parses a low user dynamic resulting in a ±5 kHz Doppler range with steps smaller than 2/(3TI) [1], where TI is the coherent integration time. Valid almanacs could lead to warm starts targeting satellites known to be visible with a Doppler coarse estimate, while valid ephemerides could even further narrow down the search space by providing expected Doppler and code offset estimations, provided the time, user location and speed are approximately known. This useful knowledge may be obtained by either tracking other satellites or through an external communication link such as in Assisted-GPS (A-GPS) [2].

Acquisition algorithms are typically based on signal autocorrelation properties. In satellite navigation, spreading codes are periodically repeated pseudo-random sequences allowing for multiple satellite signals access. Hence, spreading codes are truly deterministic, although exhibiting random signal properties to a GNSS receiver. Nevertheless, the correlation process indicates how well the received signal is aligned (in both time and frequency) with its locally generated replica. In the case of code offsets greater than one chip, the correlation product tends towards 0, whereas a Binary Phase Shift Keying (BPSK) modulated code alignment within ±1 chip would be located somewhere on the 2-chip wide isosceles triangular shaped correlation peak. More precisely, achieving a correlation normalized threshold of say 1/2 implies a partial (i.e., sub-chip) alignment of the input signal
with its replica. Moreover, the targeted carrier frequency (translated down to Intermediate Frequency or IF) must also be matched, at least coarsely, in order to prevent undesired signal losses (which could compromise the acquisition process altogether), as dictated by the signal linear correlation amplitude attenuation factor $A$ (with the frequency offset $\Delta f$ between input signal and replica):

$$A = \frac{\sin(\pi \Delta f \cdot T_1)}{(\pi \Delta f \cdot T_1)} = \sin(\pi \Delta f \cdot T_1) \quad (1.1)$$

This potential loss reaches a peak when the incoming signal is exactly in between two Doppler bins, resulting in a maximum amplitude attenuation $A = \sin\left(\frac{2\pi n}{T_1} \right) = \sin(\xi) = 0.827 \ (–1.647 \text{ dB})$, which is not nearly as bad as the complete signal attenuation when the argument of the $\sin$ function tends towards an integer multiple of $\pi$ ($\sin(\pi \cdot x) = 0, \forall x \in \mathbb{Z}$). This potential pitfall applies equally to all acquisition methods.

Acquisition sensitivity may be defined as the post-correlation Signal to Noise Ratio (SNR) – or equivalently the Carrier power to Noise Density (C/\(N_0\)) defined as the SNR normalized in a bandwidth (BW) of 1 Hz – threshold required for successful signal acquisition. In hostile environments and indoors, typically characterized by harmful perturbations and attenuated, distorted signals, this acquisition threshold is not as easily achieved as in clear open sky conditions. In the case of post-correlation SNR, one is interested in the correlation of the signal of interest $s_x$ with the corresponding local replica $r_x$ compared against an independent replica $r_y$.

$$SNR = \frac{\sum_{t=0}^{T_1} s_x[n] \cdot r_x[n]}{\sum_{t=0}^{T_1} s_x[n] \cdot r_y[n]} \quad (1.2)$$

$$\frac{C}{N_0} = \frac{SNR}{BW} \quad (1.3)$$

Acquisition (or re-acquisition of lost signals) is a crucial step in satellite navigation. More specifically, indoor navigation strongly relies on successive acquisitions as more robust tracking loops simply cannot work consistently in these signal-challenging conditions.

To improve acquisition sensitivity, longer integration periods can be used to accumulate incoming signal power. Coherent integration time is limited by the navigation bit length (or symbol length for encoded messages), unless properly wiped-off through external aiding and Doppler change, as well as by an unresolved overlaid secondary chip period, if applicable. Non-coherent integration overcomes these limitations at the cost of greater noise, known as squaring losses [3]. Furthermore, some signals now have the advantage of also being composed of a data-less pilot component, which may offer a signal strength gain and allow for longer coherent integration times. A combined data and pilot channels acquisition scheme would allow harvesting all the available signal power, thus achieving better acquisition performance, although with greater computational efforts and larger resource costs.

In all acquisition modes (cold to very hot starts with external aiding), finding the right code offset remains a time-consuming task. With the advent of longer codes, the expected Mean Time To Acquire (MTTA), without any a priori knowledge, becomes even greater. Considering the multitude of GNSS signal standards emerging and accounting for the previous considerations, a generic acquisition approach would appear to be of interest.

### 1.1 Acquisition Objectives

The blooming mobile device market is being driven towards minimal power/resource configurations, favoring more computationally efficient approaches. In this context, a minimalistic GNSS parallel acquisition channel design should accommodate all four GNSS constellations (i.e., GPS, GLONASS, Galileo, and BeiDou) while consuming minimal power. To alleviate the first requirement, at least one signal type per constellation must be successfully acquired. Indeed, the information obtained from one signal may then be extrapolated to other signals from the same satellite, minimizing their sequential acquisition effort. This knowledge is even more valuable than almanacs or ephemerides. Currently, the simplest and fastest signals to acquire are GPS L1 C/A, GLONASS L1OF, Galileo E1B/C, and BeiDou B1-I (characterized in Table 1–I), which also have the advantage of being on the same

<table>
<thead>
<tr>
<th>GNSS Constellation</th>
<th>RF carrier Frequency [MHz]</th>
<th>Chipping rate [Mchip/s]</th>
<th>Length [chip]</th>
<th>Period [ms]</th>
<th>Data bit length [ms]</th>
<th>Modulation type</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS L1 C/A</td>
<td>1575.420</td>
<td>1.023</td>
<td>1023</td>
<td>1</td>
<td>20</td>
<td>BPSK</td>
</tr>
<tr>
<td>GLONASS L1</td>
<td>1602.000 + 0.5625[-7, 6]</td>
<td>0.511</td>
<td>511</td>
<td>1</td>
<td>10</td>
<td>BPSK</td>
</tr>
<tr>
<td>Galileo E1-B/C</td>
<td>1575.420 ±1.023</td>
<td>1.023</td>
<td>4092</td>
<td>4</td>
<td>4</td>
<td>MBOC(6,1,11,±)</td>
</tr>
<tr>
<td>BeiDou B1-I</td>
<td>1561.098</td>
<td>2.046</td>
<td>2046</td>
<td>1</td>
<td>1</td>
<td>BPSK</td>
</tr>
</tbody>
</table>
frequency band (provided the front-end Local Oscillator frequency can be adjusted to adapt to a slightly different RF carrier for GLONASS at 1602 MHz and BeiDou at 1561.098 MHz), thus conveniently simplifying the RF front-end required for demonstrating its implementation.

In GNSS signals, unsynchronized secondary code and unknown binary message (encoded whenever applicable) introduce phase inversions, thus limiting the coherent integration time. Even worse, the destructive effect reaches a maximum for a correlation straddling two different symbols; the correlation peak can be completely eluded if a phase inversion were to occur at 50% of the integration window. Luckily, the navigation message rates are low for GPS L1 C/A (50 Hz) and GLONASS L1 (100 Hz). However, this phase inversion would have a more frequent impact in the case of Galileo E1 as both secondary code (in the case of E1-C) and navigation symbol (in the case of E1-B) duration match their primary code duration. Nevertheless, the 50 Hz BeiDou B1-I navigation message is laid over a 1 kHz secondary code, making it the most challenging signal to acquire in terms of phase inversion probabilities.

Signal processing of these four signals by the simplest acquisition approach creates several challenges, including:

1. Different code generators;
2. Primary code lengths from 511 to 4092 chips with rates varying from 0.511 to 2.046 Mchip/s;
3. Secondary code or navigation bits introducing phase inversions with 1–20 ms intervals;
4. Code duration varying between 1–4 ms;
5. Multiple modulations including BPSK, BOC, and even MBOC;
6. RF band center frequencies ranging from 1561.098 to 1602.000 MHz; and
7. GLONASS FDMA plan requires the carrier wipe-off component to span over [−7, +6] · 562.5 kHz.

