A Hardware-Based Acquisition Engine For GNSS Receivers

by

Dana Jamal

A thesis submitted in partial fulfilment of the requirements of
London South Bank University for the degree of
Doctor of Philosophy

School of Engineering
Division of Electrical Electronic Engineering
London South Bank University
London, United Kingdom

Sep. 2021

Acknowledgements

I would like to express my sincere gratitude to all those that supported me to achieve this goal throughout my PhD journey.

My thanks to my supervisor Dr Perry Xiao for the continuous support of my PhD study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of my research and the writing of this thesis. Beside my supervisor I would like to also thank my second supervisor Dr Mohammad Ghavami for his support during this journey.

I would like to thank Dr Jan Safar for his contribution to record real data which is used for validation of the developed system.

Last but not least, I would like to thank my family: my wife and my two boys who have accompanied me in this journey and motivated me to step forward.

Abstract

The undergoing modernisation of global navigation satellite system (GNSS) has resulted in development of new navigation signals with longer spreading codes and new techniques to improve signal power and better multipath mitigation, such as binary offset carrier (BOC), this will improve the pseudorange calculation and enables more robust navigation. On the other side, this development requires receivers capable to process these new signals to deliver accurate positioning, navigation, and timing (PNT) solutions. To achieve this, the satellite acquisition process has to be improved to acquire the modernised signal in an efficient way. The double-block zero padding (DBZP) algorithm is commonly used for acquiring and detecting weak GNSS signals. It improves the acquisition method by reducing the number of operations in the block correlation used to determine the Doppler frequency and time of the received signal. However, 50% of the power consumption and time during the partial correlation process is lost because of performing correlation on doubled blocks (two periods) and maintaining the output from only the first block.

This work presents an innovative modelling method for the development of an efficient and stand-alone core based on the DBZP methodology in a field programmable gate array (FPGA) for the acquisition of GNSS signals. The main core consists of two components. The first component performs a partial correlation on the incoming signal without double blocking the incoming signal and zero padding the replica code. The second component performs discrete Fourier transform (DFT) on the output of the partial correlation results obtained from the first component. The core is designed without using any third-party fast Fourier transform (FFT) intellectual property (IP) cores, digital signal processor (DSP) blocks, or multipliers.

Performing partial correlation on the incoming signal without forming double blocks of the incoming signal and zero padding the replica code limits the computation burden and complexity, thus leading to a substantial improvement in the efficiency of the acquisition process in the hardware based GNSS receiver. The feasibility and performance of the proposed approach are investigated by developing a structural test bench containing the main core under test and a stimulus generator, simulated in ModelSim and the developed system is validated using real recorded Galileo E1 signal in FPGA.

Contents

List of Figures

List of Tables

List of symbols

A	Amplitude
A_{B1I}	Amplitude of BDS B1I signal
A_{B1Q}	Amplitude of BDS B1Q signal
A_{B3I}	Amplitude of BDS B3I signal
A_{B3Q}	Amplitude of BDS B3Q signal
$rf(\tau)$	Auto Correlation Function for a rectangular pulse
B_{AD}	Bandwidth of the ADC.
B_{RF}	Bandwidth of the signals of interest.
b_u	Clock bias error expressed in distance
b_{ut}	Clock bias error
c	speed of light
C_{B1I}	Ranging code of BDS B1I signal
C_{B1Q}	Ranging code of BDS B1Q signal
C_{B3I}	Ranging code of BDS B3I signal
C_{B3Q}	Ranging code of BDS B3Q signal
C_L^D	Galileo E5 lower-side band (data)code
C_L^p	Galileo E5 lower-side band (pilot)code
C_U^D	Galileo E5 upper-side band (data)code
C_U^p	Galileo E5 upper-side band (pilot)code
$c(t)$	Random sequence of pulses
$C_s(t)$	Complex sub-carrier
D	Random start time in interval $[0, T]$
D_{B1I}	Navigation data modulated on ranging code of BDS B1I signal
D_{B1Q}	Navigation data modulated on ranging code of BDS B1Q signal
D_{B3I}	Navigation data modulated on ranging code of BDS B3I signal

