UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

Transceivers with positioning capabilities

A Master's Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Yanfang Zhu

In partial fulfilment

of the requirements for the degree of

MASTER IN TELECOMMUNICATIONS ENGINEERING

Advisor: Ana Isabel Pérez-Neira &

Carles Fernández Prades

Barcelona, January 2018

Abstract

The objective of this work is the analysis of various algorithms for the acquisition of GNSS signals and find an efficient acquisition algorithm that can be implemented on a software-based receiver.

In the first place, a description of the GNSS signals and the principle of acquisition theory is made. Next, a detailed description of the basic methods of searching and processing of the GPS signal on the acquisition stage. With this description, it is easier to understand the acquisition principle and the bottleneck of this process.

The main part of this project is the description of four algorithms and a full performance analysis. The chosen algorithms are compared with the reference one in term of the required number of operations, in other words, making a comparison between these methods in order to find the fastest algorithm.

All the analysed algorithms are implemented in Matlab and be evaluated with simulated signals. Finally, these algorithms are evaluated with a real signal source provided by a software receiver.

Acknowledgements

I would like to begin by thanking my advisors Dra. Anna Peréz and Dr. Carles Fernández of the communication Systems department at CTTC, for giving me the opportunity to research this project topic and for all the advices and guidance that they have given me over all this time. I would also like to thank the experts who were always available to resolve all my doubts: Dr. Javier Arribas and Dr. Marc Majoral. Finally, I must express my very profound gratitude to my family and all my friends for giving me assistance and support during my studies.

Table of contents

List of Figures

List of Tables

Chapter 1 : Introduction

1.1 Overview of GNSS acquisition

In Global Navigation Satellite System (GNSS) receivers, the acquisition process is the first stage of the signal-processing module. It consists of detecting visible satellite and estimating the coarse values of the carrier Doppler shift and the code delay of the satellite signals to then initialize the tracking stage.

Most of the Global Navigation Satellite Systems use Code Division Multiple Access (CDMA) techniques to multiplex several satellite signals on a common carrier frequency. Thus, each satellite is assigned with a unique Pseudo-Random Noise (PRN) code that modulates the transmitted signal. The mentioned PRN code spreads the signal over the spectrum, making it noisy and allowing for greater resistance to interference. Furthermore, the PRN codes are orthogonal or near orthogonal to provide good auto- and cross-correlation properties. (The correlation function is at a maximum when they are completely aligned, otherwise, it will be nearly to zero). Then the acquisition process is based on the correlation operation. GNSS receivers have prior knowledge of each satellite's PRN code and correlate the incoming signals with locally generated code replicas, to determine if a given satellite is visible or not.

The basic GNSS signal acquisition method is called serial search acquisition, it based on the use of a two-dimensional acquisition grid, which should cover the uncertainty of the GNSS signal Doppler and code delay values. To detect the presence of the GNSS signal, the received signal is correlated with a succession of locally generated replicas of the GNSS signal until the acquisition detector crosses a predefined threshold. This method is the simplest but slowest one, hence it search sequentially through all possible values of the two parameters, frequency and code delay. Thanks to the relation between the Fourier Transform and the circular correlation properties, the parallel code phase search (PCPS) acquisition method is introduced and the performance of the procedure is increased significantly. Nevertheless, the acquisition process is still the bottleneck of the whole GNSS system receiver module.

1.2 Motivation and Objective

Any GNSS receiver must acquire the signal before the signal processing to determine position or time. The critical step of initial synchronization, namely acquisition, determines whether a satellite is visible and then makes a rough estimation of frequency and spreading code delay.

GNSS-SDR (Software Defined Receiver) is a project developed by the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC). Thus, the software GNSS receiver allows a great flexibility in terms of configurations compared to the hardware receivers. For instance, it can target GPS L1 C/A or Galileo E1 B/C signals in L1 band as well as GPS L5 or Galileo E5a signals in L5 band. Until now, the acquisition algorithm implemented by this software receiver, namely Parallel Code Phase Search (PCPS) method, is not optimal in term of speed and there is still room for improvements.

In order to increase the whole receiver system speed, the time consumed by acquisition process is critical. Therefore, the main objective of this project consist of studying and analyzing as many as possible the existed acquisition methods and comparing with the implemented acquisition method at the present time.

Chapter 2 : GNSS signal Acquisition

2.1 GNSS signal structure

GNSS signals are composed of a carrier modulated by PRN and a binary navigation data. Initially, the GNSS signals were only based on one component, such as the global positioning system (GPS) legacy civil signal (L1 C/A), which was used for both data communication and ranging. The new generation of signals (such as GPS L1C, GPS L5, Galileo E1 OS and Galileo E5a/b) has two components. One is called the data component, which contains the navigation message, and the other is the pilot component or data-less component, which is not modulated by a navigation data stream.

The received signal in white Gaussian noise environment without multipath effect can be represented by:

$$y(t) = \sum_{i=1}^{N_s} r_{RF,i}(t) + n_{RF}(t) \tag{1}$$

That is the sum of N_s useful signals, suppose there are N_s visible satellites. The expression of the signal transmitted by the i^{th} satellite and its received signal is affected by frequency Doppler and delay, after the down conversion and filtering, it can be generically represented as follow:

$$\begin{aligned} S_i(t) = A_{i,d} c_{i,d}(t-\tau_i) d_i(t-\tau_i) \cos\left(2\pi\left(f_{IF} + f_{d,i}\right)t + \ \phi_{i,D}\right) \\ + A_{i,p} c_{i,p}(t-\tau_i) \sin\left(2\pi\left(f_{IF} + f_{d,i}\right)t + \ \phi_{i,P}\right) + n(t) \end{aligned} \tag{2}$$

Where:

- A_i: signal amplitude that depends on the total signal power C
- τ: received PRN code phase delay
- $c_i(t-\tau_i)$: spreading sequence which is given by the product of several terms and it is assumed to take value in the set $\{-1, 1\}$;
- $d_i(t-\tau_i)$: Navigation message, BPSK modulated, containing satellite data
- f_{IF}: receiver intermediate frequency
- f_d: Doppler frequency
- ϕ_i: initial phase offset
- n: incoming noise

The spreading sequence $c_i(t)$ can be expressed as:

$$c_i(t) = c_{1,i}(t) c_{2,i}(t) p_i(t) \tag{3}$$

Where:

- $c_{1,i}(t)$: periodic repetition of the primary spreading code
- $c_{2,i}(t)$: secondary code
- $p_i(t)$: subcarrier signal

The subcarrier $p_i(t)$ is the periodic repetition of a basic wave that determines the spectral characteristic of $S_i(t)$ such as BPSK and Binary Offset Carrier (BOC). The primary spreading code $c_{1,i}(t)$ is a sequence of chips that can be expressed as $c_{1,i}(t) = \sum_k c_{1,i,k} P_{T_c}(t-kT_c)$, where $c_{1,i,k}$ is the k^{th} chip of the PRN sequence with a chip rate $R_c = \frac{1}{T_c}$, and $P_{T_c}(t)$ is a unitary rectangular pulse with duration T_c. Taking into account the spreading code on the pilot component is different from the spreading code present in the data component.

The generic expression of relevant GPS and Galileo received signals are characterized by the parameters provided by the table 2-2, and the table 2-1 provides the key features for the design of GNSS signals (Boto, 2014) (p.5-p.23), these parameters are:

- The carrier frequency f_L
- The spreading code chip modulation
- The length of spreading code expressed in chips $N_{cI}(chips)$ or chip duration $T_{cI}(ms)$ and the spreading code rate
- The navigation message data on the data component characterized by the symbol duration and the secondary code on the pilot component characterized by the code length and duration

		$f_L\ (MHz)$	Modulation	Spreading code		Data	Secondary code	
				Length $T_{cI}(ms)$ $N_{cI}(chips)$	Rate MHz	Symbol duration $T_d(ms)$	Code length $T_{c2}(ms)$ $N_{c2}(bits)$	Bit duration (ms)
GPS L1 C/A		1575.42	BPSK	1 1023	f_0	20	None	None
GPS L1C	Data	1575.42	BOC(1,1)	10 10230	f_0	10	None	None
	Pilot	1575.42	TMBOC(6,1,1/11)	10 10230	f_0	None	18000 1800	10
GPS L5	Data	1176.45	BPSK(10)	1 10230	$10f_0$	10	10 10	1
	Pilot	1176.45	BPSK(10)	1 10230	$10f_0$	None	20 20	1
Galileo E1 OS	Data	1575.42	CBOC(6,1,1/11,'+')	4 4092	f_0	4	None	None
	Pilot	1575.42	CBOC(6,1,1/11,'-')	4 4092	f_0	None	100 25	4
Galileo E5a	Data	1176.45	BPSK(10)	1 10230	$10f_0$	1	20 20	1
	Pilot	1176.45	BPSK(10)	1 10230	$10f_0$	None	100 100	1
Galileo E5b	Data	1207.14	BPSK(10)	1 10230	$10f_0$	1	4 4	1
	Pilot	1207.14	BPSK(10)	1 10230	$10f_0$	None	100 10	1

Table 2-1: Key feature of GNSS signals

	Data component				Pilot component		
	A_D	$c_{2,D}$	p_D	$\phi_{0,D}$	A_p	p_p	$\phi_{0,P}$
GPS L1 C/A	$\sqrt{2C}$	1	1	ϕ_0	0	None	None
GPS L1C	$\sqrt{\frac{C}{2}}$ 25%	1	$P_{BOC(1)}(t)$	ϕ_0	$\sqrt{\frac{3C}{2}}$ 75%	$P_{TMBOC\left(6,1,\frac{1}{11}\right)}(t)$	$\phi_0 + \frac{\pi}{2}$
GPS L5	$\sqrt{C}$ 50%	NH_{10}	1	ϕ_0	$\sqrt{C}$ 50%	1	ϕ_0
Galileo E1 OS	$\sqrt{C}$ 50%	1	$P_{CBOC\left(6,1,\frac{1}{11},'+'\right)}(t)$	ϕ_0	$\sqrt{C}$ 50%	$P_{CBOC\left(6,1,\frac{1}{11},'-'\right)}(t)$	$\phi_0 - \frac{\pi}{2}$
Galileo E5a	$\sqrt{C}$ 50%	1	1	ϕ_0	$\sqrt{C}$ 50%	1	ϕ_0
Galileo E5b	$\sqrt{C}$ 50%	1	1	ϕ_0	$\sqrt{C}$ 50%	1	ϕ_0
With							

$$P_{TMBOC\left(6,1,\frac{1}{11}\right)}(t) = \begin{cases} p_{BOC}(6), & for\ 4chips\ every\ 33\ chips \\ p_{BOC}(1), & otherwise \end{cases}$$

$$P_{CBOC\left(6,1,\frac{1}{11},'+'\right)}(t) = \frac{\sqrt{10}}{\sqrt{11}} p_{BOC(1)}(t) + \frac{1}{\sqrt{11}} p_{BOC(6)}(t)$$

$$P_{CBOC\left(6,1,\frac{1}{11},'-'\right)}(t) = \frac{\sqrt{10}}{\sqrt{11}} p_{BOC(1)}(t) - \frac{1}{\sqrt{11}} p_{BOC(6)}(t)$$

Where

$$p_{BOC(y)}(t) = sign\ (\sin(2\pi \cdot y \cdot f_0 t))$$

Table 2-2: GNSS signals characteristics

Composite GNSS signals can be acquired by ignoring one of the two channels. In this project, the acquisition principle is considered when one component is used, in particular, GPS L1 C/A signal have been used for simulation and analysis of all acquisition methods.

The received single GPS L1 C/A signal is affected by Doppler frequency and code delay, it can be expressed as:

$$r(t-\tau) = Ac(t-\tau)d(t-\tau)\cos(2\pi(f_1+f_d)(t-\tau)+\phi)+n(t) \qquad (4)$$

The received signal after down-converted and digitized can be represented as:

$$r[n] = A\, d[n]c[n-\tau]\cos(2\pi f_d n T_s + \varphi) + n[n] \qquad (5)$$

The acquisition process is based on a correlation operation between the locally generated signals with the received signal. In the presence of Doppler and code misalignment, the in-phase and quadrature correlator outputs can be expressed as (Borio, 2008) (Chapter 6):

$$Y_I(\tau, F_D) = \frac{Ad}{2}\operatorname{sinc}(\pi\varepsilon_f T_I)\, R(\varepsilon_\tau)\cos\left(\varepsilon_{\phi_0}\right) + \eta_I \qquad (6)$$

$$Y_Q(\tau, F_D) = \frac{Ad}{2}\operatorname{sinc}(\pi\varepsilon_f T_I)\, R(\varepsilon_\tau)\sin(\varepsilon_{\phi_0}) + \eta_Q \qquad (7)$$

Where

- $\varepsilon_\tau = \tau - \hat{\tau}$ is the Code delay error
- $\varepsilon_f = f_D - \widehat{f_D}$ is the Doppler frequency error
- R is the cross-correlation between the local and the incoming spreading sequence and
- η represents the noise at the correlator output, which is assumed to follow Gaussian distribution with variance equal to $\sigma^2 = \frac{N_0}{4T_I}$

Due to the orthogonality property presented by PRN sequence, the correlator output will give a peak value when the received signal is perfectly aligned with the local replica, in other words, the Doppler frequency error and code delay error are zero.

2.2 Fundamentals of GNSS signal acquisition

The motivation of acquisition is to detect the presence of the signal and provide a rough estimation of Doppler frequency and the code phase delay. The code phase of the C/A code refers to the time alignment of the PRN code associated with a specific satellite. This parameter is necessary to be known in order to generate a local PRN code that is perfectly aligned with the incoming code. In another side, the satellite motion will introduce a Doppler frequency shift both on the carrier frequency and on the PRN sequence code. It is a relevant parameter for performing the acquisition of the GNSS signal. For a stationary observer, the maximum Doppler frequency shift is around ±5 kHz. In a more general case, if the receiver is carried on a vehicle, which moves at a high speed, it is reasonable to assume that the maximum Doppler shift is ±10 kHz. Thus, the acquisition involves a two-dimensional search through various frequencies and code delays that are bounded by the frequency and time uncertainty. For GPS L1 C/A signal, the frequency and time uncertainty bounds are defined by ±10 kHz and 1023chips respectively.

In addition, it is important to determine the frequency bin size, thus the frequency error is limited by the Doppler bin step size.

- **Frequency Step size in acquisition:**

As explained in the previous section, the acquisition process is based on the correlation between the local replica and the received signal. The locally generated signal is shifted by the Doppler frequency in order to cover the frequency searching domain. As mentioned before, the Doppler frequency range that needs to be searched is ±10 kHz, and the frequency step size is defined as a function of the maximum acceptable degradation on the detector.