Addressing these sub-objectives, a parallel acquisition architecture may be optimized in terms of resources, thus meeting the reduced resources (i.e., the second) requirement. After reviewing a few fundamental concepts and further describing the signals to be acquired, this paper details the proposed parallel GNSS acquisition channel and justifies the choices that have led to its design. Iterative software algorithms were developed to alleviate the limitations introduced by channel hardware reductions. New metrics are introduced to support the analysis of several trials in different conditions. Finally, the paper presents concluding remarks on the general performance of the proposed method, called HEAD-start for Highly Efficient Acquisition Degree (−start).

2 PARALLEL ACQUISITION ARCHITECTURES

Acquisition algorithms may be divided into two categories: sequential and parallel algorithms. The proposed algorithm is based on a parallel architecture, the literature thereof being outlined below, where it is assumed that \( T_I = 1 \text{ ms with 2 samples per chip} \).

Van Nee [9] has established that parallel algorithms based on Fast Fourier Transform (FFT) provide faster results through a reduced computation complexity \( O(N \cdot \log(N)) \) versus their sequential counterparts \( O(N^2) \), where \( N \) is the correlated sequences length. There are two main types of 1-D parallel acquisition: Parallel Frequency and Parallel Code, both being repeated for each of their un-parallelized dimension; 2-D parallel acquisition combines both these 1-D acquisition types.

The parallel search in the frequency domain algorithm relies on the fact that perfectly wiping-off the code results in a sine wave, which can then be identified by a strong frequency coefficient at the FFT output [10].

\[
D_n = \text{FFT}\left\{ (s[_{baseband}]_n \cdot c[n])_{decimated} \right\} \quad \text{(2.1)}
\]

where:
- \( D_n \) is the Doppler bin estimate for a particular code alignment
- I/FFT are the Inverse/Fast Fourier Transform operator
- \( P_s \) is the sampling period
- \( s[n] \) is the received signal with \( s[n] = s(n \cdot P_s) \) and \( c[n] \) is the code replica.

The associated frequency resolution for ±5 kHz frequency search and 1 ms integration time impose:

\[
\text{Res}_D \leq \frac{B_{\text{FFT}}}{N_{\text{FFT}}} \Rightarrow 667 \leq \frac{2 \cdot (5 \cdot 10^3)}{N_{\text{FFT}}} \Rightarrow N_{\text{FFT}} \geq 15 \quad \text{(2.2)}
\]

where:
- \( \text{Res}_D \) is the Doppler frequency resolution
- \( B_{\text{FFT}} \) is the searched signal Doppler frequency span, and
- \( N_{\text{FFT}} \) is the number of FFT points, typically a power of 2.

An increased coherent integration time \( T_I \) requires narrower Doppler bin spacing and therefore greater FFT lengths, as per (2.2). To avoid computation of a large number of different code offset iterations, Van Nee proposes to rather apply parallelism on the code search. To do so, the parallel search in the code domain algorithm must resolve the following equations [9]:

\[
C[f] = \text{FFT}(c[n])
\]

\[
S[f] = \text{FFT}(s[n])
\]

\[
R_{cs}[n] = \text{IFFT}(C^*[f] \cdot S[f])
\]

where:
- \( S[f] \) is the received signal Fourier coefficients
- \( C[f] \) is the replica Fourier coefficients
- \( \ast \) is the complex conjugate operator, and
- \( R_{cs} \) is the correlation between the input signal \( s \) and corresponding code replica \( c \).
Here, the frequency-domain correlation must be computed for each tested Doppler bin. One way to remove a frequency offset consists of multiplying the time-domain signal by a complex exponential \( s'[n] = s[n] \cdot e^{-j\Delta f_D n} \). Alternately, one can multiply the real code time-domain replica by a complex exponential to add a corresponding Doppler frequency offset \( c'[n] = c[n] \cdot e^{+j\Delta f_D n} \). A computationally optimized approach consists of circularly shifting the frequency-domain replica (i.e., \( C(f) \)) to reproduce discrete approximations of different Doppler shifts, thus greatly reducing the number of FFT computations at the cost of signal sensitivity losses due to lower frequency resolution. For each Doppler alignment tested, an Inverse FFT (IFFT) is performed on the complex product, producing a time-domain cross-correlation product at every sample. Further details, such as signal correlation and Doppler side lobes as well as artefacts resulting from zero-padding may be found in [1, 11].

### 2.1 Parallel Acquisition Architectures Comparison

Compared to the parallel frequency algorithm, the parallel code algorithm requires fewer operations, as seen in Table 2–I. Indeed, considering the GPS L1 signal with two samples per chip, the former requires 1 FFT and 2046 multiplications for each tested chip offset. In its most complex implementation, the latter requires 1 code FFT and 16 signal FFTs, 15 of which are prior compensated in frequency (2046 complex multiplications), followed by 2048 complex multiplications and an IFFT for each of the 16 iterations.

If the baseband real code replica were multiplied by a real exponential to induce a Doppler frequency shift, some computations would be saved at the cost of reduced accuracy. Nevertheless, circular shifting the frequency coefficients of the replica comes at no hardware cost and saves 15 FFT computations, achieving the best resources per computation ratio.

Such an injected Doppler frequency offset may be quantified as:

\[
Af_D = \frac{f_s}{N_{FFT}} \cdot N_{shift}
\]

where:

\( f_s \text{avg} \) is the sampling frequency of the signal at the input of the FFT module.

Thus, the parallel code phase search appears to be the most computationally efficient acquisition method, but requires more resources because of its greater algorithm complexity, where the sampling frequency is set at twice the chipping rate. However, if the three Fast Fourier transforms (two direct and one inverse) are done sequentially through a single I/FFT module, the parallel code algorithm becomes even more interesting, as it allows for keeping the iterations count equal to the Doppler bins, no matter how large the spreading codes get, while also keeping the resources low through sharing. In this paper, the parallel code (i.e., 1-D) with circular shifting approach is chosen and further analyzed, after an outline of similar work in literature.

### 2.2 Other Parallel Frequency Domain Acquisition Approaches

FFT-based acquisition was first introduced by Van Nee and Coenen [9]. Later, Double Block Zero Padding (DBZP), also known as Circular Correlation by Partition and Zero Padding, was proposed by Tsui, as summarized in [1]. Recently, a lot of efforts were invested in a weak signals acquisition trend. Although this paper favors low complexity over high sensitivity, their work (often based on longer integration time) is outlined herein.

In [11], Ziedan proposed the Circular Correlation with Multiple Data Bits (CCMDB) and the Modified Double Block Zero Padding (MDBZP), which was further developed in [12]. Another optimization of the DBZP method is available in [13], where

<table>
<thead>
<tr>
<th>Acquisition method</th>
<th>Addition</th>
<th>Real multiplication</th>
<th>16-point FFT</th>
<th>2048-point FFT</th>
<th>2048-point IFFT</th>
<th>Computational order ( O(\cdot) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential (i.e., traditional correlator)</td>
<td>16:2046-2046</td>
<td>16:2046-2046</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16:2046^2</td>
</tr>
<tr>
<td>Parallel Frequency</td>
<td>2046-2046</td>
<td>2046-2046</td>
<td>2046</td>
<td>0</td>
<td>0</td>
<td>2046 [2046 + 16 \log_2(16)]</td>
</tr>
<tr>
<td>Parallel Code with complex exponential (Doppler injection &amp; FFT product)</td>
<td>2:16-2048</td>
<td>2:15-2046 + 4:16-2048</td>
<td>0</td>
<td>15 + 2</td>
<td>16</td>
<td>33 [2048-log_2(2048)] + 30:2046 + 64:2048</td>
</tr>
<tr>
<td>Parallel Code with real sine wave (Doppler injection &amp; FFT product)</td>
<td>2:16-2048</td>
<td>15-2046 + 4:16-2048</td>
<td>0</td>
<td>15 + 2</td>
<td>16</td>
<td>33 [2048-log_2(2048)] + 15:2046 + 64:2048</td>
</tr>
<tr>
<td>Parallel Code with circular shifting (Doppler injection &amp; FFT product)</td>
<td>2:16-2048</td>
<td>4:16-2048</td>
<td>0</td>
<td>2</td>
<td>16</td>
<td>18 [2048-log_2(2048)] + 64:2048</td>
</tr>
</tbody>
</table>
Complete Correlation Results (CCR), rather than Partial ones (PCR), allow a 1.3 dB processing gain. Mollaiyan proposed a Pre-Correlation Accumulation (PCA) method, also derived from DBZP [14].