D_{B3Q}	Navigation data modulated on ranging code of BDS B3Q signal
$D_{(t)}$	Navigation data
$s_{E1-BC}(t)$	Galileo E1-BC composite signal
f	Frequency
f_1	BDS B1 carrier frequency
f_3	BDS B3 carrier frequency
f_b	Frequency step size (Doppler bins)
f_{E5}	Galileo E5 carrier frequency
$f_{C/A}$	The chipping rate of L1 C/A code and
f_D	Doppler frequency
f_{Dmax}	Maximum estimated Doppler frequency
f_{Dmin}	Minimum estimated Doppler frequency
f_{Dr}	Doppler shift due to receiver movement
f_{L1}	L1 carrier frequency
f_{Local}	Locally generated carrier frequency
f_{I5}	The chipping rate of I5 code
f_{IF}	Down-converted IF frequency
f_{im}	image frequency
f_{LO}	Local oscillator tune frequency
f_{RF}	RF carrier frequency
f_{Rs}	Sampling frequency
$f_{(t)}$	Fourier transform of a rectangular pulse
$f_{(t)}$	Rectangular pulse
$G^{even}_{AltBOC}(f)$	Power spectral density of even AltBOC modulation
$G^{odd}_{AltBOC}(f)$	Power spectral density of odd AltBOC modulation
G_{dBi}	the gain is expressed in a logarithmic scale
$G_{OS_D+OS_P}(f)$	Galileo E1 OS Composite power spectrum density (Data and Pilot)
M	Number of blocks in DBZP algorithm
N	Total number of samples across the coherent integration time
N_b	Number of samples per block in DBZP algorithm
N_{chip}	Number of chips in one PRN sequence
$p_{(t)}$	Unit amplitude pulse of duration T

$S_{xx}(f)$	PSD of the random binary sequence
$R_{CBOC(6,1,1/11)}(\tau)$	The ACF of $CBOC_{(6,1,1/11)}$
$R_{xx}(t_1, t_2)$	ACF for arbitrary values of t_1 and t_2
$S_{AltBOC}(t)$	Galileo E5 AltBoc signal
S^i_{B1}	BDS B1 signal
S^i_{B3}	BDS B3 signal
$SC_{BOC(1,1)}(t)$	BOC(1,1) sub-carrier
$SC_{BOC(6,1)}(t)$	BOC(6,1) sub-carrier
$S_{E1}(t)$	Galileo E1 signal
$S_{E5}(t)$	Galileo E5 signal
$S_{E6-CS}(t)$	Galileo E6 CS signal
T_C	coherent integration time
T_s	sub-carrier time period
$SC_{BOC(6,1,1/11,+)}(t)$	The in-phase composed sub-carrier
$SC_{BOC(6,1,1/11,-)}(t)$	The anti-phase composed sub-carrier
V_d	The velocity component causes the Doppler effect
V_{Dr}	The receiver velocity component causes the Doppler effect
V_s	GPS Satellite speed
$X_{Im}(k)$	The imaginary part of the DFT frequency output at the kth spectral point
$X_{(t)}$	spreading code
$X_{Re}(k)$	The real part of the DFT frequency output at the kth spectral point

Greek letters

Δf	Doppler frequency bin resolution
Δf_b	Frequency bin resolution
ϕ	longitude
λ	latitude
λ_{L1}	L1 C/A wave length
λ_{L5}	L5 wave length
ρ	Distances
τ	Time difference between two arbitrary values

List of Acronym

ACE-BOC	Asymmetric constant envelope BOC
ACF	Auto correlation function
ADC	Analog to digital converter
AltBOC	Alternative BOC
ARNS	Aeronautical radio navigation service
BC	Barker code
BDS	BeiDou navigation satellite system
BDT-UTC	BDS time-universal time coordinated
BGD	Broadcast group delay
BOC	Binary offset carrier
BPSK	Binary phase shift keying
C/A	Coarse/acquisition
CDMA	Code division multiple access
CL	Civil long
CM	Civil moderate
CPU	Central processing unit
CS	Control segment
DBZP	Double-block zero-padding
DC	Direct current
DDC	Digital down converter
DDR	Double data rate
DF	Dual-frequency
DFT	Discrete Fourier transform
DFT	Discrete Fourier transform