The frequency error is limited by the Doppler bin step size and in particular

$$-\frac{\Delta F}{2} \leq \varepsilon_f < \frac{\Delta F}{2} \tag{8}$$

In the same way, the code delay residual error is limited by the size of the step that the acquisition process uses to search all the possible code delays:

$$-\frac{\Delta \tau}{2} \leq \varepsilon_\tau < \frac{\Delta \tau}{2} \tag{9}$$

By considering the in-phase and quadrature components expressions, it is possible to evaluate the coherent output SNR that becomes (Bernhard C. Geiger, 2013)(p.4-p.6):

$$\rho_{c,e} = \frac{A^2}{4}\frac{\sin^2(\pi T_I\, \varepsilon_f)}{(\pi T_I\, \varepsilon_f)^2}R^2(\varepsilon_\tau)\frac{2T_I}{\sigma_{IF}^2} = 2\frac{C}{N_0}T_I\frac{\sin^2(\pi T_I\, \varepsilon_f)}{(\pi T_I\, \varepsilon_f)^2}R^2(\varepsilon_\tau) \tag{10}$$

It can be noted that for $\varepsilon_f = 0$ and $\varepsilon_\tau = 0$, the coherent output SNR will be the maximum. Moreover, it is possible to define a loss associated to the Doppler frequency and code delay errors given by $\rho_{c,e}$ divided by the ideal coherent output SNR. In particular, the following expression is obtained:

$$L(\varepsilon_\tau,\, \varepsilon_f) = \frac{\sin^2(\pi T_I \varepsilon_f)}{(\pi T_I \varepsilon_f)^2}R^2(\varepsilon_\tau) \tag{11}$$

The loss is given by the product of two terms that respectively depend on the Doppler frequency and code delay errors. Thus, it is possible to isolate the loss associated with the frequency residual error and the one associated with the code delay error namely

$$L(\varepsilon_f) = \frac{\sin^2(\pi T_I \varepsilon_f)}{(\pi T_I \varepsilon_f)^2} \tag{12}$$

And

$$L(\varepsilon_\tau) = R^2(\varepsilon_\tau) \tag{13}$$

Noticing that the frequency error ε_f is multiplied by the factor T_I that represents the coherent integration time of the input signal. Thus, the effect of Doppler frequency bin size has to be lowered as the coherent integration time is increased and a common criterion, for setting ΔF, is the following

$$\Delta F \ \leq \frac{2}{3T_I} \tag{14}$$

For $\Delta F \ = \frac{2}{3T_I}$, the losses are 1.65 dB

$$10 * \log_{10}\left(\frac{\sin^2\left(\pi T_I \cdot \frac{2}{3T_I \cdot 2}\right)}{\left(\pi T_I \frac{2}{3T_I \cdot 2}\right)^2}\right) = 10 * \log_{10}\left(\frac{\sin^2\left(\frac{\pi}{3}\right)}{\left(\frac{\pi}{3}\right)^2}\right) = -1.65\ dB \tag{15}$$

For $\Delta F \ = \frac{1}{2T_I}$, the losses are 0.91 dB

$$10 * \log_{10}\left(\frac{\sin^2\left(\pi T_I \cdot \frac{1}{2T_I \cdot 2}\right)}{\left(\pi T_I \frac{1}{2T_I \cdot 2}\right)^2}\right) = 10 * \log_{10}\left(\frac{\sin^2\left(\frac{\pi}{4}\right)}{\left(\frac{\pi}{4}\right)^2}\right) = -0.9121\ dB \tag{16}$$

For integration time $T_I = 1\ ms$ and sampling frequency $f_s = 1.203\ MHz$, then the step size should be 500 Hz for a degradation of the received signal $\frac{C}{N_0}$ of 0.9 dB.

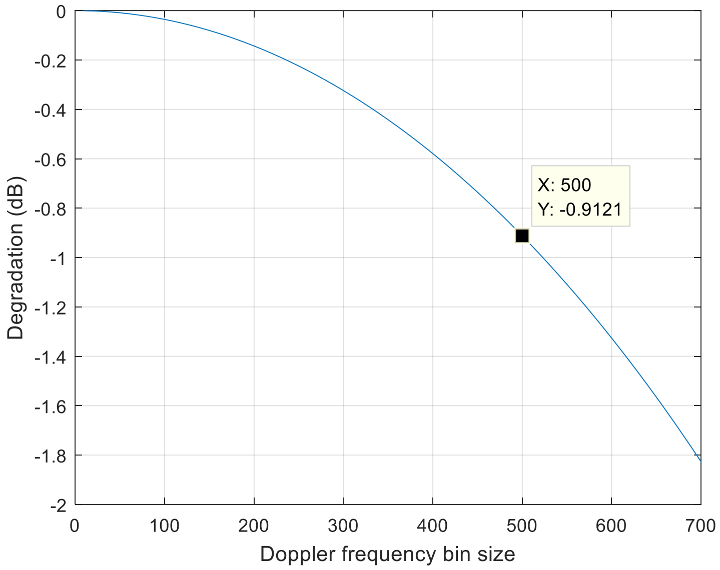


Figure 2-1: Peak Degradation for different Doppler bin sizes

To summarise, the Doppler bin size should be smaller than $\frac{2}{3T_I}$Hz, where T_I the dwell time and the code bin size is thus depends on the autocorrelation function shape. It is typically searched in increments of ½ chip.

2.3 GNSS Signal Detection

The acquisition block is a detection process and different processing techniques can be adopted, such as coherent and non-coherent integrations. The output of the processing block is a random variable, namely the decision variable, characterized by two probability density function (pdf) referring to the presence or absence of the desired signal. The probability of the decision variable passes a threshold, denote γ, is called the detection probability if the desired signal is present and false alarm probability if it is absent. The plot of the detection probability versus the false alarm probability is called the Receiver Operating Characteristic (ROC).

- **Coherent integration:**

As represented in the figure 2-2, the coherent integration consists in accumulating signal over T_c ms, where $T_c = MT_i$. M is the number if periods coherently summed and T_i is one spreading code period. In fact, it provides a good performance in terms of noise variance reduction. With the increase of the single trial integration time, the sensitivity is also increased. However, it also means that the computation complexity will be increase and acquisition process will be slower. Moreover, its performance is limited by the bit transitions due to the data navigation message.

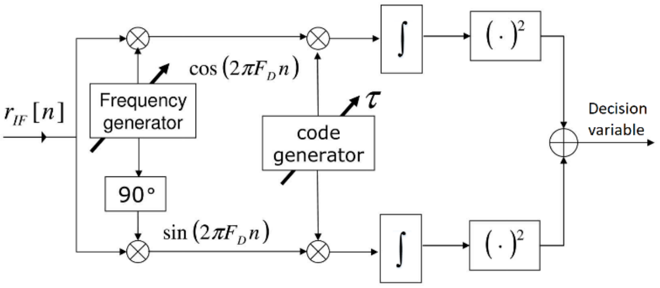


Figure 2-2: Coherent integration scheme

- **Non-coherent integration:**

In this project, the non-coherent integration technique has been chosen, it does not affect so much with bit transitions as the coherent integration. As shown in the figure 2-3 it consists in accumulating a number, denoted by k, of squared correlator outputs computed over T_i. In general, T_i coincide with one code spreading period. As a result, the total accumulating time T_I is equal to k times T_i ($T_I = KT_i$).

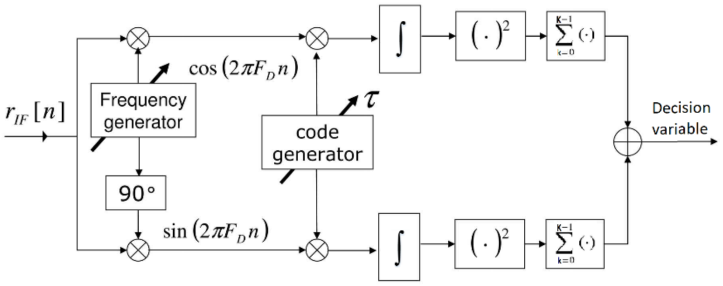


Figure 2-3: Non-coherent integration scheme

Then, the decision variable can be defined as (Foucras, 2015)(Chapter 3):

$$T = \sum_{k=1}^{K}\left(\left(\frac{Y_I(\tau, F_D)}{\sigma_\eta}\right)^2 + \left(\frac{Y_Q(\tau, F_D)}{\sigma_\eta}\right)^2\right) \tag{17}$$

- **Signal Detection statistic**

Suppose the navigation data is constant during the integration interval $[0, T_I)$ and assume the following approximated expressions, represented in (Borio, 2008)(Chapter 3):

$$Y_I(\tau, F_D) \approx \begin{cases} \frac{A}{2}\cos(\phi_0) + \eta_I & if \;\; \tau = \tau_0 \cap f_d = f_{d0} \\ \eta_I & otherwise \end{cases} \tag{18}$$

$$Y_Q(\tau, F_D) \approx \begin{cases} \frac{A}{2}\sin(\phi_0) + \eta_Q & if \;\; \tau = \tau_0 \cap f_d = f_{d0} \\ \eta_Q & otherwise \end{cases} \tag{19}$$

$Y_I(\tau, F_D)$ and $Y_Q(\tau, F_D)$ are acted as Gaussian random variable with mean equal to $\frac{A}{2}\cos(\phi_0)$ and $\frac{A}{2}\sin(\phi_0)$ respectively.

When non-coherent integrations are employed, the decision variable is obtained by squaring $Y_I(\tau, F_D)$ and $Y_Q(\tau, F_D)$ and summing K independent realizations of those random variables. The random variable $Y_I^2(\tau, F_D)$ is distributed according to a non-central chi-squared distribution with 2K degrees of freedom, whereas, if the signal is not present, $Y_I^2(\tau, F_D)$ is a central chi-squared random variable.

Usually, Binary Hypothesis is applied for GNSS signal detection problem:

- The null hypothesis, H_0: the signal is not present or not correctly aligned with the local replica;
- The alternative hypothesis, H_1: the signal is present and correctly aligned

Thus, the detection and the false alarm probabilities are defined as:

$$P_{fa}(\gamma) = P(Y > \gamma \mid H_0) = P(Y > \gamma \mid \tau \neq \tau_0 \ \cup f_d \neq f_{d0}) \quad (20)$$

$$P_d(\gamma) = P(Y > \gamma \mid H_1) = P(Y > \gamma \mid \tau = \tau_0 \ \cap f_d = f_{d0}) \quad (21)$$

Where the Y represents the decision variable and γ is a predefined variable, denoted as detection threshold.

Under H_0:

By using central chi-squared probability density function, the false alarm probability can be obtained as (Borio, 2008)(Section 5.1):

$$P_{fa,K}(\gamma) = \exp\left(-\frac{\gamma}{2\sigma_n^2}\right) \sum_{i=0}^{K-1} \frac{1}{i!}\left(\frac{\gamma}{2\sigma_n^2}\right)^i \quad (22)$$

Under H_1: Applying the non-central chi-squared distribution function

The detection probability is thus given by (Borio, 2008)(Section 5.1):

$$P_{d,K}(\gamma) = Q1\left(\sqrt{K\frac{\lambda}{\sigma_n^2}}, \sqrt{\frac{\gamma}{\sigma_n^2}}\right) \quad (23)$$

Where $Q_k(a, b)$ is the generalized Marcum Q-function defined as

$$Q_k(a, b) = \frac{1}{a^{k-1}} \int_b^{+\infty} x^k \exp\left(-\frac{a^2 + x^2}{2}\right) I_{k-1}(ax)\, dx \quad (24)$$

And λ is the non-centrality parameter:

$$\lambda = E^2[I] + E^2[Q] = \frac{A^2}{4} \quad (25)$$

When the square root of the decision variable is used to determine the presence of a satellite, then the decision variable is expressed as:

$$T = \sqrt{\sum_{k=1}^{K}\left(\left(\frac{Y_I(\tau,F_D)}{\sigma_\eta}\right)^2 + \left(\frac{Y_Q(\tau,F_D)}{\sigma_\eta}\right)^2\right)} \tag{26}$$

The false alarm and detection probabilities are defined as (Borio, 2008)(Section 5.1):

$$P_{fa,K}(\beta) = \exp\left(-\frac{\beta^2}{2\sigma_n^2}\right)\sum_{i=0}^{K-1}\frac{1}{i!}\left(\frac{\beta^2}{2\sigma_n^2}\right)^i \tag{27}$$

$$P_{d,K}(\beta) = Q1\left(\sqrt{K\frac{\lambda}{\sigma_n^2}}, \frac{\beta}{\sigma_n}\right) \tag{28}$$

By acknowledge of the distribution function under H0 and H1, it is possible to determine the detection threshold for a desired probability of false alarm and evaluate the probability of detection.

$$\beta = \sigma_n\sqrt{-2\ln(P_{fa})} \tag{29}$$

For example, in case of GPS L1 C/A signal with dwell time = 1 ms, the figure 2-4 shows the detection probabilities in function of false alarm probabilities for different C/N0 levels.

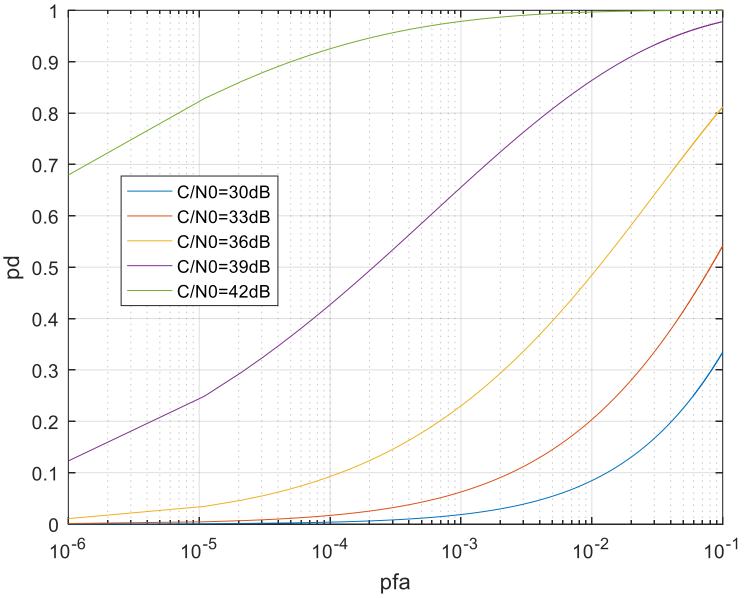


Figure 2-4: ROC curves for different values of C/N0

2.4 Basic GNSS signal acquisition methods

2.4.1 Serial search

The acquisition is a complex process that takes time and can be performed in numerous ways. The serial search strategy is the most basic of the search strategies in terms of algorithm complexity, but also the slowest. This method consists of trying every possible combination of carrier frequency and code phase. In the case of GPS L1 C/A signal, the search space has 1023 x (20000/500+1) = 41943 combinations, where 1023 represents all possible code delays and the number of frequency bins should be searched is calculated as a frequency searching range of 20 kHz divided by frequency step size of 500 Hz.