Other high sensitivity acquisition approaches are referred to in [15], although these are beyond the scope of this paper. Instead, a theoretical investigation of the proposed architecture performance is highlighted in Section 6 to serve as benchmark against its experimental results.

3 ACQUISITION CHANNEL IMPLEMENTATION DETAILS

The acquisition channel presented in this paper targets a Xilinx Virtex IV FPGA implementation. The synthesis tool provides built-in and configurable entities such as RAM, Digital Signal Processing on 48-bit (DSP48) slices, and IFFT cores. Furthermore, a GNSS receiver has already been implemented [16–19], which cannot afford enough dedicated resources for all available GNSS civil signals. The system is composed of a computer host, the above mentioned FPGA in which a programmable processor is synthesized. In the current implementation, the same 60 MHz sampling frequency $f_s = 1/P_s$ and a 15 MHz Intermediate Frequency (IF) are used from the existing GNSS receiver development platform, regardless of which GNSS signal is being processed, hence allowing for a generic architecture.

The 60 Msample/s sampling frequency leads to an unmanageable 65,536 samples/ms FFT radix-2 sizes. Hence, the signal first needs to be down-sampled. To determine an appropriate FFT length (considering GPS L1 signal in a first attempt), many factors need to be considered, including the Nyquist sampling theorem to avoid aliasing [10] while still harvesting to be considered, including the Nyquist sampling frequency at which the samples to allow a radix-2 FFT module to one code period per iteration. Zeros are added after Period Zero-Padding (SPZP) algorithm processes only anywhere in the code period, which may not be introduced of such a bundle of zeros may occur anywhere in the code period, which may not be introduced of such a bundle of zeros may occur anywhere in the code period, which may not be introduced of such a bundle of zeros may occur anywhere in the code period, which may not be introduced of such a bundle of zeros may occur anywhere in the code period, which may not be introduced of such a bundle of zeros may occur anywhere in the code period, which may not be introduced of such a bundle of zeros may occur anywhere in the code period, which may not be introduced of such a bundle of zeros may occur anywhere in the code period, which may not be introduced of such a bundle of zeros may occur anywhere in the code period, which may not be introduced.

Indeed, with 2 Msample/s, the BeiDou B1-I signal could compromise the acquisition in some cases. As seen later with fine increments in section § 4, the 6 dB worst case (2.5 dB average) loss associated with 1 sample/chip can be mitigated.

The 2000 averaged samples (for decimation from 60 to 2 Msample/s) must then be padded with $N_{\text{zeros}} = 48$ zeros, although this comes at a cost. Indeed, a Single Period Zero-Padding (SPZP) algorithm processes only one code period per iteration. Zeros are added after the samples to allow a radix-2 FFT module to efficiently process power of two lengths, i.e., from 2000 to 2048. In the case of the incoming signal, the introduction of such a bundle of zeros may occur anywhere in the code period, which may not be aligned with the integration window; in the case of the replica code, it always occurs at the end of the code sequence. This process causes two partial correlation peaks to occur rather than one. For example, a 50% offset between the zero-padded signal and replica sequences generates two peaks of about half the nominal amplitude, each of which are found at

$$N_{\text{FFT}}$$

is the FFT length.

With the objective of minimizing hardware resources, the constraint on the Doppler frequency resolution is relaxed to $1 \text{kHz}$ rather than $500 < 667 \text{Hz}$ in (3.2), reducing the first constraint to $N_{\text{FFT}} \geq 2000$. The FFT length is selected to be $N_{\text{FFT}} = 2048$ points (i.e., the next power of 2). (3.2) results in:

$$R_{\text{SD}} = \frac{BW_{\text{FFT}}}{N_{\text{FFT}}} \geq 2 \times 10^6 \Rightarrow N_{\text{FFT}} \leq 976.5625 \text{Hz}$$

$$R_{\text{SC}} = \frac{N_{\text{FFT}}/T_{\text{ms}}}{N_{\text{FFT}} - N_{\text{zeros}}} \geq 1023 \Rightarrow 0.5115 \text{chip}$$

(3.3)

where:

$R_{\text{SD}}$ is the Doppler frequency resolution

$R_{\text{SC}}$ is the chip resolution

$BW_{\text{FFT}}$ is the bandwidth at the FFT input

$N$ is the spreading code length in chips, that of GPS L1 C/A being used in (3.2), and

$N_{\text{FFT}}$ is the FFT length.

Note that considering square chips with chipping rate $f_c$, a single sample per chip would allow for complete code reconstruction, but with lower expected performances. Then, the Doppler frequency and code resolution conditions must be considered. Assuming a 1023 chip long code sampled at 2 Msample/s during 1 ms, slightly below what is prescribed in (3.1), we get:

$$R_{\text{SD}} \leq \frac{BW_{\text{FFT}}}{N_{\text{FFT}}} \Rightarrow 500 \leq \frac{2 \times 10^6}{N_{\text{FFT}}} \Rightarrow N_{\text{FFT}} \geq 4000$$

(3.2)

where:

$R_{\text{SD}}$ is the Doppler frequency resolution

$R_{\text{SC}}$ is the chip resolution

$BW_{\text{FFT}}$ is the bandwidth at the FFT input

$N$ is the spreading code length in chips, that of GPS L1 C/A being used in (3.2), and

$N_{\text{FFT}}$ is the FFT length.
suffer at most a 6 dB loss, i.e., a partial correlation over half its samples. Although the number of the zeros may be negligible, the partial peak they cause is quite significant in the case of SPZP as well as the navigation bit inversion issue. Hence, these partial peaks should either be detected and combined, or simply avoided all together (cf. section §4).

An alternative to this limitation would be the DPZP, where the zeros appended at the end of two code periods always leave one full code unaltered; the other period most probably being split, as with SPZP [1], as in Figure 3–II. This analysis simplification comes at the cost of increased hardware resources and execution time as twice the samples are required, given the same chip resolution is to be achieved. As depicted in Figure 3–III, the proposed parallel code acquisition channel is composed of several modules, which are detailed in the following paragraphs.

3.1 Input Signal Quantization Module

Signal quantization is another concern in Very Large Scale Integration (VLSI) Hardware Design Language (VHDL) design: as mathematical operations proceed, the number of bits required to prevent signal loss increases. In the current case, the analog input signal (at the RF front-ends output) is conditioned by an Automatic Gain Controller (AGC) and sampled on 14 real bits, of which only 4 signed bits are retained for digital signal processing.

Given the established resources constraint, integer versus floating point operations must be decided upon. While the latter choice is expected to provide better performance, it also requires the implementation of a whole floating point Arithmetic Logic Unit (ALU). In order to save resources and time, integer operations were implemented, inducing additional noise due to truncation. These ratios are further detailed in Section 3.6.

3.2 Carrier Wipe-off Module

In the acquisition channel, a down-conversion of the incoming signal from IF to baseband is first performed. The carrier wipe-off module sine and cosine waves’ frequency configuration must comply with all GNSS signals. Hence, the carrier Numerically Controlled Oscillator (NCO) must cover the frequency range defined by the union of all the considered signals, as detailed in Table 1–I. For simplicity, a full wave period is synthesized from 64 samples, each defined on 4 signed bits balanced around 0, i.e., ±7 linear amplitude levels. This standard sin/cos takes the input signal down to baseband, now quantized on 7 signed bits.