DS-CDMA	Direct sequence CDMA
DSP	Digital signal processor
DSSS	Direct sequence spread spectrum
EOP	Earth orientation parameters
ESA	European space agency
F/NAV	Free navigation
FDMA	Frequency division multiple access
FE	Front-end
FEC	Forward error correction
FFT	Fast Fourier transform
FMDBZP	Fast modified double-block zero padding
FOC	Full operational capability
FPGA	Field programmable gate array
FT	Fourier Transform
GCS	Ground control segment
GLONASS	Globalnaya navigatsionnaya sputnikova sistema
GMS	Ground mission segment
GNSS	Global navigation satellite system
GPS	Global positioning system
GSS	Galileo sensor stations
HOW	Hours of week
I/NAV	Integrity navigation
IC	Integrated circuit
ICAO	International organization for air safety
ICD	Interface control document
IF	Intermediate frequency
IFMDBZP	Improved fast modified double-block zero padding
IGS	International global navigation satellite system service
IGSO	Inclined geosynchronous orbit

IP	Intellectual property
ISC	Inter-signal correction
ITU	International telecommunications union
LFSR	Linear-feedback shift register
LHCP	Left-handed circularly polarised
LNA	Low noise amplifier
LNAV	Legacy navigation
LO	Local oscillator
LOS	Line of sight
MA	Multiple access
MCMF	Multi-constellation and multi-frequency
MCS	Master control station
MDBZP	Modified double-block zero padding
MEO	Medium earth orbit
MGEX	Multi-global navigation satellite system experimen
MS	Monitor stations
NH	Newman-Huffman
OS	Open service
OTA	Over-the-air
P	Precision
P(Y)	Encrypted precision code
PLL	Phase-locked loop
PNT	Positioning, navigation, and timing
PPM	Part per million
PR	Pseudo random
PRN	Pseudo-random noise
PRS	Public regulated service
PSK	Phase shift keying
QMBOC	Quadrature multiplexed BOC

QPSK	Quadrature phase shift keying
RDSS	Radio demonstration satellite service
RF	Radio frequency
RHCP	Right-handed circularly polarised
RNSS	Radio navigation satellite service
SAR	Search and rescue
SatHl	Satellite health
SAW	Surface acoustic wave
SDR	Software defined radio
SF	Single-frequency
SISRE	Signal-in-space range error
SoC	System on chip
SOW	Second of week
SPP	Single point positioning
SRAM	Static random access memory
ST	Standard accuracy
SWaP-C	Size, weight, power and cost
TDMA	Time division multiple acces
TOW	Time of week
TS/US	Time synchronisation/upload stations
TTFF	Time to first fix
UHF	Ultra-high frequency
ULS	Uplink stations
URAI	User range accuracy index
WN	Week number

Chapter 1

Introduction

1.1 Background and motivation

GNSS is the key technology for PNT because of its global availability, high accuracy (thanks to the onboard atomic clock), and has no direct cost to users. When four or more satellites are in view, a receiver can calculate the distance to each satellite by measuring the time delay between the signal transmission and reception. From this, a GNSS-embedded device can derive accurate time and its own position with metre-level accuracy, enabling navigation.

Currently, there are four systems with full operational capability (FOC), global positioning system (GPS) , globalnaya navigatsionnaya sputnikova sistema (GLONASS), Galileo, and BeiDou navigation satellite system (BDS). All four systems transmit encrypted signals for military use and open services for civil applications. GPS is an American system that has been open for civil applications since the 1990s. GLONASS is a Russian system that is developed approximately at the same time as the GPS. Galileo is the European union's global navigation satellite system providing a highly accurate, guaranteed global positioning service under civilian control, open for civil applications since 2020. BDS has been independently constructed and operated by China, providing high-accuracy PNT service to global users, and provides a global service since 2020.