The Serial Search method system is represented by figure 2-5 (Borre, 2007)(Section 6.2). As explained previously, the incoming signal is multiplied by a locally generated spreading code. Then, another multiplication by a local carrier signal generates the in-phase, or I , and quadrature, or Q, channel. The I and Q signals are coherently integrated, squared and added. Further non-coherent integration may be performed by adding subsequent coherent integration results. The serial search must process all possible frequency and code delay combinations and each result is compared to a predetermined threshold value to determine the presence or absence of the single satellite signal.

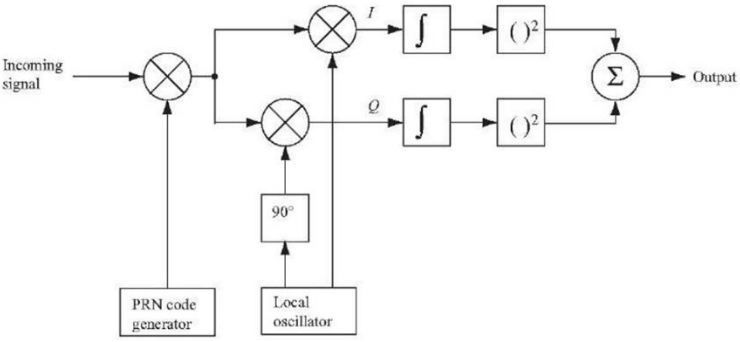


Figure 2-5: Serial Search acquisition algorithm

2.4.2 Parallel Code Phase Search

A procedure that parallelizes one of the searching dimension can reduce significantly the acquisition processing time. Since the amount of search steps in the code phase dimension is significantly larger than Doppler frequency steps, 1023 compared to 41 for the reference signal GPS L1 C/A, and then it is clear a procedure to parallelize the search for code phase is preferred. For such case, it only needs to step through each of the possible frequency bins in the search space.

A basic description of the method, namely Parallel Code Phase Search acquisition method, is illustrated by the figure 2-6 (Borre, 2007)(Section 6.4). For searching all possible code delays, a circular cross correlation is implemented in the frequency domain by multiplying the Fourier Transform of the incoming signal by the complex conjugate of the local code transform. The multiplication output is turn back to time domain by applying the inverse Fourier Transform, and the absolute value is taken and squared. In time domain, the correlation between the incoming signal and the local code is retrieved and a peak will exist at the code phase of the incoming signal if the correct frequency estimate was used.

This method based on the fact that the expression of the entire correlation function can be expressed as decomposition in Fourier series:

$$R(m) = \sum_{l=0}^{N-1} c(l)c(l-m) = IFFT\left(FFT\big(c(k)\big)\overline{FFT}\big(c(k)\big)\right) \qquad (30)$$

This method will be considered as the reference acquisition method in this project since it is commonly used and implemented in software-defined receiver.

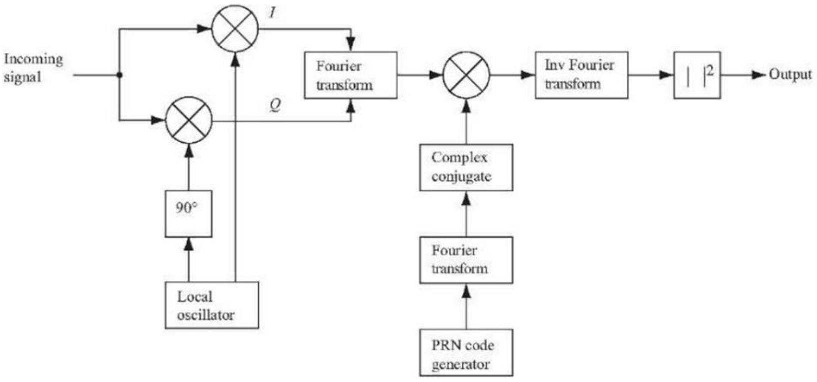
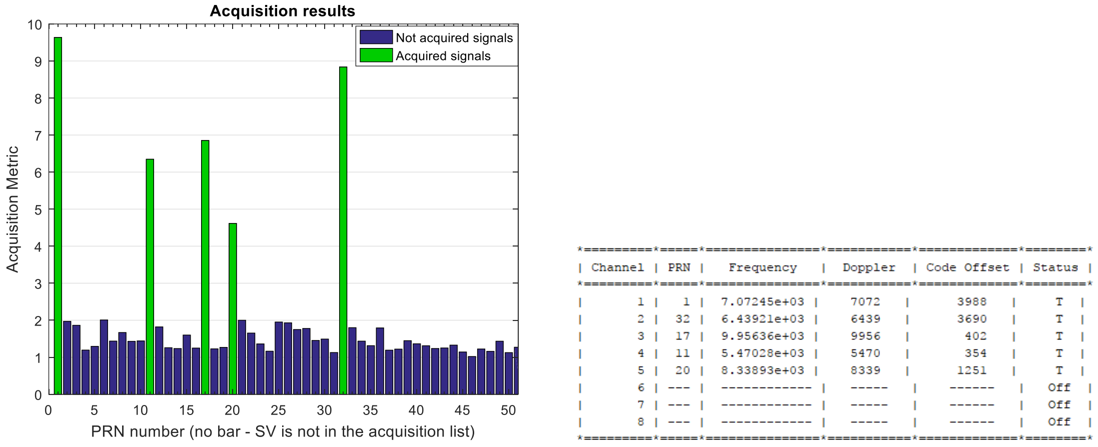


Figure 2-6: Parallel code phase search algorithm

This method has been already implemented in Matlab and the simulation result with real GPS signal, obtained from (Fernández, 2018), is shown in the figure 2-7.



```
*=========*=====*===============*===========*=============*========*
| Channel | PRN |   Frequency   |  Doppler  | Code Offset | Status |
*=========*=====*===============*===========*=============*========*
|       1 |   1 |  7.07245e+03 |    7072   |      3988   |    T   |
|       2 |  32 |  6.43921e+03 |    6439   |      3690   |    T   |
|       3 |  17 |  9.95636e+03 |    9956   |       402   |    T   |
|       4 |  11 |  5.47028e+03 |    5470   |       354   |    T   |
|       5 |  20 |  8.33893e+03 |    8339   |      1251   |    T   |
|       6 | --- |  ------------ |   -----   |    ------   |   Off  |
|       7 | --- |  ------------ |   -----   |    ------   |   Off  |
|       8 | --- |  ------------ |   -----   |    ------   |   Off  |
*=========*=====*===============*===========*=============*========*
```

Figure 2-7: Acquisition results with real data by PCPS

Chapter 3 : Study and analysis on GNSS signal Acquisition methods

3.1 Reduction of FFT computation size using the Overlap and Save method

The main advantage of Parallel Code Phase Search method is parallelizing code phase search through the Fourier Transform operation in order to reduce significantly the acquisition processing time. Obviously, most of the time consumed by this method depend on the operating time of the Fourier Transform. If the coherent integration time of acquisition process increases, the length of the FFT input will increase considerably and consequently the FFT time will also increase. One of the ideas to overcome this problem is to split the input signal into various blocks, therefore the FFT operation of each block will be faster due to the reduced input length. Finally, all the partial results are combined to reconstruct the complete correlation output. This ideal is feasible by applying the Overlap and Save (OS) or Overlap and Add (OA) method. Both methods are quite similar and share the same objective, in this case, the OS will be reviewed at following section.

3.1.1 Overview of Overlap and Save method

Theoretical background:

In digital signal processing, the overlap-save is an efficient way to compute the discrete convolution of a very long signal with a short length FIR filter. The concept is to divide the problem into multiple convolution of FIR response with short segment of the signal. Each convolution operation can be computed efficiently by applying the circular convolution theorem:

$$y[n] = DFT^{-1}\big(DFT(x[n]) \cdot DFT(h[n])\big) \tag{31}$$

Finally, these short segments $y[n]$ are concatenated to obtain the complete result. The implementation of the algorithm (Encyclopedia, 2017) is illustrated by figure 3-1:

Overlap and Save method:

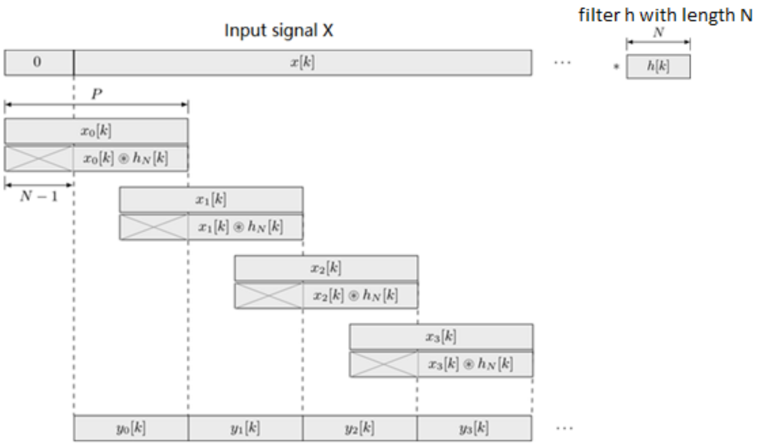


Figure 3-1: Overlap and Save Algorithm

3.1.2 Adaption and Implementation

As be mentioned before, the input length of FFT will be longer as the coherent integration acquisition time increases. Based on the principle of OS method, the correlation between the local replica and the received signal can be computed in smaller segments, thus making each FFT operation faster. Then reconstruct the partial results to obtain the complete correlation output.

Both methods are modified slightly respect to the original OS and OA methods in order to adapt the acquisition process. The implementation of both algorithms to perform correlation between two long signals of same length are illustrated by figure 3-2 and figure 3-3:

Overlap and Add:

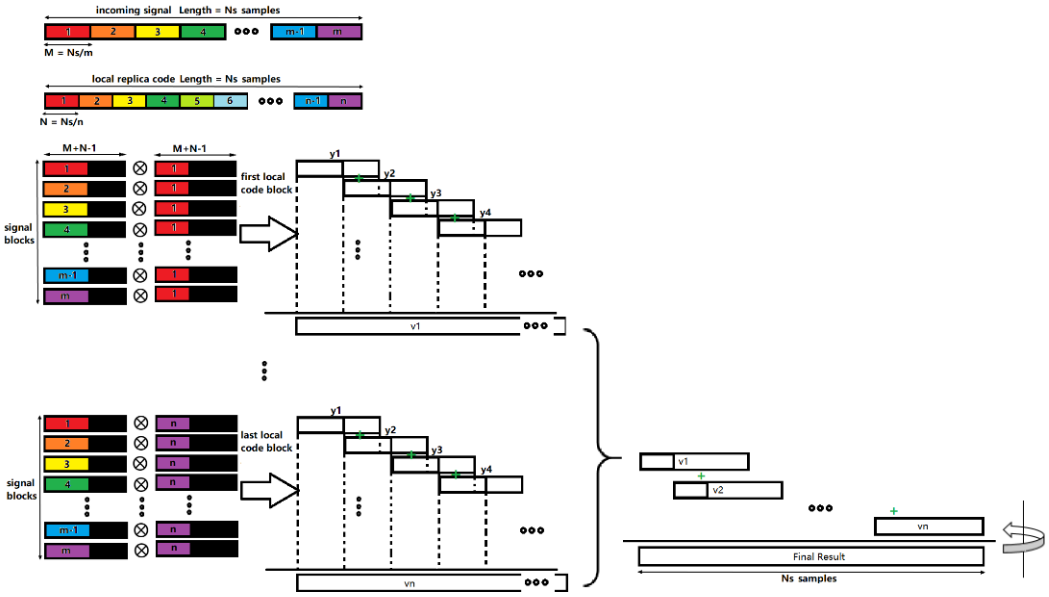


Figure 3-2: Overlap and Add modified and adapted for acquisition process

Overlap and Save:

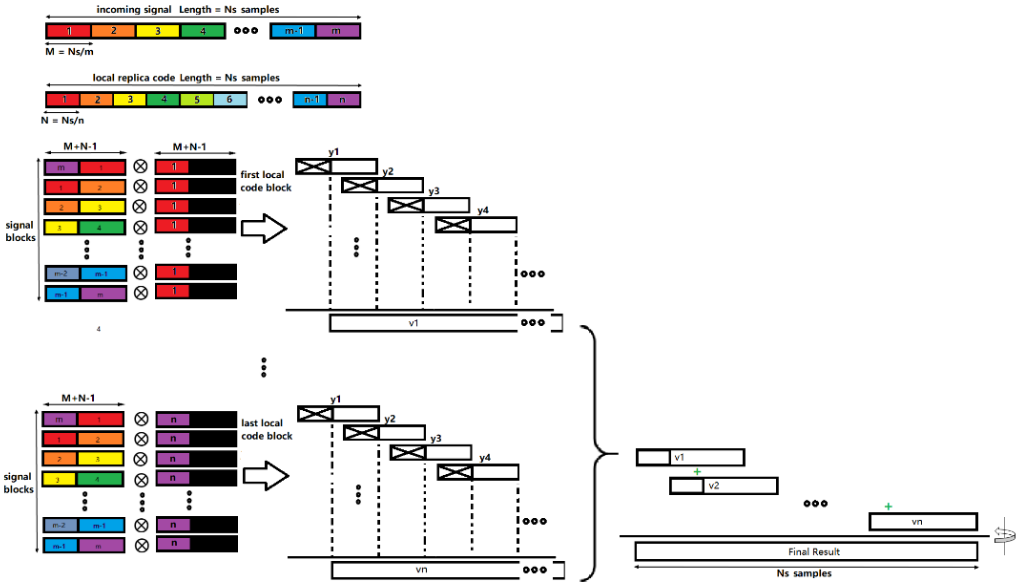
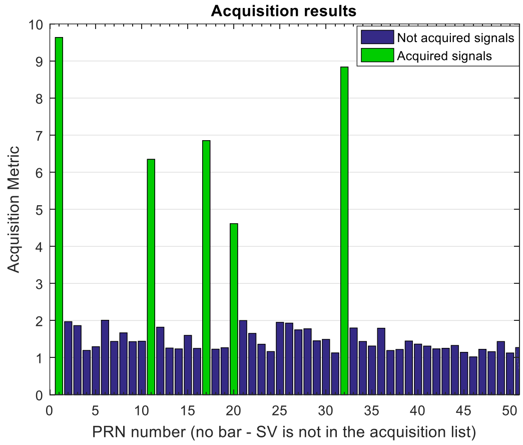


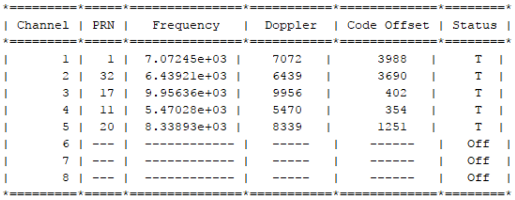
Figure 3-3: Overlap and Save modified and adapted for acquisition process

3.1.3 Result

Both methods were implemented in Matlab. As expected, the result of the acquisition process is the same as the original one. In fact, the acquisition method is not changed, but it is modified the way to compute the correlation between the two signals.