The Galileo E1 signal is CBOC modulated, where a second 6/1 square sub-carrier is introduced with one tenth of the signal power of the first square sub-carrier, rated at 1.023 MHz. If Single Side Lobe (SSL) [20] were used to process either one of two main lobes of the BOC(1,1) modulation, acquisition would suffer a 3 dB loss. Also, considering either equally powered data or pilot component (another 3 dB loss), neglecting the second sub-carrier implies an additional loss of $10 \log_{10} (10/11) = -0.4$ dB. The Galileo signal power admitted for a single component acquisition would thus be $-160.0 - 3.4 = -163.4$ dBW, which is
almost 5 dB lower than GPS L1 C/A, i.e., –158.5 dBW. This alone, explains why poorer performances are to be expected for the Galileo signal with the proposed acquisition approach, requiring a non-coherent integration over a few milliseconds (cf. section § 3.6.1).

### 3.3 Spreading Code Generation Module

In order to avoid multiplying resources for the generation of each GNSS constellation code, a common “memory code” approach is used, where the pre-computed spreading code of interest is loaded into local memory. To account for their specific chipping rates, a Delay Lock Loop (DLL) command drives a configurable code NCO, as per Table 1–I.

### 3.4 Averaging Module

In order to implement a reduced size I/FFT core, the baseband signal \( c \) is averaged over 30 consecutive samples \( c_{\text{avg}} \), hence decimated from 60 to 2 Msample/s, while keeping the samples on 7 bits as inputs to the FFT module, to minimize its implementation size. Averaging acts as a low-pass filter prior to decimation, thus avoiding aliasing. For performance purposes, the division by 30 is approximated by a binary shift of five positions (representing a division by 32), which raises an implementation problem: i.e., 30 consecutive positive 1-bit samples are summed and shifted, it results in 0. To avoid this, and to maximize the frequency-domain complex product, the spreading code is scaled by 5 through a look-up table, translating the memory code (cf. section § 3.3) from 0/1 to ±5, without multiplication.

\[
\begin{align*}
    c_{\text{avg}} &= \frac{\sum_{k=1}^{30} c(k)}{32} = \left\lfloor \frac{30}{32} \right\rfloor = 0 \\
    c_{\text{avg}} &= \frac{\sum_{k=1}^{30} 5 \cdot c(k)}{32} = \left\lfloor \frac{150}{32} \right\rfloor = 4
\end{align*}
\]
A slight loss is still suffered as the approximation results in 4 rather than 5. The signal is further altered when the averaging window straddles 2 consecutive bits.

### 3.5 Direct and Inverse FFT Module

One of the main design choices to be made while synthesizing Direct and Inverse I/FFT cores relates to bus sizes. In the proposed acquisition channel, the width of the signal after averaging still stands on 7 signed bit samples, which should correspond to the FFT input width. However, the input of the IFFT (which is actually the same instance of the FFT core) must also account for the result of a 15-bit complex multiplication of two complex samples. The FFT inputs and outputs were thus fixed to 16 bits, a subset of which are meaningful for the FFT computations, i.e., the 8 MSB of the FFT output are saved in RAM for subsequent computations.

Another important I/FFT core design parameter is its operation mode: burst vs. pipelined. The pipelined version represents less management, as it requires a second clock domain to synchronize the signal inputs and outputs at a reduced rate (e.g., 2 MHz). The FFT takes much longer to compute as its clock domain is now 30 times slower. On the other hand, burst mode uses the 60 MHz clock domain to load the FFT samples, unload the FFT coefficients, and compute the FFT. The averaging module produces samples at 2 MHz, necessitating a buffer RAM insertion, not shown in Figure 3–III. Four additional RAM blocks are needed to store the FFT complex results for both the replica and the input signals, of which only the 8 MSB are kept. This option allows non-coherent summing of several acquisition results because of its reduced computation time; the pipelined mode computation takes longer than a code period to execute.

Because of the Xilinx IP core characteristics, a 16-bit 2048-point I/FFT module is thus synthesized in burst mode. Since a single I/FFT block is used, the parallel code search algorithm must be divided into three sequential phases to allow for sharing in time for the FFT module for the following needs: the non-frequent code FFT, the repeated signal FFT, and the ongoing IFFT.

### 3.6 Peak Detection and Noise Computation Module

The last step in the proposed algorithm is the peak detection, which is split into four clocked operations performed by a single DSP48 [21]. First, it computes the IFFT samples squared magnitude, avoiding the computationally expensive square root operation. Each of the real and imaginary squaring operations takes one clock cycle. Then, they may be non-coherently accumulated with the previous result at the same chip offset. Note that the proposed solution does not compensate for code scaling due to Doppler, because of its associated complexity impact. On the last clock cycle, the current complex power is accumulated into the total noise measurement. This noise accumulation consists of adding the squared amplitude of every cell, including the correlation peak. By detecting the maximum cell value over the 2D search grid, the Peak to Noise Ratio (PNR) can be obtained.

\[
\text{Total Power} = \sum_{d=1}^{N_{\text{Doppler}}} \sum_{c=1}^{N_{\text{FFT}}} \left( I_{d,c}^2 + Q_{d,c}^2 \right)
\]

\[
\text{PNR} = \frac{(I^2 + Q^2)_{\text{first peak}}}{\text{Total Power} - (I^2 + Q^2)_{\text{first peak}} - (I^2 + Q^2)_{\text{second peak}}}^{N_{\text{Doppler}}-1}
\]

Here, the noise floor density per cell is estimated as the total cells’ accumulated power as per (3.5), from which the first and second greatest peaks are removed, before being divided by the total cells’ number minus the two removed peaks. This first metric could be seen as a Threshold Comparison (TC) with an adaptive threshold (Hybrid Search) [22], but considering both first and second peaks computed only once, i.e., without cell subsets.

According to the Neyman-Person lemma, TC would be optimal in maximizing the detection probability for a given False Alarm (FA) probability if the noise cells variance were known [22]. In the proposed acquisition channel, this variance is not known, and a constant threshold is used. It is yet assumed the PNR should approach the generalized likelihood-ratio test performances.

Another performance metric, a First to Second Squared Peak Ratio (FSSPR) is computed. It is used as an acquisition quality metric to further support (3.6), rather than a Ratio Detection (RD) threshold [22] in itself with lower detection performances. The second peak search neglects cells adjacent to that of the first peak.

\[
\text{FSSPR} = \frac{(I^2 + Q^2)_{\text{first peak}}}{(I^2 + Q^2)_{\text{second peak}}}
\]

Indeed, if multiple peaks have the same value, it means that a true correlation peak does not stand out in the search. In Figure 3–IV (left), both first and second peaks are similar in (squared) magnitude because the true Doppler is closer to –488.28 Hz, i.e., in between the hardware Doppler bins at 0 and –976.56 Hz. In Figure 3–IV (right), a 488.28 Hz offset is applied; the result really being –976.56 + 488.28 = –488.28 Hz, as expected.

In the case of non-coherent integrations, the IPFFT operation and the Doppler search must be computed within 1 ms to allow time to compute the FFT of the
next incoming signal epoch. Experimentation indicates that only two IFFTs – and hence two different Doppler bins – can be processed during 1 ms with extra hardware. Considering the resources constraint, the Doppler search is limited to a single bin when multiple (partial or non-coherent) integrations are required, as further analyzed in section § 3.6.1. Hence, only the following three scenarios are made available in the acquisition channel:

1. All hardware Doppler bins are covered within a single iteration (i.e., computation time < 1 ms for non-coherent integration accumulation $M = 1$)
2. Only one Doppler bin is accumulated over multiple non-coherent integrations $M > 1$ (i.e., 1 Doppler/ms)
3. Only one Doppler bin is accumulated with partial acquisition (cf. section § 3.6.1)

### 3.6.1 Partial Code Acquisition Methods

The basic integration time is 1 ms, which matches GPS, GLONASS, and BeiDou L1 civil signals spreading code periods, but only a quarter of the 4 ms long Galileo E1B/C primary codes. Different approaches have been investigated to consider partial code acquisition. For efficiency purposes, a reduced ±4 kHz Doppler span is used, resulting in only nine hardware Doppler bins. Also, the navigation message bit transitions do not need to be accounted for as 1 ms integrations are considered.