Most navigation systems employ code division multiple access (CDMA) spread spectrum modulation, while GLONASS employs frequency division multiple access (FDMA) spread spectrum modulation which is an older technique. Receivers for CDMA technology are equally complex and diverse in their architecture.

The acquisition of GNSS signals based on signal processing is a complicated task, and this explains why the first generation of the GPS receivers had a long time to first fix (TTFF) . TTFF is considered to be one of the key performance indicators of GNSS receivers. TTFF is the time between the activation of a GNSS receiver and the first computation of a navigation solution from a sufficient number of satellites, and depends on several factors such as the algorithm used to acquire the signal, demodulation, navigation data decoding, and processing speed. Considering no assistance, the TTFF can be described as follows:

$$TTFF = T_{acq} + T_{track} + T_N + T_{PNT} \tag{1.1}$$

Where: T_{acq} is the acqusition time; T_{track} is the settling time for code and carrier tracking; T_N is the navigation data read time; T_{PNT} is time to compute the navigation solution. The time required to receive the required navigation message is constant, while the detection of

a sufficient number of satellites depends on the type of the input data such as pseudorange code length and sampling rate, deployed acquisition method and correlation process speed. Therefore, the acquisition process mainly affects TTFF.

The development of silicon chips in circuit technology such as FPGA, it is possible to implement any combination of digital processing. In addition, it can implement algorithms in a massively parallel method, which means that a substantial amount of data processing can be performed very quickly and efficiently. The FPGA can be used to process front-end (FE) intermediate frequency (IF) sample streams, such as down-converting to baseband, and acquiring and tracking GNSS satellites. It takes less than one second to detect a GPS L1 signal with an open and clear view of the sky, but this does not mean that the research on this topic is completed (Leclère, 2014). For example, it is always important to reduce power consumption and detect weak signals owing to the long distance and bad geometry.

The recent version of GNSS signals such as Galileo will lead to better performance because of its longer codes, higher chipping rate, and new modulation methods which have a much more stable correlation function, but at the same time will require more complex signal processing (Rodríguez and Ángel, 2008). Therefore, there is a need to find new methods to reduce the complexity of signal processing.

1.2 Aim and objectives

The aim of the study is to develop an effective and affordable (low-cost) hardware-based data acquisition system based on DBZP algorithm for GNSS receivers that acquire modernised GNSS signals in an efficient way.
The objectives of the study are:

- To investigate the utilisation of DBZP algorithm for data acquisition of modernised GNSS signals.
- To investigate methods of reducing signal acquisition time.
- To identify and use efficient and affordable (low-cost) techniques for development of the GNSS signal acquisition system.
- Develop an evaluation platform (test-bench) for simulation and assessment of the developed system.
- To investigate the feasibility of the DBZP algorithm implementation in hardware/FPGA.

1.3 Related works

Acquisition is the most time-consuming process in a receiver, and fast acquisition is required. Reducing the number of operations during acquisition increases the speed and reduces the energy requirements of commercial GNSS receivers. Several efforts have been made to address acquisition problems in GNSS receiver designs. The authors in (Van Nee and Coenen, 1991) proposed the first FFT-based acquisition. This technique uses the FFT

to compute all possible frequency bins (in frequency space) in one step, thereby eliminating the time-consuming code phase-shift process. Comparisons with existing systems have shown a theoretical reduction in acquisition time of approximately 2000 times.

The authors in (Lin and Tsui, 2000) introduced the parallel code phase search via the circular correlation to calculate the correlation results using FFT and IFFT operations. In (Akopian, 2001), two-dimensional FFTs for both the code phase and Doppler frequency are proposed, this method reduced the complexity to a considerable extent by reusing intermediate computations.

In (Lin and Tsui, 2003a) the DBZP algorithm is proposed. This algorithm improved the acquisition method by reducing the number of operations in the block correlation used to determine the Doppler frequency and time of the received GNSS signal.