Result of the acquisition stage with “OA” or “OS” methods incorporated:





```
*=========*=====*===============*===========*=============*========*
| Channel | PRN |   Frequency   |  Doppler  | Code Offset | Status |
*=========*=====*===============*===========*=============*========*
|       1 |   1 |  7.07245e+03 |    7072   |      3988   |     T  |
|       2 |  32 |  6.43921e+03 |    6439   |      3690   |     T  |
|       3 |  17 |  9.95636e+03 |    9956   |       402   |     T  |
|       4 |  11 |  5.47028e+03 |    5470   |       354   |     T  |
|       5 |  20 |  8.33893e+03 |    8339   |      1251   |     T  |
|       6 | --- |  ------------ |    -----   |     ------    |   Off  |
|       7 | --- |  ------------ |    -----   |     ------    |   Off  |
|       8 | --- |  ------------ |    -----   |     ------    |   Off  |
*=========*=====*===============*===========*=============*========*
```

Figure 3-4: Acquisition results by real data with Overlap and Save method incorporate

3.1.4 Performance

One thing should to be taken into account is that FFT operation is implemented in FPGA, then the input FFT length has to be power of two. In the case of the length is not fulfil this condition, zero padding should be applied.

Considering the total number of operations required by an optimized FFT algorithm as the Cooley-Tukey algorithm (or radix 2) with complex input signal size N, which should be power of two, is $\frac{3}{2}Nlog_2(N)$. Which the number of multiplication is $\frac{N}{2}log_2(N)$ and the number of addition is $Nlog_2(N)$ (Duhamel, 1999)(Section 7.4.2).

In order to compare the time consumed by different methods in a better way, it is assumed that the time required by an addition operation is 't_a', and one multiplication operation is four times slowly than one addition operation. Total time consumption will be expressed in normalized unit respect to 't_a'.

Comparison between 'OA' or 'OS' and PCPS (Parallel code Phase search method)

Both methods (OA and OS) are based on the idea of splitting the input signal into smaller blocks, thus making each FFT operation faster. However, this also means that more FFT operations should be perform. Then, there is a trade-off between the number of FFT operations and the FFT consumed time, which depends on the input length.

Due to the peculiarity of FPGA, some combination of split block numbers cannot be applied. Because the reduced block length will be zero padded in order to fulfil the input length condition. And it does not make sense if the zero-padded block length, which should be shorter, becomes the same as the original FFT length.

Suppose the length of incoming signal is equal to Ns, and it is split into 'm' blocks with block length equal to M (where M = Ns/m).

The locally generated signal of length Ns is also split into 'l' blocks of length L (where $L = N_s/l$). Suppose that the FFT of the local signal block have been previously calculated and stored in memory.

According to these methods, there are as many FFT operations as the number of incoming signal blocks. The number of multiplications and IFFT operations are equal to the number of all possible block combinations ($m * l = \lceil \mathrm{Ns/N} \rceil * \lceil \mathrm{Ns/M} \rceil$). Taking into account the input length of each FFT operation is equal to M+L-1.

Required number of operations by "OS" method to perform one frequency bin search:

	Size of Vector	Number of Operations		Number of repetitions
FFT of signal block	Nspb = M+N-1	$\frac{3}{2}N_{spb}\log_2(N_{spb})$	M: $\frac{1}{2}N_{spb}\log_2(N_{spb})$ A: $N_{spb}\log_2(N_{spb})$	$\lceil \mathrm{Ns/M} \rceil$
Multiplication	Nspb = M+N-1	N_{spb}		$\lceil \mathrm{Ns/N} \rceil$* $\lceil \mathrm{Ns/M} \rceil$
IFFT of partial correlation	Nspb = M+N-1	$\frac{3}{2}\ N_{spb}\log_2(N_{spb})$	M: $\frac{1}{2}N_{spb}\log_2(N_{spb})$ A: $N_{spb}\log_2(N_{spb})$	$\lceil \mathrm{Ns/N} \rceil$* $\lceil \mathrm{Ns/M} \rceil$
Addition	Ns-N	Ns-N		$\lceil \mathrm{Ns/N} \rceil$
Squaring	Ns	Ns		1

Table 3-1: Required number of operations by "OS"

Required number of operations by parallel code phase search method to perform one frequency bin search:

	Size of Vector	Number of Operations		Number of repetitions
FFT of incoming signal	N_s	$\frac{3}{2}N_s\log_2(N_s)$	M: $\frac{1}{2}N_s\log_2(N_s)$ A: $N_s\log_2(N_s)$	1
Multiplication	N_s	Ns		1
IFFT	N_s	$\frac{3}{2}N_s\log_2(N_s)$	M: $\frac{1}{2}N_s\log_2(N_s)$ A: $N_s\log_2(N_s)$	1
Squaring	N_s	N_s		1

Table 3-2: Required number of operations by "PCPS"

Numerical Example:

Overlap and Save	Acquisition Parameters	Coherent integration time: 1ms Sampling frequency: 8 MHz Ns : 8000 N: 4 M: 4 Nspb: 4096 $\lceil \mathrm{Ns/M} \rceil$: 2000 $\lceil \mathrm{Ns/N} \rceil$: 2097			
		Size of Vector	Number of Operations	Number of repetitions	Total
	FFT of signal block	4096	M: $\frac{1}{2}N_{spb}\log_2(N_{spb})$ = 24,576 A: $N_{spb}\log_2(N_{spb})$ = 49,152	4	Multiplication: 565,056 Addition: 1,006,652 Total: 1,571,708 Total time: 3,266,876
	Multiplication	4096	4096	16	
	IFFT of partial correlation	4096	M: $\frac{1}{2}N_{spb}\log_2(N_{spb})$ = 24,576 A: $N_{spb}\log_2(N_{spb})$ = 49,152	16	
	Addition	8000 – 2097 = 5903	5903	4	
	Squaring	8000	8000	1	

Table 3-3: Numerical example by "OS"

Parallel code Phase	Acquisition Parameters	Coherent integration time: 1ms Sampling frequency: 8 MHz Ns : 8000 Zero padded: 8192 (length to be power of 2)			
		Size of Vector	Number of Operations	Number of repetitions	Total
	FFT of incoming signal	8192	Multiplication : $\frac{1}{2}N_s\log_2(N_s)$ = 53,248	1	Multiplication: 122,880

			Addition : $N_s \log_2(N_s)$ = 106,496		Addition: 212,992
	Multiplication	8192	8192	1	Total: 335,872 Total time: 704,512
	IFFT	8192	Multiplication : $\frac{1}{2} N_s \log_2(N_s)$ = 53,248 Addition : $N_s \log_2(N_s)$ = 106,496	1	
	Squaring	8192	8192	1	

Table 3-4: Numerical example by "PCPS"

It can be observed the proposed method is slower than the Parallel Phase Code Search method. The graphic 3-5 shows that the total consumption of the operating time increases with the number of blocks, in other words, the time required by the combination of all blocks cannot be compensate by the time saved by taking a FFT with less input length.

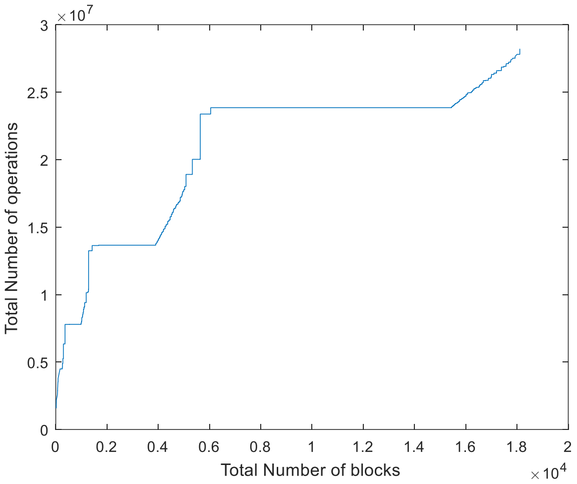


Figure 3-5: Number of blocks Versus Required Number of operations

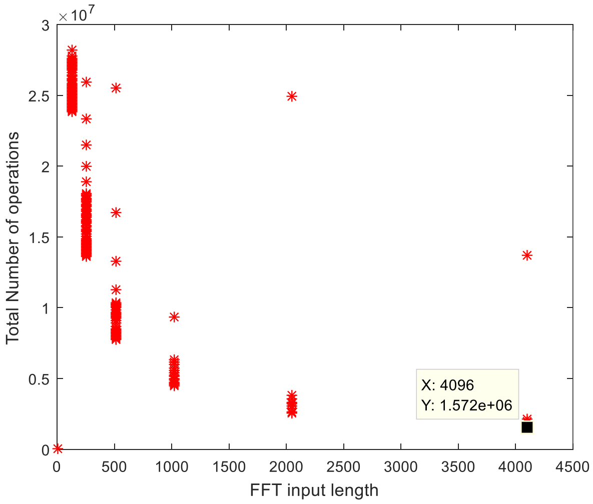


Figure 3-6: Length of input FFT size Versus Required number of operations

Table of comparison between ‘OS’ and ‘PCPS’ method using different input length:

		Incoming signal length				
		8000	16000	40000	80000	32000
number of operations & Total normalized time consumption	OS	M: 565056 A: 1006652 T: 3,266,876	M: 1212032 A: 2177148 T: 7,025,276	M: 2759744 A: 5024124 T: 16,063,100	M: 5847168 A: 10703612 T: 34,092,284	M: 2587904 A: 4681980 T: 15,033,596
	PCPS	M: 122880 A: 212992 T: 704,512	M: 262144 A: 458752 T: 1,507,328	M: 1179648 A: 2097152 T: 6,815,744	M: 2490368 A: 4456448 T: 14,417,920	M: 557056 A: 983040 T: 3,211,264
		Incoming signal length				
		64000	12500	25000	62500	125000
number of operations & Total normalized time consumption	OS	M: 5503488 A: 10019324 T: 32,033,276	M: 909524 A: 1631334 T: 5,269,430	M: 1941928 A: 3508433 T: 11,276,145	M: 5501988 A: 10011824 T: 32,019,776	M: 11659336 A: 21334372 T: 67,971,716
	PCPS	M: 1179648 A: 2097152 T: 6,815,744	M: 262144 A: 458752 T: 1,507,328	M: 557056 A: 983040 T: 3,211,264	M: 1179648 A: 2,097152 T: 6,815,744	M: 2490368 A: 4456448 T: 14,417,920

Table 3-5: Comparison between "OS" and "PCPS" for different FFT input size

In conclusion, only if the FPGA can perform all FFT block operations in parallel, these methods have the possibilities to accelerate the acquisition stage. In other cases, both proposed methods should be discarded.

3.2 Delay and Multiply

3.2.1 Introduction

The great improvement of Parallel Code Phase Search method compared to the traditional Serial Search method is that it only requires the search on the frequency domain and the search on the code phase domain can be replaced by performing the correlation using FFT operator.

The main purpose of "Delay and Multiply" method is to remove the Doppler frequency information of the input signal in an efficient way, and then only one time search of the code phase bins is required. Once the code delay is found, a quick search on the frequency bins of that code delay can determine the Doppler frequency. This method is represented in (David M. Lin, 1998)(p.321).

The basic idea of the method consisted of multiplying the incoming signal with a delayed version of itself in order to eliminate the Doppler frequency; it can be expressed with following equations:

The received signal after down conversion:

$$r(t) = \sum_{i=1}^{N_s} A_i\, D_i(t)\, C_i(t) \exp\left(j2\pi f_{d_i} t\right) = \sum_{i=1}^{N_s} s_i(t)\, C_i(t) \exp\left(j2\pi f_{d_i} t\right) \qquad (32)$$

Only one received satellite signal is analysed for simplification:

$$r(t) = C_s(t) \exp(j2\pi f_d t) \qquad (33)$$

The received signal multiply by complex conjugate of itself in order to eliminate the frequency component. It can be observed that the result signal is independent of frequency component. The module of the exponent $\exp(j2\pi f_d \tau)$ is a constant.

$$\begin{aligned} r(t) * r^*(t) &= C_s(t) C_s^*(t-\tau) \exp(j2\pi f_d t) \exp\left(-j2\pi f_d (t-\tau)\right) \\ &= C_s(t) C_s^*(t-\tau) \exp(j2\pi f_d \tau) \end{aligned} \qquad (34)$$

The key point is that after the C/A code multiplied with self-delayed version, the result new code maintains the orthogonality property. Moreover, the Matlab simulation show that the code phase delay can be find by its cross correlation, and it is the same as the code phase delay of the C/A code.

3.2.2 Implementation

Matlab implementation and simulation result:

1. Generating a received signal:

$$r(t) = \sum_{i=1}^{6} A_i\ C_i(t-\tau_i) \exp\left(j2\pi f_{d_i} t\right) + noise \qquad (35)$$

   Suppose there are six visible satellites with corresponding amplitude and code delay. Then multiply with its delayed conjugate version to obtain the new signal.
2. Generate local new code by multiplying the C/A code with the self-delayed signal.

3. Find the cross correlation between the new signal and new code in order to decide if there is a visible satellite and in affirmative case, obtain the code delay.
4. Finally, the Doppler frequency can be find by Fast Fourier Transform of the correlation between the signal and local code with corresponding code delay.

3.2.3 Simulation Result and conclusion

- Case 1: Received signal with constant amplitude and without noise: all six satellites can be detected with corresponding code delay. The figure 3-7 represents the cross correlation result of one visible satellite, the peak appears at correct code delay position and it is clear to be detected.

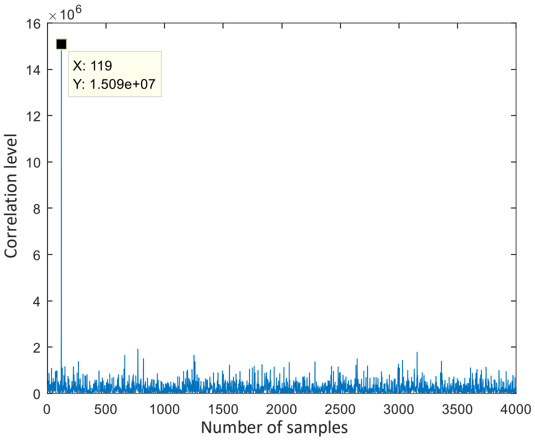


Figure 3-7: Delay and Multiply simulation result without noise

- Case 2: Received signal with variable amplitude and Gaussian noise. The simulation results show that weak signal cannot be detected. The figure 3-8 represents the cross correlation result of a weak signal with local generate code. It can be observed that neither the signal can be detected nor the peak position coincide with the correct code delay value.