Initial Matlab simulations with recorded real signals have shown that correlating only a quarter of the Galileo E1B/C spreading code would not result in proper acquisition detection rates, especially with the SSL approach. Hence, four non-coherent accumulation methods are compared below and summarized. Alternatively, a maximum selection could have been used for methods 1XXX and 1111.

#### 3.6.1.1 Method 1234.

In a first attempt, four 1 ms of input signal are sequentially correlated with the four quarters of the replica code, as shown in Figure 3–V. Because of processing time constraints, the acquisition channel can only process 1 ms of signal with its results accumulated non-coherently. With an unknown code alignment, this process is executed four times, each with a different code quarter offset.

In Figure 3–V, there is no correlation result when the received signal is not at least partially aligned with its replica, for each code quarter. In a 4 ms window, each replica code quarter is equally offset from the received signal; hence, their non-coherent accumulation produces a main correlation peak. Furthermore, since the FFT must be computed for every ms of input signal, the IFFT and the correlation peak search computations must all be done simultaneously with the averaging of the next ms of input signal. Thus, only one Doppler bin may be searched at a time. To cover the full ±4 kHz span, this algorithm needs to process four
different code alignments for each of the nine different hardware Doppler bins, accumulating 4 ms of correlated signal (for every Doppler bin, at every code quarter alignment).

3.6.1.2 Method 1XXX. In the 1XXX method, a single quarter of code is correlated with the received signal over several periods, as in Figure 3–VI. Since only one code quarter is used in the correlation process, the resulting idle time (NOP) can be used to search through all nine hardware Doppler bins. The total execution time for this technique is expected to be nine times lower than that of the 1234 method. However, the correlation peak amplitude should be four times smaller.

3.6.1.3 Method 1X3X. Alternately correlating two different quarters of code with the received signal led to the 1X3X method. Using half of the code results in a correlation peak twice as low as that of the 1234 method, but twice as high as that of the 1XXX method. In addition, the reduction of idle time only allows for searching within four hardware Doppler bins, resulting in an execution time slightly more than two times longer than that of the 1XXX method, but shorter than the 1234 method.

3.6.1.4 Method 1111. Finally, in order to avoid idle states, the incoming signal is correlated over four quarters of a code period, each time with the same quarter of the replica code. Since only one code quarter matches each 4 ms window of input signal, the correlation peak amplitude of this technique should be as high as the amplitude observed in the 1XXX method, while the accumulated noise should be four times greater. The idle time absence limits the Doppler search to a single bin, resulting in an execution time comparable to that of the 1234 method.

3.6.2 Resource and Performance Comparison

The theoretical execution time has been extrapolated from VHDL simulations for the four different techniques and is presented in Table 3–II. Every algorithm has a setup time based on the computation of the FFT of the replica (for each code quarter used) and the averaging of the first ms of signal. Then, the processing time allows computing of the following: the input signal FFT, the IFFT of the frequency-domain complex multiplication results, and the correlation peak search. Note that this peak search is performed simultaneously with the IFFT of the next hardware Doppler bin or the averaging of the next ms of input signal, whichever applies. Finally, the search time includes the search through all nine 976.56 Hz hardware Doppler bins, for the four quarter code alignments. A negative value indicates that the last Doppler bins ended early, leaving an idle time before the next search was launched. Note that the reduction of the total Doppler span search to ± ~4 kHz (rather than ±5 kHz) greatly simplifies the computation complexity of the 1234 method.

These four methods also require different resources to store the FFT results of the different code quarters used in the correlation process. However, most RAM blocks and DSP48 slices are reused by all of them, as part of their common modules, where the common IFFT core is obtained from the Xilinx library (made up of slices, flip flops, and 4-input Look-Up Tables or LUT), and thus kept out of Table 3–III. The replica code 16 kbit RAM remains the same, as it already supports all civil GNSS codes, except for L2CL (not considered here). Each method computes the maximum achievable Doppler bins within their idle time, which requires a single DSP48 slice in all cases.

Their resources usage may be further compared, targeting the varying number of FFT results for replica code quarters to be stored in additional RAM blocks during the correlation process. Simulations were run over the same total duration, allowing the faster 1XXX and 1X3X methods to accumulate results non-coherently with updated resources evaluation. Hence, eight DSP48 slices and 16 RAM blocks were added to the 1XXX method for accumulation results on nine hardware Doppler bins. Similarly, the 1X3X

Table 3—II: Execution Times of Four Partial Acquisition Algorithms over ±4 kHz Doppler Span

<table>
<thead>
<tr>
<th>Execution time [ms]</th>
<th>1234</th>
<th>1XXX</th>
<th>1X3X</th>
<th>1111</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup time (#quarter * time + avg. time)</td>
<td>4 1.2 + 1.0</td>
<td>1 1.2 + 1.0</td>
<td>2 1.2 + 1.0</td>
<td>1 1.2 + 1.0</td>
</tr>
<tr>
<td>Processing time (#alignment * #bins * #ms)</td>
<td>4 1.4</td>
<td>4 1.4</td>
<td>4 3.4</td>
<td>4 3.4</td>
</tr>
<tr>
<td>Search time</td>
<td>0.2</td>
<td>-0.2</td>
<td>-3.4</td>
<td>0.2</td>
</tr>
<tr>
<td>Total</td>
<td>144.0</td>
<td>15.8</td>
<td>44.4</td>
<td>144.0</td>
</tr>
</tbody>
</table>
method requires three additional DSP48 slices and six RAM blocks to accumulate results for four distinct Doppler bins, as summarized in Table 3–IV.

Note that the resource evaluation of a “Full Method” (not implemented herein) has been appended to these two tables as a comparison benchmark. In this evaluation, it is considered that one Carrier Wipe-off module processes 4 ms worth of signal, the resulting samples being stored prior to the FFT module, where they are later processed in a burst and stored again. The IFFT produces 8192 complex samples. Equivalently, four smaller FFT modules used in parallel would require the same memory resources.

In order to get faster results, Matlab simulations were run where each algorithm accumulated their results non-coherently for 80 ms, searching only for Doppler bins over ±4 kHz. In Table 3–V, the 1234 and 1XXX methods outperform the other two. This may be explained by fewer non-coherent integrations possible with 1X3X and increased accumulated noise (i.e. squaring losses) in the 1111 method.

From this comparison, the 1XXX method presents better performance than the 1234 (within a 80 ms limit and considering reduced resources) and should ideally be chosen. However, its 60 % resources increase is not worth its <20 % increase of both performance ratios. Finally, the 1234 method is chosen.

3.7 Justification of the Proposed Architecture

Bearing in mind that the GNSS navigation messages targeted in this paper are rated from 50 to 250 symbol/s, a slightly longer sub-optimal acquisition method may not be user perceivable; the worst case being the loss of the first data bit of a received sub-frame. This justifies resources reduction at the cost of a slightly longer Mean Time To Acquire (MTTA). These reduced resources can be compensated for by software intelligence:

1. SPZP requires synchronized triggering to minimize partial code auto-correlation losses and navigation bit inversion through a few acquisition iterations at different offsets within a 1 ms window. This reduces (and even drops) the requirement for managing two partial peaks.
2. Doppler resolution can be compensated for by a second acquisition search with a 488.28 Hz shifted Carrier NCO command. Indeed, on top of saving half the resources, a reduced computation time can be achieved: two SPZP iterations over 2000 samples, assuming the FFT is computed once for the replica and twice for the input (for both normal and shifted carrier), doesn’t take as long as a single DPZP iteration over 4000 samples, assuming one FFT computation for both replica and input signals.
Nevertheless, code resolution reduction from DPZP to SPZP cannot be compensated for; the worst case being a correlation peak loss of 25% in sub-chip alignment resulting in \(10\log_{10}0.75 = -1.25\) dB. In order to achieve parallel multi-signal acquisition, some compromises must be made: one of them is coherent integration being limited to 1 ms as a result of unknown data bit transitions and secondary codes, when applicable.