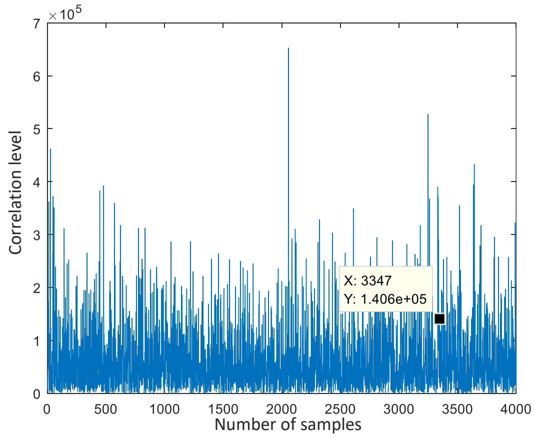
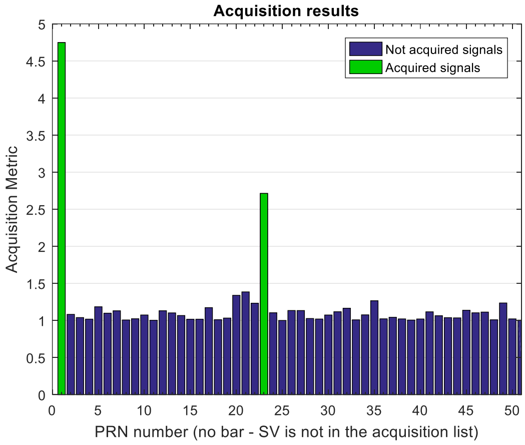


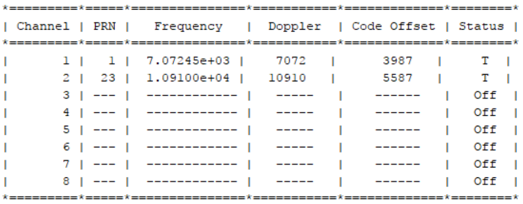
Figure 3-8: Delay and Multiply simulation result with Gaussian noise

As shown in the previous simulation result, the delay and multiply method represents a big drawback. The multiplication of the incoming signal by its delayed version introduces a squared noise component and consequently the weak signal cannot be detected.

Since the real received satellite signal is very weak, the SNR is too low to allow the direct application of this method in the acquisition stage. The received signal should be pre-processed to increase the SNR, for example, this method can combined with another one in order to make feasible to detect the visible satellite with correct code delay information.

This method is implemented in acquisition stage to analyse real received satellite signal. As expected there is no detected satellite signal with small coherent integration time, in this case $T_i = 1\ ms$. Thus, it is necessary to increase the coherent integration time to 10ms and result shows that only two satellites can be detected:





```
*=========*=====*===============*===========*=============*========*
| Channel | PRN |   Frequency   |  Doppler  | Code Offset | Status |
*=========*=====*===============*===========*=============*========*
|       1 |   1 |  7.07245e+03 |    7072   |      3987   |    T   |
|       2 |  23 |  1.09100e+04 |   10910   |      5587   |    T   |
|       3 | --- |  ------------ |   -----   |    ------   |   Off  |
|       4 | --- |  ------------ |   -----   |    ------   |   Off  |
|       5 | --- |  ------------ |   -----   |    ------   |   Off  |
|       6 | --- |  ------------ |   -----   |    ------   |   Off  |
|       7 | --- |  ------------ |   -----   |    ------   |   Off  |
|       8 | --- |  ------------ |   -----   |    ------   |   Off  |
*=========*=====*===============*===========*=============*========*
```

Figure 3-9: Delay and Multiply simulation result with real data

The main advantage and drawbacks of this method are:

- Advantage: Simple to implement and reduces the acquisition time.
- Drawbacks: the weaker signal cannot be detected, unless if there is some way to remove the detected stronger satellite signals. Another idea is pre-processing the incoming signal to separate the superposed signal into various signals. For example, SVD method can be applied in pre-processing stage, but this method is complex and require a lot of computation time.

In conclusion, this method is simple and easy to be implemented. Despite the fact that this method can speed up the acquisition process, it cannot be applied due to the degradation of signal detection. This method should be studied with more detail and propose some viable solutions to resolve the weak signal detection problem.

3.3 DBZP & DBZPTI

3.3.1 Overview of DBZP

This method is widely used by GNSS receivers due to its efficiency. It consists of splitting the incoming signal into smaller blocks in order to accelerate the correlation process. This idea is similar as the previous methods (OA or OS), but in this case, it is not necessary to reconstruct the complete correlation.

The detail description of DBZP found in (Foucras, 2015)(Section 5.1), here is a brief description:

Selection of parameters:

- User defined parameters:
  - t_i : Coherent integration time
  - $[-f_{Max};\ f_{Max}]$: Doppler frequency searching bandwidth
- Definition of others algorithm inputs:
  - N : Incoming signal length which depends on the sampling frequency and the coherent integration time
  - $N_b = \frac{2\,f_{max}}{\frac{1}{t_i}} = 2f_{max} * t_i$: Number of blocks in which the incoming signal is split in
  - $Nspb = N/N_b$: number of samples per block
  - $N_r = \frac{N_b}{t_i/T_c}$, where T_c is equal to the spreading code period. N_r is the number of circular permutations, if the coherent integration time is equal to T_c, then $N_r = N_b$.
- Resolution of Doppler frequency is fixed:
  - $\Delta f = \frac{2\,f_{max}}{N_b} = \frac{2\,f_{max}}{2\,f_{max}} * \frac{1}{t_i} = \frac{1}{t_i}$, longer is the integration time; better is the Doppler frequency resolution, but more computational consumption.

Description of the DBZP algorithm:

1. Suppose the incoming signal has been down-converted into baseband with total length equal to N samples. Then divide the input signal and the local replica code into Nb blocks of size Nspb, where Nspb = N/Nb.

2. Regrouping two consecutive input sub-blocks to form a new block of size 2*Nspb. In another side, sub-blocks of local replicas are zero padded by Nspb elements to produce the new double length block. Next, each one of new double size signal sub-blocks will correlated with corresponding new local sub-block. Finally, at the output of sub-block correlations, only the first part of the result values will be stored in a matrix of size Nb*Nspb and the second part will be discarded as shown in the fig 3-10.

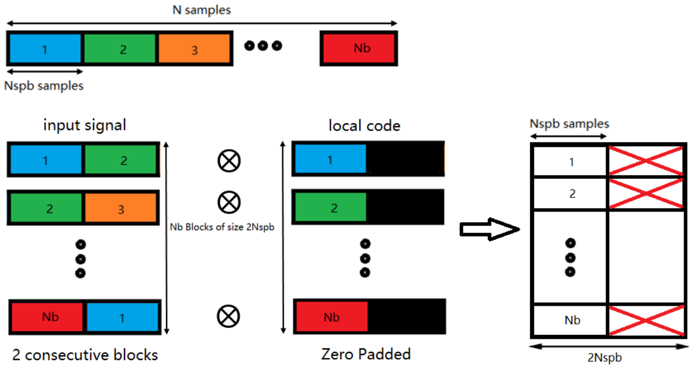


Figure 3-10: Partial correlation of DBZP

3. Up to now, only Nspb code delays has been tested. To ensure whole code delays be explored, the local code sub-blocks should be circularly permuted and repeate the previous correlation operation with the input sub-blocks. The number of circular permutation depends on the predetermined coherent integration time. If the coherent integration time is longer than the spreading code period, the number of circular permutations is less than the number of blocks due to the spreading code periodicity. The result output matrix produced by each one of the permutations will be stored and form a global matrix of size Nb*(Nr*Nspb).

4. Finally, applying Fourier Transform on each column of the output matrix to obtain the DBZP outputs. Thus, each row of the DBZP output matrix corresponding to a Doppler frequency bin, and each column to a code delay. If there is a peak present in the matrix, then the position of the peak reveals the Doppler frequency and code delay information.

The summary of the method is illustrated by the figure 3-11:

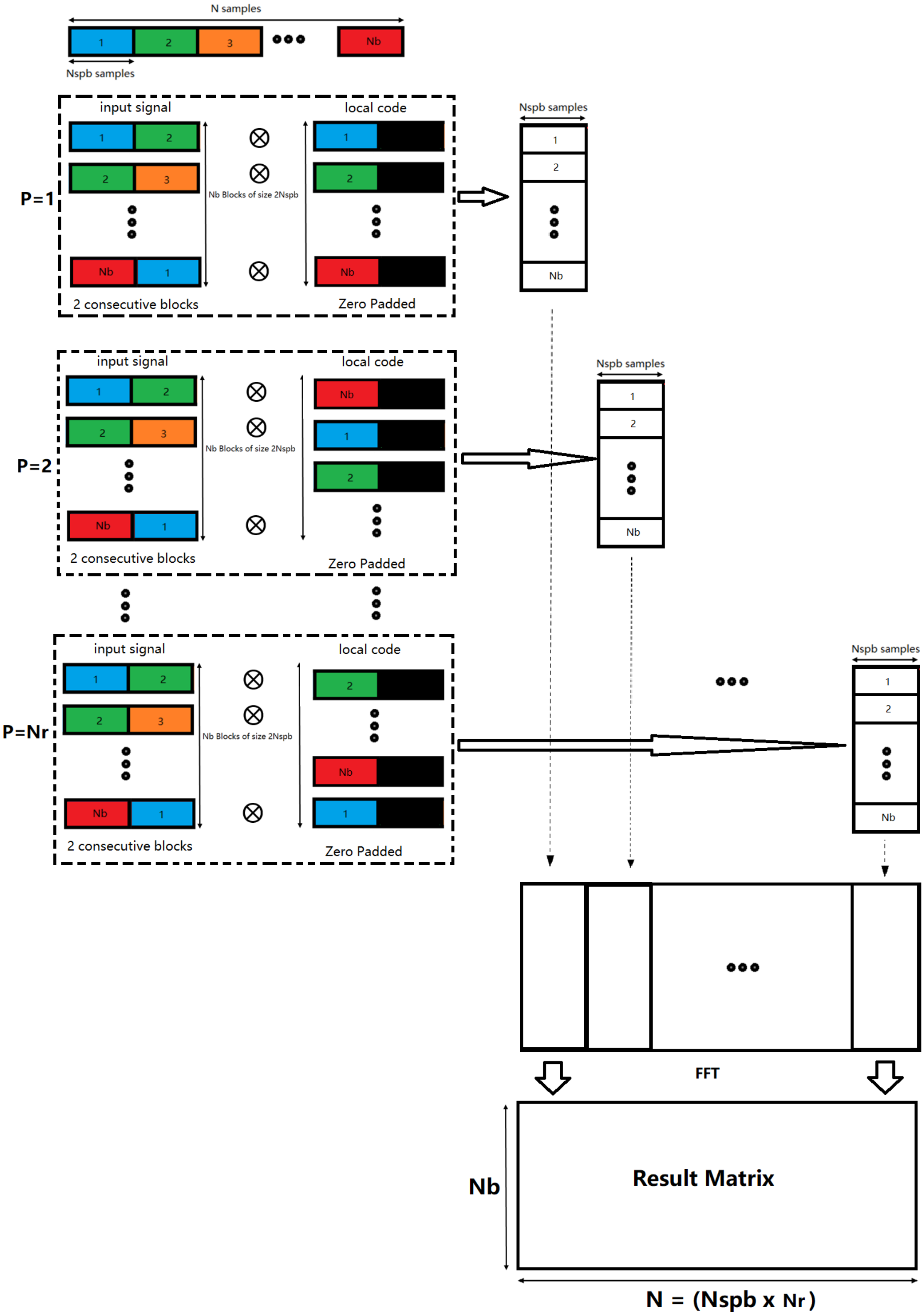


Figure 3-11: DBZP method description scheme

3.3.2 Partial correlation effect

To demonstrate the correct functionality of DBZP method, the key point is that the peak position of partial correlation corresponding to the peak position of the full correlation when the local and incoming spreading codes are perfectly aligned. Therefore it has no problem to find the correct value of the incoming signal code delay. However, the partial correlation represents a smaller peak power since its input length is much smaller than the length of full correlated signal. Thus it can cause posible undetection of the desirable signal.

Matlab implementation and analysis of partial correlation VS full correlation:

The simulation result represents the correlation between two generated signals with 193 samples shift. As shown in the figure 3-12, it can be proved that the correct sample (time) delay can be determined clearly. But due to the smaller length of partial correlation, the relation between peak and noise power is degraded.

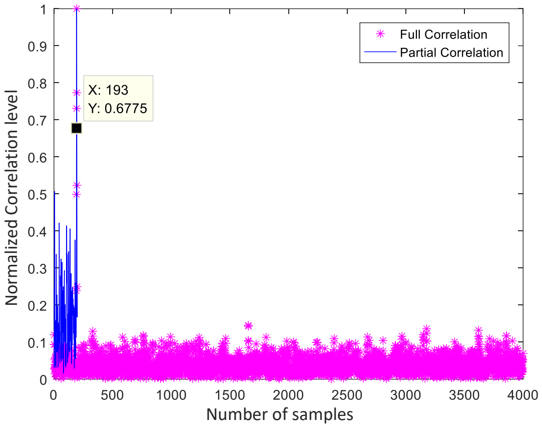


Figure 3-12: Partial correlation Versus Full correlation

As shown in the figure 3-13, every sub-block partial correlation outputs will represents the peak at correct code delay position. Then through the Fourier Transform operation, the Doppler frequency can be found.

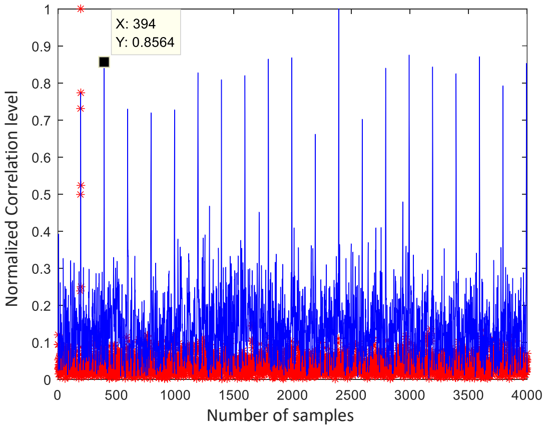


Figure 3-13: Partial correlation outputs of all sub-blocks

- Impact of the signal power in the partial correlation and the full correlation:

The performance of the DBZP depends on the partial correlation output. In case that the input signal SNR is four times smaller respect to the previous case, the figure 3-14 shows that the partial correlation is unable to detect the code delay, but the peak in the full correlation can still be distinguised. In others words, the weak signal can be undetectable using the DBZP method.