4 TEST METHODOLOGY

In order to get a representative assessment of the acquisition channel performances, a 1000-step Monte-Carlo test was automated, providing the following numbers:

1. Detection count;
2. False-Peak count, which may be caused by detection of bad Doppler or chip alignment;
3. Miss count; and
4. Total duration.

To achieve signal detection, several criteria must be met:

1. \(\text{PNR} \geq 10\);
2. \(\text{FSSPR} \geq 1.5\);
3. \(f_{\text{Doppler, acq}} = f_{\text{Doppler, track}} \pm 1.5 \cdot 488.28\) Hz; and
4. \(\tau_s \in [\tau_s - 1, \tau_s + 1] \) chips with code loop management and relative triggering delay compensation.

where:

\(\tau_s\) is the chip delay of the acquisition step \(s\), and
\(\delta_s\) is the theoretical chip drift due to Doppler during current \(T_s\) long step.

A minimal tolerance of ±3 chips accounts for noise and timing resolution limitations. For the third criterion, it is important to consider that the full Doppler span search is split into two consecutive acquisition iterations. At high signal strength, signal detection may occur with \(f_{\text{Doppler, acq}} = [-5, 5] \cdot 976.56\) Hz, even before acquisition is launched with \(f_{\text{Doppler, acq}} = 488.28 + [-5, 5] \cdot 976.56\) Hz, offsetting the Doppler frequency by 488.28 Hz compared to the current \(T_s\) long step. The last criterion introduces a chip tolerance between consecutive Monte-Carlo steps. In the case of several successive iterations, the chip can be expected to vary by \(\pm \delta_s\) derived in Equations (4.1) to (4.5).

\[
\begin{align*}
    f_{\text{code, Doppler}} &= f_{\text{code}} \cdot \left(1 + f_{\text{Doppler}}/f_{\text{RF}}\right) \quad \text{(4.1)} \\
    T_{\text{code, Doppler}} &= N \cdot f_{\text{code, Doppler}} \quad \text{(4.2)} \\
    \Delta_{\text{chip}}|_{T_{\text{code}}} &= |T_{\text{code, Doppler}} - T_{\text{code}}| / f_{\text{code}} \quad \text{(4.3)} \\
    \Delta_{\text{chip}}|_{T_{i}} &= N \cdot f_{\text{Doppler}} \cdot \left| f_{\text{RF}} + f_{\text{Doppler}} \right| \cdot T_{i} / T_{\text{code}} \quad \text{(4.4)} \\
    \delta_s &= \Delta_{\text{chip}}|_{T_{i}} \cdot T_s \quad \text{(4.5)}
\end{align*}
\]

where:

- \(f_{\text{code}}\) is the nominal code rate,
- \(f_{\text{Doppler}}\) is the Doppler frequency,
- \(f_{\text{RF}}\) is the RF frequency,
- \(f_{\text{code, Doppler}}\) is the code rate resulting of Doppler shift,
- \(T_{\text{code, Doppler}}\) is the corresponding code period,
- \(T_{\text{code}}\) is the nominal code period,
- \(\Delta_{\text{chip}}|_{T_{\text{code}}}\) is the chip drift after one code period,
- \(\Delta_{\text{chip}}|_{T_{i}}\) is the chip drift after one integration period, and
- \(T_s\) is the duration of one Monte-Carlo step.

Furthermore, in order to compensate for the proposed identified design weaknesses, coarse/fine increments \((\Delta_{\text{coarse}}, \Delta_{\text{fine}})\) are used to delay start time in the following manner:

- Given a 60 MHz sampling frequency \(f_s\), there are \(N_{TI} = 60000\) samples/ms leading to a \([0, N_{TI}]\) search span for chip alignment. Integer factors of \(N_{TI}\) are used to impose an offset on the incoming signal window relative to a 1 ms global pulse. In fact, this allows dealing with the SPZP potential pitfall identified in Section 2.2.
- Decimating the incoming signal down to 2 Msample/s implies averaging \(N_{\text{avg}} = 30\) consecutive samples into one, which could straddle two consecutive chips. In the case of GPS L1 C/A, there are \(N_{TI}/1023 = 58.651\) samples/chip. This phenomenon justifies the use of fine increments chosen in the \([1, N_{\text{avg}}]\) samples range, excluding 0 as it is tested within the preceding coarse search.
- A triggering mechanism synchronized with the 1 ms global pulse applies delayed start time with these coarse and fine increments along with the following logic, until detection is achieved:
  - The maximum PNR and FSSPR ratios are saved along with their corresponding code and Doppler bins for each coarse delay tested while the applied delay \(\delta_{\text{coarse}} = i \cdot \Delta_{\text{coarse}} \) with \(i < N_{\text{avg}}\), with the coarse iteration \(i\).
  - Then the delay \(\delta_{\text{fine}}\) at which the maximum ratios were obtained becomes the basis for the subsequent fine increment searches \((\delta_{\text{fine}} = \delta_{\text{coarse}} + j \cdot \Delta_{\text{fine}})\), while \(j \cdot \Delta_{\text{fine}} < N_{\text{avg}}\), with the fine iteration \(j\).
  - The code and Doppler bins associated with the maximum ratios obtained at \(\delta_{\text{coarse}} + \delta_{\text{fine}}\) are output to the managing software for validation and statistics computation.
  - Both 0 and 488.28 Hz carrier wipe-off offsets are tested along with the above procedure. Note that the second wipe-off search is only initiated after an unsuccessful 0 Hz offset search over \(i_{\text{max}} + j_{\text{max}} = N_{TI}/\Delta_{\text{coarse}} + N_{\text{avg}}/\Delta_{\text{fine}}\) iterations. Hence, Doppler frequencies located halfway between the hardware Doppler bins are
expected to take longer to detect, especially at low signal strengths.

- The managing software accumulates the execution duration over all the acquisition iterations \((i+j)\) of a Monte-Carlo step \((s)\). Obviously, the more iterations that are required, the longer the acquisition step takes.

Another validation proved no detection occurred in the absence of the targeted signal. The outcome was a 129.807658 s total time for 1000 steps of the above Monte-Carlo search performed with 20 000/3 coarse/fine increments, with 0 and 488.28 Hz frequency offsets. Since no signal was found, each one of the 1000 steps performed \((i_{\text{max}}+j_{\text{max}}) \cdot 2 = (3+9) \cdot 2 = 24\) acquisition iterations. One can thus estimate 5.41 ms per acquisition iteration, assuming the replica FFT computation is performed only once at the beginning. In VHDL simulations, the FFT computation, including averaging of 1 ms worth of input signal, takes 1.259 ms. Consequently, each IFFT takes 0.259 ms and the last peak search lasts 0.136 ms. Therefore, an acquisition iteration lasts 1.259 + 11 \cdot 0.259 + 0.136 = 4.244 ms. Hence, the average measured acquisition iteration time corresponds to the VHDL simulations with a slight overhead due to the time to access and process the results and update commands as well as to vary the triggering time alignment (within 1 ms), thus further corroborating the design.

The results presented next are obtained from a static location, according to either of two scenarios. First, a validation is conducted through simulation (cf. section § 5). Then, real signals are acquired (cf. section § 6). In both cases, a proper inline amplification chain ensures decent signal levels reach the analog 22.3 MHz wide RF frontend, before the IF signal may be digitized.