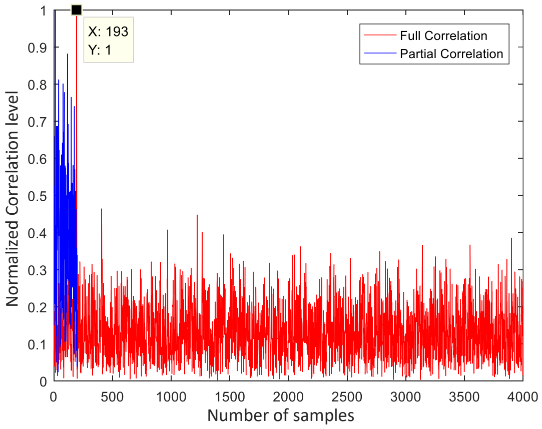


Figure 3-14: Signal amplitude versus correlation output

3.3.3 Matlab implementation and Simulation Result

1. Generating a received signal as superposition of six satellite signals:

$$r(t) = \sum_{i=1}^{6} A_i \; C_i(t-\tau_i)\exp\left(j2\pi f_{d_i} t\right) + noise \qquad (36)$$

Where:

- Amplitude values $A_i = [0.4810 \;\; 0.2887 \;\; 0.7066 \;\; 0.2833 \;\; 0.8008 \;\; 0.9901]$
- Doppler shift $f_{d_i} = [-7363 \;\; -318 \;\; 9601 \;\; 1645 \;\; 8140 \;\; 1787]$
- Code delay $\tau_i = [1267 \;\; 1125 \;\; 3280 \;\; 1369 \;\; 3488 \;\; 1072]$
- Additive white Gaussian noise

2. Suppose sampling frequency is 4 MSamples/s and Doppler range to be search is [-10 kHz, 10 kHz], the coherent integration time is 1ms.
3. Applying the DBZP method with following parameters definition:
   - Number of blocks: 20
   - Length of each block: 200 samples/block
   - Doppler Resolution: 1 kHz

Result of simulation: five of six satellite signals have been detected with

- codeDelay: [1267 1125 3280 3488 1072]
- Doppler: [-7000 0 10000 8000 2000]

As expected, all detected satellite signals have correct codeDelay values but with a certain doppler uncertainty less than 1 kHz.

Example:

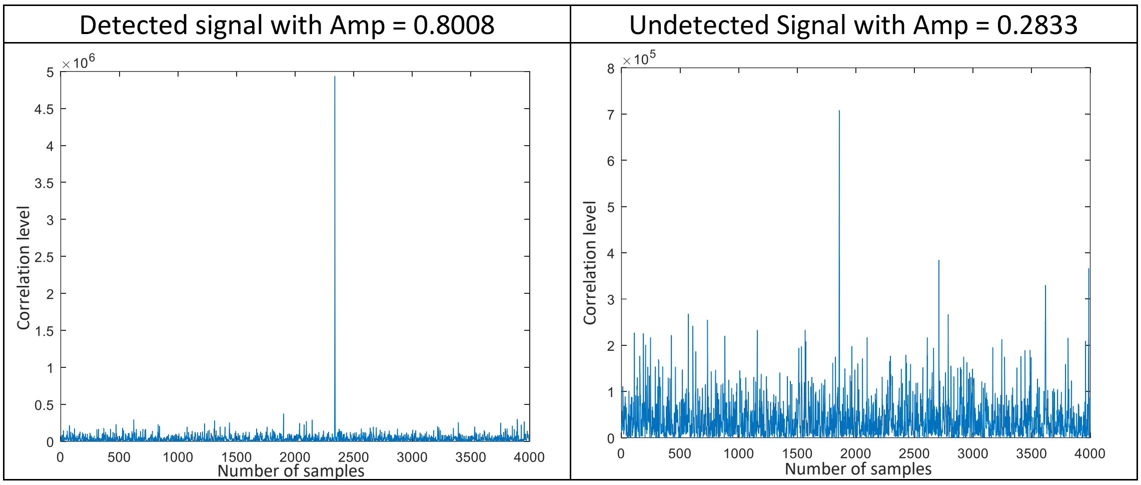
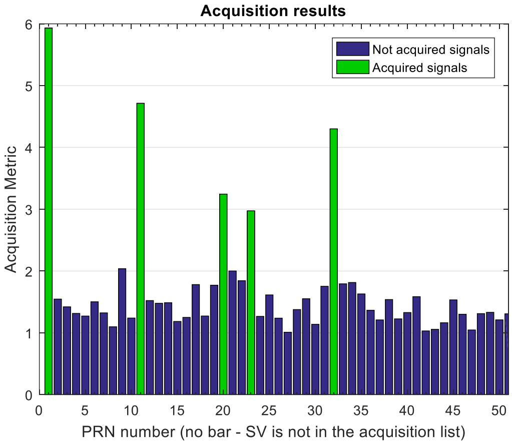
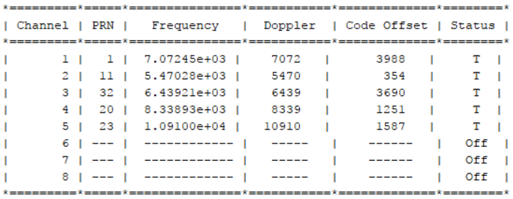


Figure 3-15: DBZP simulation results

- Simulation Result with real data: the satellite number 17 is not detected





```
*=========*=====*===============*===========*=============*========*
| Channel | PRN |   Frequency   |  Doppler  | Code Offset | Status |
*=========*=====*===============*===========*=============*========*
|       1 |   1 |  7.07245e+03 |    7072   |      3988   |    T   |
|       2 |  11 |  5.47028e+03 |    5470   |       354   |    T   |
|       3 |  32 |  6.43921e+03 |    6439   |      3690   |    T   |
|       4 |  20 |  8.33893e+03 |    8339   |      1251   |    T   |
|       5 |  23 |  1.09100e+04 |   10910   |      1587   |    T   |
|       6 | --- |  ------------ |   -----   |    ------   |   Off  |
|       7 | --- |  ------------ |   -----   |    ------   |   Off  |
|       8 | --- |  ------------ |   -----   |    ------   |   Off  |
*=========*=====*===============*===========*=============*========*
```

Figure 3-16: DBZP simulation results with real data

3.3.4 Performance

Due to FFT hardware implementation in FPGA requires the data length to be power of two. Thus, both incoming signal and local code sub-blocks are zero padded to their next power of two before the partial correlation computation.

Accordingly, sub-block length (2*Nspb) should be Zero padded to it next power of two (CNspb). Number of blocks Nb, in this case it corresponds to the number of permutations, also need to be zero padded before the last FFT computation (next power2 (Nb) = Nbz).

Number of Operations required by DBZP Method:

<table>
<tr><th></th><th>Size of Vector</th><th colspan="2">Number of Operations</th><th>Number of repetitions</th></tr>
<tr><td rowspan="2">FFT of signal block</td><td rowspan="2">CNspb</td><td rowspan="2">$\frac{3}{2}CN_{spb}\log_2(CN_{spb})$</td><td>M:
$\frac{1}{2}CN_{spb}\log_2(CN_{spb})$</td><td rowspan="2">$N_b$</td></tr>
<tr><td>A:
$CN_{spb}\log_2(CN_{spb})$</td></tr>
<tr><td>Multiplication</td><td>CNspb</td><td colspan="2">M : CN_{spb}</td><td>$N_b * N_r$</td></tr>
</table>

IFFT of partial correlation	CNspb	$\frac{3}{2} CN_{spb} \log_2(CN_{spb})$	M: $\frac{1}{2} CN_{spb} \log_2(CN_{spb})$ A: $CN_{spb} \log_2(CN_{spb})$	$N_b * N_r$
FFT of result partial correlation output	Nbz	$\frac{3}{2} N_{bz} \log_2(N_{bz})$	M: $\frac{1}{2} N_{bz} \log_2(N_{bz})$ A: $N_{bz} \log_2(N_{bz})$	$N_b * N_{spb}$
Squaring	Nbz	M : Nbz		$N_b * N_{spb}$

Table 3-6: DBZP required operation numbers

Number of Operations required by the reference Method: Parallel Code Phase Search

Nsz = next power 2 of (Ns)

N_f: number of Doppler frequency bins

	Size of Vector	Number of Operations		Number of repetitions
FFT of incoming signal	Nsz	$\frac{3}{2} N_{sz} \log_2(N_{sz})$	M: $\frac{1}{2} N_{sz} \log_2(N_{sz})$ A: $N_{sz} \log_2(N_{sz})$	N_f
Multiplication	Nsz	N		N_f
IFFT	Nsz	$\frac{3}{2} N_{sz} \log_2(N_{sz})$	M: $\frac{1}{2} N_{sz} \log_2(N_{sz})$ A: $N_{sz} \log_2(N_{sz})$	N_f
Squaring	Nsz	N_{sz}		N_f

Table 3-7: PCPS required operation numbers

The Doppler frequency resolution by DBZP is fixed and equal to (2fmax/Nb), but with the zero padding before the last FFT in order to increase the input length of FFT to be equal to the next power of two, therefore the Doppler resolution is changed. For example, if the number of blocks is 20, then it will be zero padded to reach 32 and the Doppler frequency resolution becomes to be 20 kHz/32 = 625 Hz.

In another side, the Doppler frequency bin size for PCPS method is determined by condition $\Delta F \leq \frac{2}{3T_I}$, as it has been explained in section 2.2. For instance, the number of frequency bins can be equal to 32 in order to fulfil the previous condition when $T_I = 1\,ms$. Moreover, both methods have the same Doppler frequency resolution.

Adquisition parameters		
Coherent integration time	1ms	2ms
Sampling frequency	8 MHz	8 MHz
Search frequency bandwidth	[-10 kHz, 10 kHz]	[-10 kHz, 10 kHz]
N	8000	16000
Nb	20	40
Nspb	400	400
Nr	20	20
Nf	32	64
DBZP	M: 3,456,000 A: 5,580,800 T: 19,404,800	M: 9,216,000 A: 14,745,600 T: 51,609,600

PCPS	M: 3932160 A: 6815744 T : 22,544,384	M: 16777216 A: 29360128 T : 96,468,992
Adquisition parameters		
Coherent integration time	4ms	5ms
Sampling frequency	8 MHz	8 MHz
Search frequency bandwidth	[-10 kHz, 10 kHz]	[-10 kHz, 10 kHz]
N	32000	40000
Nb	80	100
Nspb	400	400
Nr	20	20
Nf	128	128
DBZP	M: 28672000 A: 45875200 T: 160,563,200	M: 35840000 A: 57344000 T: 200,704,000
PCPS	M: 71303168 A: 125829120 T: 411,041,792	M: 150994944 A: 268435456 T: 872,415,232
Adquisition parameters		
Coherent integration time	10ms	1ms
Sampling frequency	8 MHz	12.5 MHz
Search frequency bandwidth	[-10 kHz, 10 kHz]	[-10 kHz, 10 kHz]
N	80000	12500
Nb	200	20
Nspb	400	625
Nr	20	20
Nf	256	32
DBZP	M: 637534208 A: 1.1409e+09 T: 718,848,000	M: 6950080 A: 11461760 T: 39,262,080
PCPS	M: 498073600 A: 891289600 T: 3.6910e+09	M: 8388608 A: 14680064 T: 48,234,496
Adquisition parameters		
Coherent integration time	2ms	5ms
Sampling frequency	12.5 MHz	12.5 MHz
Search frequency bandwidth	[-10 kHz, 10 kHz]	[-10 kHz, 10 kHz]
N	25000	62500
Nb	40	100
Nspb	625	625
Nr	20	20
Nf	64	128
DBZP	M: 17500160 A: 28523520 T: 98,524,160	M: 63750400 A: 103308800 T: 358,310,400
PCPS	M: 35651584 A: 62914560	M: 150994944 A: 268435456

	T: 205,520,896	T: 872,415,232
Adquisition parameters		
Coherent integration time Sampling frequency Search frequency bandwidth N Nb Nspb Nr Nf	10ms 12.5 MHz [-10 kHz, 10 kHz] 125000 200 625 20 256	
DBZP	M: 215500800 A: 350617600 T: 1,212,620,800	
PCPS	M: 637534208 A: 1.1409e+09 T: 3.6910e+09	

Table 3-8: Comparison between DBZP and PCPS in term of number of operations

3.3.5 Overview of DBZPTI

The DBZPTI is proposed to overcome the data bit sign transition problem, presented in (Foucras, 2015)(Section 5.2). In the case of Galileo E1 signal, the spreading code duration is same than the duration of the data bit. That means there is a high probability of data bit sign transition during the correlation period. Consequently, it will decrease the probability of detection significantly and this method is appropriate to overcome this problem.

The basic ideal is kept the same as the DBZP method. Here the difference between the DBZPTI and the DBZP will be briefly explained:

1. With reference to DBZPTI, 2Ti ms of incoming signal are required to be divided into Nb blocks.
2. Instead of circularly permutating the local code block, it is the incoming signal that is time shifted by one block to cover all possible code delays, it makes the bit transition resistance be possible.
3. Before the last FFT operation, each column of the result matrix should be zero padded. It can reduce the peak degradation due to the Doppler bin size effect.

Taking into account the correlation output only cover one coherent integration time even it require 2Ti ms of input signal.