5 ANALYSIS OF SIMULATED SIGNALS RESULTS

HEAD-start validation is conducted through Spirent GSS 7700 simulated GPS L1 C/A signals with fixed signal strength.

5.1 Coarse/Fine Search Resolution

Successive Monte-Carlo steps were launched seeking the optimal combination of coarse and fine increments in terms of execution time and detection performances. The tested coarse increments were 5, 7.5, 10, 12, 15, 20, and 30 thousands while the fine increments were 1, 2, 3, 5, 6, 10, and 15. It is expected that the coarse and fine increment combinations presenting the highest computational effort should lead to better signal detection performances, at the cost of a possibly longer execution time. The computational complexity \(C\) of coarse and fine increments (over the full span) can be defined by their worst case number of iterations.

\[
C = \frac{N_{T_i}}{\Delta_{\text{coarse}}} + \frac{N_{\text{avg}}}{\Delta_{\text{fine}}} - 1 \quad (5.1)
\]

At first, different combinations of equivalent total complexity tests were launched to compare the impacts of these increments against one another. In Table 5–I, a computational effort of seven allows comparison of four different combinations.

In Figure 5–I, one may acknowledge the benefits of having a low \(\Delta_{\text{fine}}\), rather than a low \(\Delta_{\text{coarse}}\), as expected. While seeking the best fine increment, the coarse increment should intuitively be \(N_{T_i}/3\) to overcome the SPZP limitations with the lowest computational effort. Thus, extensive simulations at constant signal strength led \(\Delta_{\text{coarse, optimal}} = 20000\) and \(\Delta_{\text{fine, optimal}} = 3\), resulting in a search effort of 12 (cf. Table 5–I), achieving a performance improvement over those with a search effort of seven presented in Figure 5–I. These increments remain constant for the following tests.

5.2 Optimal PNR and FSSPR Thresholds

Another series of tests were conducted to establish the \(\text{PNR}_{\text{optimal}}\) and \(\text{FSSPR}_{\text{optimal}}\) thresholds. The FSSPR is first assessed from 1.0 (i.e., the second peak is completely ignored) to 2; the second peak being restrained to the noise floor level at FSSPR = PNR. Then, the PNR is increased until the detection rate clearly drops.

From these results see (Figure 5–II), \(\text{PNR}_{\text{optimal}} = 10\), meaning the noise floor level is negligible, while maximizing robustness (i.e., reducing false peak occurrences) without significantly increasing execution time. \(\text{FSSPR}_{\text{optimal}} = 1.5\), as a higher value has proven to be penalizing.

5.3 Negligible Impact of Doppler Frequency Offset

In order to prove the acquisition channel performance, a simulation is conducted over the 0
to $-3\,\text{kHz}$ Doppler range with $\text{PNR} = 10$ or 22 and $C/N_0 = 45$ dB-Hz. From Figure 5–III, several detection drops can be observed, at the profit of false peaks. Since the signal is known to be present, this is less of a concern as the miss rate does not exceed 1%; it may hence be explained by a software misinterpretation such as a slight lack of tolerance in the detection criteria or even a zero-padding removal management issue with respect to the 1 ms global pulse. These false peaks result in a greater standard deviation for the results in a step sequence taken over a satellite course from zenith to horizon (i.e., 0 to $-3\,\text{kHz}$ Doppler range). Nevertheless, the detection statistics detailed in Table 5–II are very comforting. Also, the duration time increases from $\sim$10 s for Doppler multiples of 976.56 Hz to $\sim$30 s peaks for Doppler at odd multiples of 488.28 Hz. Note that this duration accounts for 1000 acquisition steps, which really corresponds to 10–30 ms per step, i.e., 2 to 6 times a 5.41 ms long iteration per step.

### 6 ANALYSIS OF REAL-WORLD SIGNAL RESULTS

Real signals are acquired through a passive Novatel 704 GNSS antenna located on a 1 m high pole on the flat roof with clear visibility and a $10^\circ$ mask angle. For this paper’s objectives, it is assumed that all satellites of a given constellation perform equally in terms of acquisition, provided they are in the same conditions; the PRN is therefore not explicitly defined below.

From the detailed analysis performed in Section 3, the implemented solution is expected to suffer from a theoretical loss of $\sim 6\,$dB, on top of the frontend noise figure and extra processing losses specific to Galileo and BeiDou signals:

- **3.5 dB current sequential acquisition losses:**
  - 3.17 dB for the 22.3 MHz wide front-end Noise Figure (NF)
  - 0.29 dB for the sine approximation ($20\cdot\log_{10}(\sum_{\text{amplitude resolution}}/64)$)
- **4.3 dB general design (sub-optimal, yet low cost generic) losses:**
  - $< 0.5$ dB for integer (instead of float) computations
  - 1.58 dB for coarse alignment limitation (half of a third of a code alignment)
- ~1 dB for averaging (decimation and down-sampling)
- 0.28 dB for performing and storing FFT only on 8 MSB
- < 0.87 dB for 488 Hz Doppler bin resolution

- 1.25 dB design losses for BeiDou B1-I due to chip resolution (0.75 average ACF amplitude)
- 6.4 dB design loss for Galileo E1B/C:
  - 3 dB for using either data or pilot component
  - 3 dB for SSL approach
  - 0.4 dB for neglecting its 6 MHz sub-carrier.

### 6.1 Zenith to Horizon Acquisitions of Real GPS L1 C/A Signals

A Monte-Carlo acquisition step, as defined in Section 4, was continuously triggered upon its completion, resulting in a massive data collection. In order to preserve its good behavior over a few hours of log, the acquisition channel was provided with an updated Doppler frequency reference obtained through an independent tracking channel assigned to the same signal. Acquisition statistics were thus logged along with the corresponding tracking channel C/N₀ for post-processing over a satellite passage from zenith to horizon, thus covering a 0 to -3500 Hz Doppler frequency span as the satellite moves away from the antenna.

In Figure 6–I, the observed false peak spikes around -1100 and -2200 Hz are not due to low signal strength, but could be due to a zero-padding removal management issue. Indeed, removing zeros as in (5.2) may introduce false peak glitches, depending on which coarse offset first meets the established detection thresholds. These signal chipping rate proportional code delay glitches have amplitude of $N_{zeros}^{N_{chips}} / N_{zeros}^{N_{chips}}$ chips, causing false peaks whenever greater than $\max(3, \delta_s)$ chips.

One must bear in mind that the chip validation criterion is based on a given tolerance around the previous valid acquisition chip alignment. In the case of misses and false peaks, the chip drift over time between successful detections becomes underestimated; this contributes to further increasing the false detection rate, typical at lower signal strengths. Thus, in the case of the zero-padding glitches, the detection rate suffers most when the first Monte-Carlo acquisition step result (used as the reference for further chip index validation) is offset, compared to all following iterations.

Another interesting thing to point out is that the detection curve follows the $C/N_0$ (cf. right of Figure 6–I): the best detection scores were achieved with signal strengths above 41 dB-Hz.

### 6.2 Non-Coherent Integration Impact on Signal Strength Thresholds

Different non-coherent integrations of 1 ms windows drag down the minimal acquisition threshold, as seen in Figure 6–II, where 15 ms of non-coherent integration ensures 95% detection rate of signals throughout all tested signal strengths. Since the replica code length is not compensated for Doppler, increasing the total integration time will accumulate its code length offset, thus limiting acquisition sensitivity.

Although not easily applicable to the proposed solution, non-coherent integration could be taken to another level by a coherent/non-coherent hybrid approach, as is common practice for GPS L1 C/A acquisition. For example, using GPS L1 C/A with 10 ms coherent integration, one out of two (navigation bit transition free) being accumulated non-coherently would give a noticeable sensitivity boost, at the cost of longer acquisition time.