The summary of the method is represented by the figure 3-17:

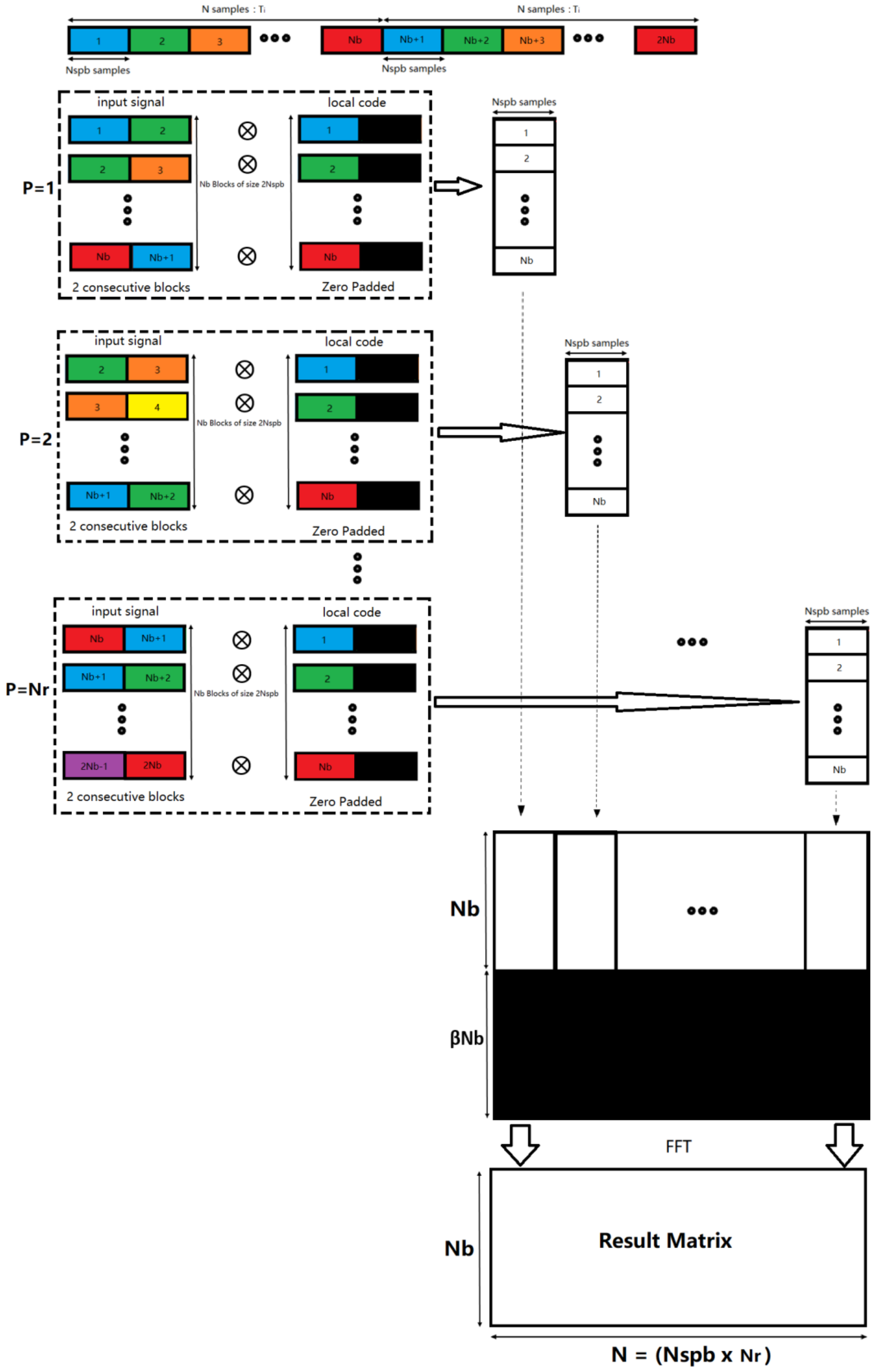


Figure 3-17: DBZPTI description scheme

3.3.6 Doppler frequency effect

The partial correlation output expressions are fully developed in (M. Foucras, 2014)(Section 5.1.2) for small Doppler frequency and code delay errors:

$$\tilde{I}_l(\tau, f_D) = \frac{Ad}{2}\mathrm{sinc}(\pi f_D t_b)\, R(\varepsilon_\tau)\cos\left(\pi f_D t_b + \varepsilon_{\phi_0}\right) + \eta_{I_l} \tag{37}$$

$$\tilde{Q}_l(\tau, f_D) = \frac{Ad}{2}\mathrm{sinc}(\pi f_D t_b)\, R(\varepsilon_\tau)\sin\left(\pi f_D t_b + \varepsilon_{\phi_0}\right) + \eta_{Q_l} \tag{38}$$

The FFT of the partial correlator outputs at the last step provides the DBZP/DBZPTI outputs, which are (M. Foucras, 2014)(Section 5.1.2):

$$\mathcal{F}\left(\tilde{I}_l(\tau, f_D)\right) = \frac{Ad}{2}R(\varepsilon_\tau)\,\mathrm{sinc}(\pi f_D t_b)\, N_b \frac{\mathrm{sinc}(\pi(m - f_D T_c))}{\mathrm{sinc}\left(\frac{\pi(m - f_D T_c)}{N_b}\right)}\cos(\phi) + \eta_{I_l} \tag{39}$$

$$\mathcal{F}\left(\tilde{Q}_l(\tau, f_D)\right) = \frac{Ad}{2}R(\varepsilon_\tau)\,\mathrm{sinc}(\pi f_D t_b)\, N_b \frac{\mathrm{sinc}(\pi(m - f_D T_c))}{\mathrm{sinc}\left(\frac{\pi(m - f_D T_c)}{N_b}\right)}\sin(\phi) + \eta_{Q_l} \tag{40}$$

where:

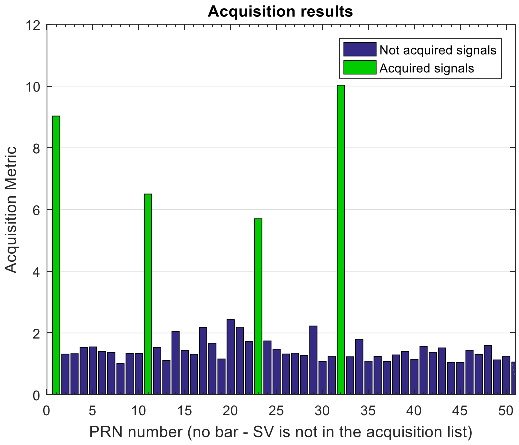
- N_b is number of blocks
- $\phi = \pi f_D t_b + \frac{\pi(N_b - 1)}{N_b}(f_D T_c - m) + \varepsilon_{\phi_0}$,
- $m = 0, \ldots, N_b - 1$ is the point where the FFT is taken and corresponds to a Doppler frequency bin

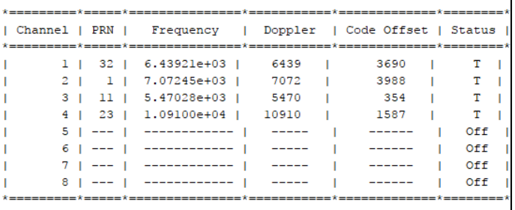
As mentioned previously each row of the DBZP output matrix corresponding to a Doppler frequency bin, it can be understood by the second sinc term of (38). It will present a peak value for the value of m which is nearest to the incoming Doppler frequency.

A related point to consider is that the degradation of the peak amplitude depends on the gap between $f_D T_c$ and m, in others words, it depends on the resolution of Doppler frequency. To overcome this problem one suggests to double the number of blocks, that means the uncertainty Doppler frequency bin is reduced to half but each partial correlation block length is also reduced to half. Another solution is using zero-padding before the last FFT operation as shown in the figure 3-17 and the length of zero-padding is defined by the parameter β. In fact, the more points to oversample the FFT result, the smaller is the Doppler frequency resolution and the smaller is the degradation of the peak.

3.3.7 Simulation Result

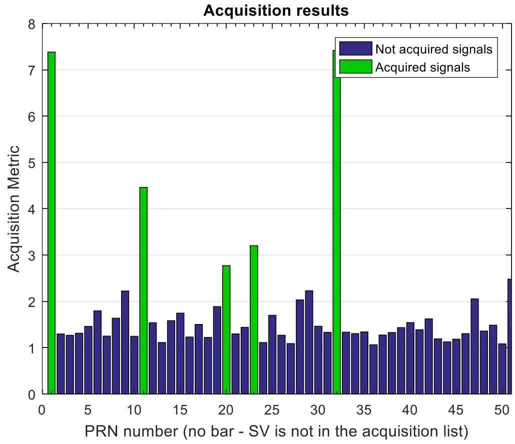
Beta = 1, Nb = 20, Nspb =200:

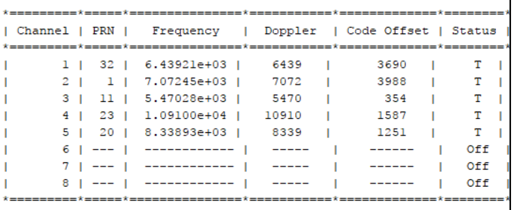




```
*=========*=====*===============*===========*=============*========*
| Channel | PRN |   Frequency   |  Doppler  | Code Offset | Status |
*=========*=====*===============*===========*=============*========*
|       1 |  32 |  6.43921e+03 |    6439   |      3690   |    T   |
|       2 |   1 |  7.07245e+03 |    7072   |      3988   |    T   |
|       3 |  11 |  5.47028e+03 |    5470   |       354   |    T   |
|       4 |  23 |  1.09100e+04 |   10910   |      1587   |    T   |
|       5 | --- |  ------------ |   -----   |    ------   |   Off  |
|       6 | --- |  ------------ |   -----   |    ------   |   Off  |
|       7 | --- |  ------------ |   -----   |    ------   |   Off  |
|       8 | --- |  ------------ |   -----   |    ------   |   Off  |
*=========*=====*===============*===========*=============*========*
```

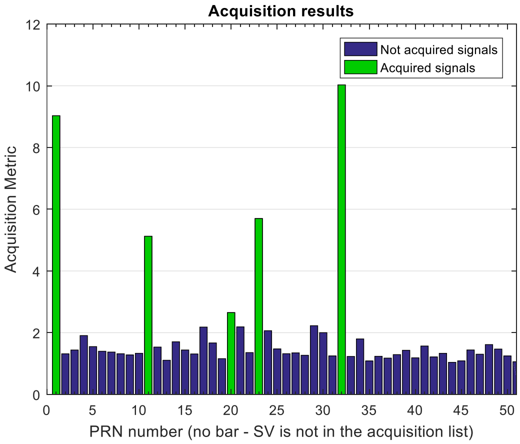
Beta = 3, Nb = 20, Nspb = 200

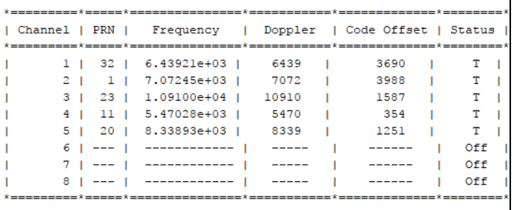




```
*=========*=====*===============*===========*=============*========*
| Channel | PRN |   Frequency   |  Doppler  | Code Offset | Status |
*=========*=====*===============*===========*=============*========*
|       1 |  32 |  6.43921e+03 |    6439   |      3690   |    T   |
|       2 |   1 |  7.07245e+03 |    7072   |      3988   |    T   |
|       3 |  11 |  5.47028e+03 |    5470   |       354   |    T   |
|       4 |  23 |  1.09100e+04 |   10910   |      1587   |    T   |
|       5 |  20 |  8.33893e+03 |    8339   |      1251   |    T   |
|       6 | --- |  ------------ |   -----   |    ------   |   Off  |
|       7 | --- |  ------------ |   -----   |    ------   |   Off  |
|       8 | --- |  ------------ |   -----   |    ------   |   Off  |
*=========*=====*===============*===========*=============*========*
```

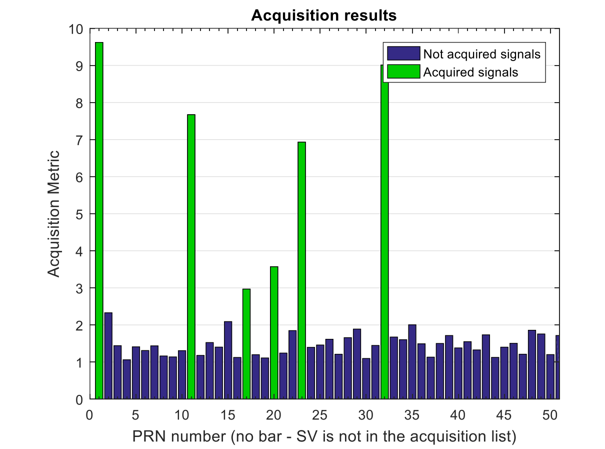
Beta = 3, Nb = 40, Nspb = 100

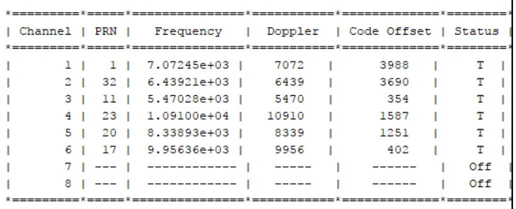




```
*=========*=====*===============*===========*=============*========*
| Channel | PRN |   Frequency   |  Doppler  | Code Offset | Status |
*=========*=====*===============*===========*=============*========*
|       1 |  32 |  6.43921e+03 |    6439   |      3690   |    T   |
|       2 |   1 |  7.07245e+03 |    7072   |      3988   |    T   |
|       3 |  23 |  1.09100e+04 |   10910   |      1587   |    T   |
|       4 |  11 |  5.47028e+03 |    5470   |       354   |    T   |
|       5 |  20 |  8.33893e+03 |    8339   |      1251   |    T   |
|       6 | --- |  ------------ |   -----   |    ------   |   Off  |
|       7 | --- |  ------------ |   -----   |    ------   |   Off  |
|       8 | --- |  ------------ |   -----   |    ------   |   Off  |
*=========*=====*===============*===========*=============*========*
```

Beta = 3, Nb = 80, Nspb = 50





```
*=========*=====*===============*===========*=============*========*
| Channel | PRN |   Frequency   |  Doppler  | Code Offset | Status |
*=========*=====*===============*===========*=============*========*
|       1 |   1 |  7.07245e+03 |    7072   |      3988   |    T   |
|       2 |  32 |  6.43921e+03 |    6439   |      3690   |    T   |
|       3 |  11 |  5.47028e+03 |    5470   |       354   |    T   |
|       4 |  23 |  1.09100e+04 |   10910   |      1587   |    T   |
|       5 |  20 |  8.33893e+03 |    8339   |      1251   |    T   |
|       6 |  17 |  9.95636e+03 |    9956   |       402   |    T   |
|       7 | --- |  ------------ |   -----   |    ------   |   Off  |
|       8 | --- |  ------------ |   -----   |    ------   |   Off  |
*=========*=====*===============*===========*=============*========*
```

3.3.8 Performance

Required number of operations by DBZPTI:

- Double Block length: $CN_{spb} = next\ power2\ (2 * Nspb)$
- Input length of the last FFT: $N_{bz} = next\ power2\ ((1+\beta) * N_b)$
- N_{bz}: depends on the value of β

The output correlation amplitude can be greatly reduced by high incoming Doppler frequency. As mentioned previously, zero-padding before the last FFT operation can reduce the peak degradation and the length of the last FFT input is defined by $(1+\beta)N_b$.

Depending on the definition of parameter β, the size of input length can be power of two or an integer multiple of the number of blocks. In case the zero-padded length is power of two, the total number of operations required by DBZPTI is the same than the DBZP method. However, DBZPTI takes the advantage that it is resistant to the data bit sign transition. In case of $\beta = 1$ or $\beta = 3$, only the last FFT operation will be affected and the acquisition time will be increased proportional on the number of input samples as shown in the figure 3-18.