### 6.3 GNSS Signals Acquisition Results on L1

With a fully functional GPS L1 parallel acquisition channel, only a few adjustments are required to...
accommodate other constellations’ civil signals on L1, as detailed in the following paragraphs. One difference is the way to recover a chip offset from the 2048 IFFT samples. The method may be generalized as:

$$c_{peak}\bigg|_{T_I=1\text{ ms}} = \begin{cases} (s_{peak} + 1)rac{N|_{T_I=1\text{ ms}}}{N_{\text{IFFT}} - N_{\text{zeros}}} - 1 & s_{peak} \leq \frac{N_{\text{IFFT}} - N_{\text{zeros}}}{2} \\ (s_{peak} + 1 - N_{\text{zeros}})rac{N|_{T_I=1\text{ ms}}}{N_{\text{IFFT}} - N_{\text{zeros}}} - 1 & s_{peak} > \frac{N_{\text{IFFT}} - N_{\text{zeros}}}{2} \end{cases}$$

(5.2)

where:

- $s_{peak}$ is the peak sample index, and
- $c_{peak}\bigg|_{T_I}$ is the corresponding relative peak chip within a 1 ms coherent integration time $T_I$.

Another difference is the absolute chip alignment computation (relative to the global 1 ms pulse) considering different code periods, where Galileo E1 has four partial code trunks of 1023 chips/ms with the code quarter offset $m_{\text{offset}}$ ranging from 0–3:

$$c = \left[c_{peak}\bigg|_{T_I=1\text{ ms}} + \{\delta_{\text{coarse}} + \delta_{\text{fine}}\}rac{N|_{T_I=1\text{ ms}}}{N_{T_I}}\right] \mod N|_{T_I=1\text{ ms}} + m_{\text{offset}}\cdot N|_{T_I=1\text{ ms}}$$

(5.3)

The following paragraphs detail these other GNSS signal results.

### 6.3.1 GLONASS L1 Results over its FDMA Range

GLONASS L1 has a shorter, unique code with an FDMA scheme, leading to better performance than those obtained for GPS L1 as it offers a greater chip resolution; the 12 chips worth of zero-padding still cause detection glitches, but with a lower impact in terms of chip tolerance. Different RF frequencies are managed by an adjusted IF to baseband down conversion frequency command (cf. Table 1–I), which makes this particularity transparent for the remaining of the acquisition channel architecture. The same 41 dB-Hz threshold was observed.

### 6.3.2 BeiDou B1-I Results with less than 1 Sample per Chip

BeiDou B1-I introduces longer codes. The 2000 averaged samples are thus slightly under-sampling the 2046 chip long spreading code. The resulting code resolution cannot be expected to have the same reliability as for GPS L1. Nevertheless, the minimal ±3 chips tolerance should have mitigated this potential limitation.

Although BeiDou B1-I shares the same 50 Hz navigation message rate as GPS L1, its 1 kchip/s secondary code is laid over the 1 ms long primary code, making it the most challenging signal to acquire with a SPZP-like approach due to phase inversion probabilities. Indeed, it prevents coherent integration over more than one 1 ms code period without prior secondary code synchronization and wipe-off. Additionally, the 48 padded zeros make initial code alignment a crucial parameter, justifying the proposed iterative coarse increments approach.

However, the miss rate is null for signal strengths above 41 dB-Hz. This corroborates a limitation in the validation method and in the chip resolution, rather than an architectural design problem. In pure acquisition, i.e., without any knowledge about the searched signal, a traditional tracking channel with early and late correlators spaced ±0.5 chip apart from the prompt correlator should handle an acquisition result with a 2046/2000 = 1.023 chip resolution in at most three (i.e., acquisition result ± 1 chip) sequential chip searches, thus mitigating this limitation.

### 6.3.3 Galileo E1 B Results with BOC(1,1) Approximation

In the case of the 4 ms long Galileo E1B/C spreading codes, the last term of (5.3) may not be used for validation purposes as the acquisition iterations are triggered relative to the 1 ms global pulse, without any knowledge about the input signal code start. As of December 2013, the Galileo system only bears four satellite vehicles, on top of the two GIOVE satellites [23], so signals are not as readily available as for the full GPS constellation. Furthermore, the SSL approach with BPSK simplification lowers the potential harvested satellite signal strength, leading to poorer detection rate expectations.

With these simplifications, four Galileo E1 signal variations may be tested against, i.e., data (B) and pilot (C) channels with either lower (−) or upper (+) BOC(1,1) lobes. In Figure 6–III, miss rates for a 1 ms integration time are compared to avoid
being misled by a false peak caused by validation methodology limitations. Here are the assumptions:

- 1 ms integration window combined with coarse/fine alignment steps ensure similar acquisition conditions for all four Galileo signal variations;
- lower and upper lobes are obtained through a shifted IF to baseband frequency down-conversion command, and
- averaging at 2 Msample/s would cause the other lobes (due to subcarriers) to be filtered out.

Although it might seem that E1C performs better, the observed variations are mostly due to acquisition validation glitches, as confirmed by the E1B low miss rate at 30–31 dB-Hz.

7 CONCLUSION

New GNSS signals have introduced longer spreading codes and new modulation schemes. These codes varying lengths and periods have made FFT-based acquisition algorithms more complex. This paper thus presented a generic parallel acquisition algorithm based on a constant low FFT length, accommodating at least one L1 civil signal from all GNSS constellations, where a partial code acquisition approach is taken for the longer Galileo E1B/C signal.

Several acquisition approaches were compared in terms of execution time and resources leading to the selection and implementation of a preferred method. Execution time of one acquisition iteration is ~5 ms, in line with VHDL simulations and foreseen channel management overhead. Its predicted and justified weaknesses were compensated for in software, triggering iterative searches with 20 000/3 (coarse/fine) incremented offsets until the 1-ms coherent integration period and sample averaging limitations were overcome. More specifically, samples averaged at 2Msample/s and a 2048-point I/FFT module resulted in a 976.56 Hz frequency coefficient hardware resolution, which is not sufficient to comply with the 2/(3TI) rule of thumb. When required, this is compensated for by a 488.28 Hz offset command of the frequency wipe-off module. In order to assume signal detection, a new metric combo was introduced – a 10 dB Peak to Noise Ratio (PNR) and a 1.5 First To Second Squared Peak Ratio (FSSPR) – with 95% detection rates achieved with real signals. Experimentation showed that a 41 dB-Hz sensitivity threshold was required for 1 ms integrations while with 15 ms long non-coherent integrations, 37 dB-Hz signal strength was sufficient.

A Monte-Carlo test bench was used to determine the detection probability over a 3 kHz Doppler frequency span and a 35–45 dB-Hz signal strength range. Some validation limitations were identified and justified mathematically, leading to the scarce false peak glitches observed. Nevertheless, low miss rates were achieved throughout most experiments, demonstrating the proposed acquisition channel efficiency. The proposed generic method proved to be efficient at successfully acquiring GPS L1 C/A, GLONASS L1OF, Galileo E1B/C and BeiDou B1-I signals.

Future improvements will reuse acquisition data to perform a snapshot acquisition on other GNSS signals (on any frequency admitted by the RF front-end) with longer spreading codes at a targeted Doppler and chip offset with the 1XXX method defined for Galileo signals, while keeping the same HEAD-start acquisition channel (with triggering mechanism and different averaging ratio). This would especially be beneficial for multi-frequency acquisition of GPS L2CM being spread over 20 ms. This would make further good usage of resources on mobile devices, before it is shut down to save power. Also, transferring acquisition data to tracking channels will allow assessing the true Time To First Fix (TTFF) impact of the proposed acquisition channel. It will also be possible to extrapolate acquisition data from an L1 civil signal onto the same satellite signals on other frequencies. Also, the replica code should be compensated for Doppler in order to maximize sensitivity.

As long as the current GNSS signals will be available, this approach should remain valid and apply to most receivers, especially those embedded in mobile devices with limited space and power.

8 References