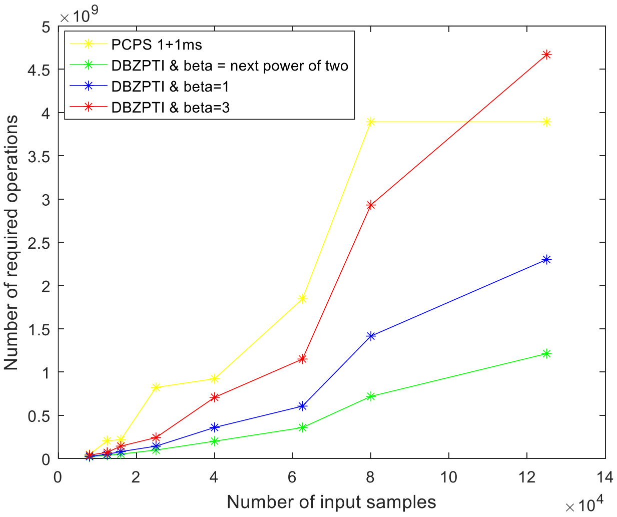


Figure 3-18 Comparison between DBZPTI with different values of beta and PCPS in term of number of operations

3.4 Proposed acquisition method: (DBZP + modified PCPS)

A 2Nspb points FFT is applied to compute the partial correlation of DBZP/DBZPTI. As it has been described before, the incoming signal is split into blocks of Nspb samples, and each two adjacent Nspb samples blocks are combined to form a double block of 2Nspb samples. The local code is partitioned in the same way but padded with Nspb zeros instead to produce 2Nspb samples block. Only the first half of correlation outputs will be preserved. It is worth noting that the usage of zero-padding to make local double block and then the half of the points at the correlation output are discarded are losses of computation efficiency. The proposed method is to compute DBZP partial correlations more efficiently. It is based on the ideal of modified PCPS algorithm, which discards fewer points at the correlation output.

3.4.1 Introduction to a modified PCPS algorithm

Initially, to overcome the problem introduced by data bit sign transition during the acquisition process, a well-known straightforward solution consists in performing a circular correlation using two consecutive periods of the incoming signal and one period of spreading local code with zero padding. In this way, there is always a complete period of the incoming signal, and thus a maximum correlation peak at the first part of correlation output, noted by y.

$$y = IFFT\left(\overline{FFT(h)}\,FFT(x)\right) \tag{41}$$

Therefore, it is enough to preserve only the first part of the result samples, y, and discards the second part of y. In order to improve the computational efficiency, the Modified PCPS algorithm is proposed as a new method to compute the circular correlation under the previous conditions.

The Modified PCPS algorithm consists in transforming the initial correlation vector into two sub-correlations of 3N/4 samples. This algorithm do not use all input points to perform the computation and the second part of the correlation output is different respect to the original result. A detail description and operation analysis can be found in (Jérôme Leclère, 2013)(Section 4). To summarise, it can be expressed by the following equation:

$$\begin{aligned} y_M &= IFFT\left(\overline{FFT(h_0)}FFT(x_o)\right) + IFFT\left(\overline{FFT(h_1)}FFT(x_1)\right) \\ &= IFFT\left(\overline{FFT(h_0)}FFT(x_o) + \overline{FFT(h_1)}FFT(x_1)\right) \end{aligned} \tag{42}$$

where

- $h_0 = \left[h[0]\,h[1]\,\ldots\,\,h\left[\frac{N}{4}-1\right]\,0\ldots\,\,0\right],\ \ \frac{N}{2}\ zeros\ padded$
- $h_1 = \left[h\left[\frac{N}{4}\right]\,h\left[\frac{N}{4}+1\right]\,\ldots\,\,h\left[\frac{N}{2}-1\right]\,0\ldots\,\,0\right],\ \ \frac{N}{2}\ zeros\ padded$
- $x_0 = \left[x[0]\,\,x[1]\,\ldots\,\,h\left[\frac{3N}{4}-1\right]\right]$
- $x_1 = \left[x\left[\frac{N}{4}\right]\,\,x\left[\frac{N}{4}+1\right]\,\ldots\,\,x[N-1]\right]$

The result signal y_M is a vector of $\left(\frac{3N}{4}-1\right)$ samples. Its first $\frac{N}{2}$ samples of y_M are equal to the first half samples of y, whereas the last $\left(\frac{N}{4}-1\right)$ samples of y_M are different than those of y. But it does not have consequences since these samples are discarded.

The algorithm computational complexity comparison:

It is assumed that the local code (in this case, vector h) has been previously calculated and stored. Then this algorithm require two FFTs and one IFFT of $\left(\frac{3N}{4}-1\right)$ points, whereas the traditional algorithm only require one FFT and one IFFT of N points.

Considering an FFT of N points requires $N/2\log(N)$ multiplications and $N\log(N)$ additions. Therefore, this algorithm requires $\frac{9}{8}N\log(\frac{3}{4}N)$ multiplications and $\frac{9}{4}N\log(\frac{3}{4}N)$ additions approximately, while the traditional algorithm requires $N\log(N)$ multiplications and $2N\log(N)$ additions. Thus, the modified PCPS algorithm needs more operations than the traditional method when the same length N is used. However, the FPGA hardware implementation requires the input length of FFT be the powers of two by using zero padding, and then the modified algorithm can be more efficient under certain input length condition.

Matlab simulation: computational complexity comparison between two methods with different input length

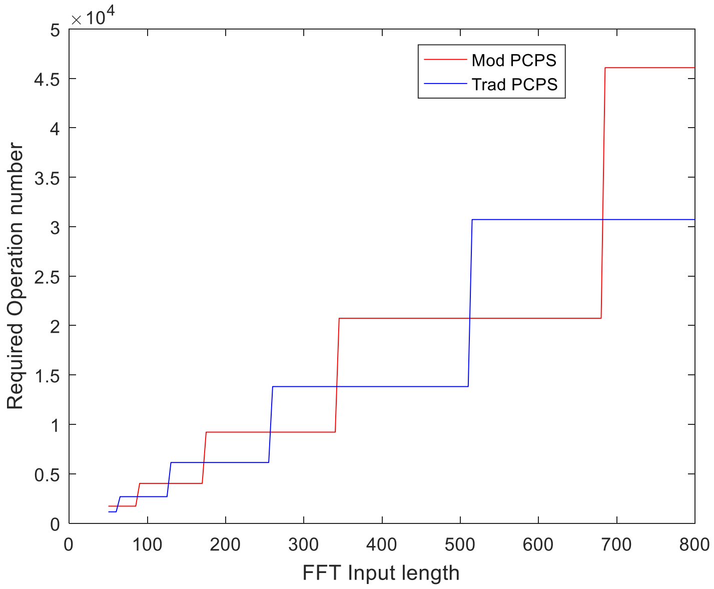


Figure 3-19: Comparison between traditional PCPS and modified PCPS in term of number of operations

As a result, if the input length of FFT (N) satisfy the following condition, the modified algorithm is more efficient:

$$N < \frac{2}{3}N_{padded} \tag{43}$$

where:

- N_{padded} is next power 2 of the N.

The combination of DBZP and modified PCPS algorithm is proposed in order to speed up the acquisition process.

For example, if the input size of the partial correlation is 625, the proposed method can reduce significantly the number of required operations respect to the DBZP method.

3.4.2 Implementation and Results

1. Generating the received signal with following characteristics:
   - superposition of six satellite signals with corresponding amplitude, Doppler frequency and code delay values:
     - ○ Amplitude values $A_i = [0.4810\ \ 0.2887\ \ 0.7066\ \ 0.2833\ \ 0.8008\ \ 0.9901]$
     - ○ Doppler shift $f_{d_i} = [-7363\ \ -318\ \ 9601\ \ 1645\ \ 8140\ \ 1787]$
     - ○ Code delay $\tau_i =\ [209\ \ 3366\ \ 599\ \ 818\ \ 2420\ 1997]$
   - Additional noise is white Gaussian noise

2. Suppose sampling frequency is 12.5MSamples/s and Doppler range to be search is [-10 kHz, 10 kHz], the coherent integration time is 1ms.
3. Applying the DBZP + modified PCPS method with following parameters definition:
   - ○ Number of blocks: 20
   - ○ Length of each block: 625 samples/block
   - ○ Doppler Resolution: 1 kHz

Result of simulation: five of six satellite signals are detected with

- codeDelay: [209 3366 600 2421 1997]
- Doppler: [-7000 0 10000 8000 2000]

As expected, all detected satellite signals have correct codeDelay values but with a certain doppler uncertainty less than 1 kHz.

3.4.3 Performance

Adquisition parameters		
Coherent integration time Sampling frequency Search frequency bandwidth N Nb Nspb Nr	1ms 12.5 MHz [-10 kHz, 10 kHz] 12500 20 625 20	2ms 12.5 MHz [-10 kHz, 10 kHz] 25000 40 625 20
DBZP	M: 6950080 A: 11461760 T: 39,262,080	M: 17500160 A: 28523520 T: 98,524,160
DBZP+modPCPS	M: 4472000 A: 6915200 T: 24,803,200	M: 12544000 A: 19430400 T: 69,606,400
Adquisition parameters		
Coherent integration time Sampling frequency Search frequency bandwidth N Nb Nspb Nr	5ms 12.5 MHz [-10 kHz, 10 kHz] 62500 100 625 20	10ms 12.5 MHz [-10 kHz, 10 kHz] 125000 200 625 20
DBZP	M: 63750400 A: 103308800	M: 215500800 A: 350617600

	T: 358,310,400	T: 1,212,620,800
DBZP+modPCPS	M: 51360000 A: 80576000 T: 286,016,000	M: 190720000 A: 305152000 T: 1.0680e+09

Table 3-9: Comparison between DBZP and DBZP + modified PCPS in term of number of operations

It can be observed in the table 3-9, the proposed method require less number of operations and less processing time than the traditional DBZP method.

By comparing the modified DBZP method, DBZP and PCPS algorithm, it can be observed by figure 3-20, the modified DBZP is the most efficient algorithm under the mentioned input size condition.

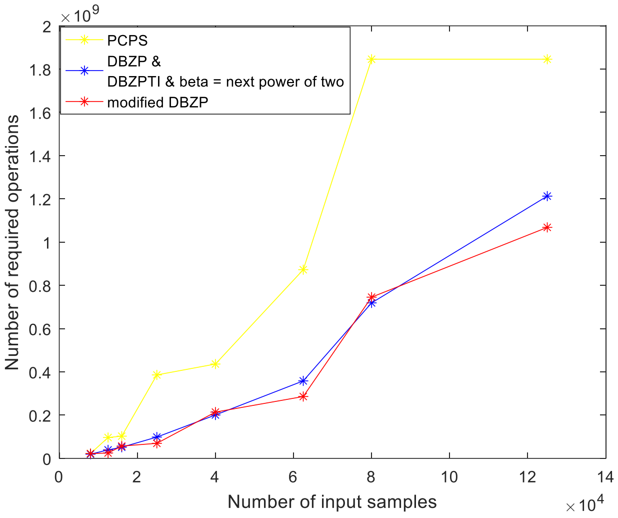


Figure 3-20: Comparison between variant of DBZP method with PCPS in term of number of operations

Chapter 4 : Conclusion and future work

The main objective of this project was to study algorithms to accelerate the acquisition process. From here four important algorithms have been introduced, which are 'Overlap and Save', 'Delay and Multiply', 'DBZP & DBZPTI' and 'Modified DBZP', and compared with the classical one, the parallel code phase search method. All four algorithms have been implemented in Matlab to prove their correct functionality by comparing the obtained simulation results with the result obtained by original acquisition method. Afterwards the efficiency of these acquisition algorithms is being analysed and compared with the original one in term of required number of operations. Hence this is convenient in comparing the processing time.

As it has been analysed in section 3.1, the combination of Overlap and Save with the classical acquisition algorithm increases the processing time. Since the time required by the combination of all blocks is greater than the time saved by taking a FFT with less input length. Therefore, the operating time increases with the number of blocks. Consequently, this method should be discarded unless the sub-block operations can be performed in parallel.

In the section 3.2, an interesting method, namely Delay and Multiply, has been analysed and its main objective is to parallelize both search domains in order to accelerate the acquisition process. However, the simulation result demonstrate this method cannot detect weak signals and requires a long coherent integration time to reduce the noise effect. This method is not suitable for the detection of satellite signals and requires a more in-depth study.

DBZP and DBZPTI have been introduced in section 3.3, both algorithms are quite similar and the simulation results show it is convenient to replace the original acquisition method by these algorithms in order to reduce the acquisition time. As shown in figure 3-18, the greater the number of input samples, the greater the benefits obtained by these algorithms.

Finally, a modified DBZP method has been proposed and analysed in section 3.4. It has been demonstrated under the condition (EQ. 43), this algorithm is more efficient than the previous ones.

In conclusion, the DBZP method and the variant of this method should be adopted at the acquisition stage in order to accelerate the acquisition operation.

For the future work it would be great to implement DBZP, DBZPTI and the proposed DBZP modified methods in the FPGA and test the functionality and performance with the entire software receiver. In addition, it would be interesting to explore more acquisition methods to further reduce the acquisition processing time.

Bibliography

Bernhard C. Geiger, C. V. (2013). *Influence of Doppler Bin Width on GPS Acquisition Probabilities.* IEEE.

Borio, D. (2008). *A Statistical Theory for GNSS Signal Acquisition .*

Borre, K. (2007). *A Software-Defined GPS and Galileo Receiver.* Boston : Birkhäuser Boston, c/o Springer Science+Business Media LLC, 233 Spring Street, New York, NY 10013, USA.

Boto, P. A. (2014). *Analysis and Development of Algorithms for Fast Acquisition of Modern GNSS Signals.* Lisboa.

David M. Lin, J. B. (1998). Acquisition Schemes for Software GPS Receiver. *ION GPS-98*, (pp. 317-325).

Duhamel, P. &. (1999). Fast Fourier Transforms: A Tutorial Review and a State of the Art. In V. K. Williams, *Digital Signal Processing Handbook.* Boca Raton: CRC Press LLC, 1999.

Encyclopedia. (2017, November 4). *Encyclopedia*. Retrieved from Overlap-Save method: https://en.wikipedia.org/wiki/Overlap%E2%80%93save_method

Encyclopedia. (2017, March 13). *Encyclopedia*. Retrieved from Overlap-add method: https://en.wikipedia.org/wiki/Overlap%E2%80%93add_method

Fernández, C. (2018). *http://gnss-sdr.org/*. Retrieved from http://gnss-sdr.org/my-first-fix/: http://gnss-sdr.org/my-first-fix/

Foucras, M. (2015). *Performance Analysis of Modernized GNSS Signals Acquisition.* Toulouse.

Jérôme Leclère, C. B.-A. (2013). *Modified parallel code-phase search for acquisition in presence of sign transition.* IEEE.

M. Foucras, O. J. (2014). *An efficient strategy for the acquisition of weak galileo E1 OS signals.*

Myriam Foucras, O. J. (2014, July/August). *Assessing the Performance of GNSS Signal Acquisition.* Retrieved from InsideGNSS: http://www.insidegnss.com/node/4097