Modern GNSS signals use equal spreading codes and navigation or secondary code bit periods, resulting in a potential bit sign transition in the code period of the received signal segments. To solve the data transition problems in the DBZP method, two consecutive data bit periods are used to ensure that there is at least one accumulation interval that does not contain data bit transitions. However, this method has high computational complexity.

To reduce the computational complexity, the authors in (Ziedan and Garrison, 2004) proposed modified double-block zero padding (MDBZP). In this method an approach is developed to compensate for the change in the code duration and relative delay, with only a small increase in the processing requirement. Simulations indicated that this algorithm can acquire signals below 15 dB-Hz. The problem of acquiring weak signals in presence of strong interfering signals is also addressed.

In (Heckler and Garrison, 2009) the fast modified double-block zero padding (FMDBZP) is proposed. This algorithm based on MDBZP algorithm eliminates the redundant FFT computations by considering possible data bit combinations and bit edge positions.

In (Zhang and Ghogho, 2010) improved fast modified double block zero padding (IFMDBZP) method is proposed. This algorithm aimes to reduce the computation complexity and save memory space further. Three methods are introduced to improve the FMDBZP algorithm. One consists of unlikely data bit combination path elimination during coherent integration steps. The second one perform unlikely code phase elimination during noncoherent integration steps. The third one perform unlikely bit edge elimination also during noncoherent integration steps. The good performance of the proposed improved FMDBZP algorithm is illustrated using both simulation and real data.

For weak signal acquisition, a non-coherent squaring detector is introduced in (Hurd et al., 1987), which increased the sensitivity of the receiver by eliminating the limitation of the navigation data bit sign. A differentially coherent combining detector is first proposed in (Chung, 1995). This technique increases the receiver sensitivity by partially mitigating the squaring loss problem which is a main issue in non-coherent squaring detectors.

Note: The above works have been published on the acquisition of GNSS signals using CDMA which has been performed through classical correlation or using a Fourier transform. Until beyond the writing of this thesis there are no researches available which studying the DBZP implementation on hardware/FPGA. Most of the researches which carried on DBZP are implemented in the software platform.

1.4 Contribution

This thesis has the following contributions:

- A state-of-the-art review of various acquisition methodologies.
- Design and development of an innovative efficient transition-insensitive hardware-based acquisition core capable of independently acquiring modernised satellite signals.
- Detailed design description and VHDL code for the proposed system in the FPGA.
- Detailed description for design and develop a test bench to simulate the developed system.
- MATLAB code to synthesise Galileo E1 signals and analysis of the collected data from simulation and real test.

1.5 Thesis outline

The performed research in this thesis is outlined as below:

- Chapter 1 provides an overview of the thesis scope, describing the motivation, challenges and the current state-of-the-art work that has been performed and the thesis breakdown structure.
- Chapter 2 describes the main idea behind the GNSS concept. It also provides an overview of the existing GNSS signals and their characteristics and limitations.
- Chapter 3 provides a description of the composed system segments and the four main global navigation systems which form GNSS. This chapter covers the allocated frequency band, the signal structure, including the PRN code, navigation data type, modulation scheme with the carrier, and power spectrum density.
- Chapter 4 provides an overview of GNSS user segments. This chapter focuses on the receiver architecture and the key parameters which indicate the receiver performance, such as antenna centre frequency and bandwidth, downconversion topologies in the frontend, and analog-to-digital conversion.
- Chapter 5 provides a description of GNSS signal acquisition process and the key parameters which affect this process including the different methods adopted in the acquisition of the GNSS signals.
- Chapter 6 provides the detailed design for building the proposed system in the FPGA and validating using real collected data.
- Chapter 7 concludes the dissertation, discussing the achievements and results of the thesis, and provides recommendations for future work.

Chapter 2

Introduction to GNSS

2.1 Concept of GNSS

GNSS positioning is based on trilateration. Trilateration is a method used to determine the location of a point (usually on the earth surface) using three or more satellites with known locations in 3 dimensional axis. To understand the concept in a simpler manner,

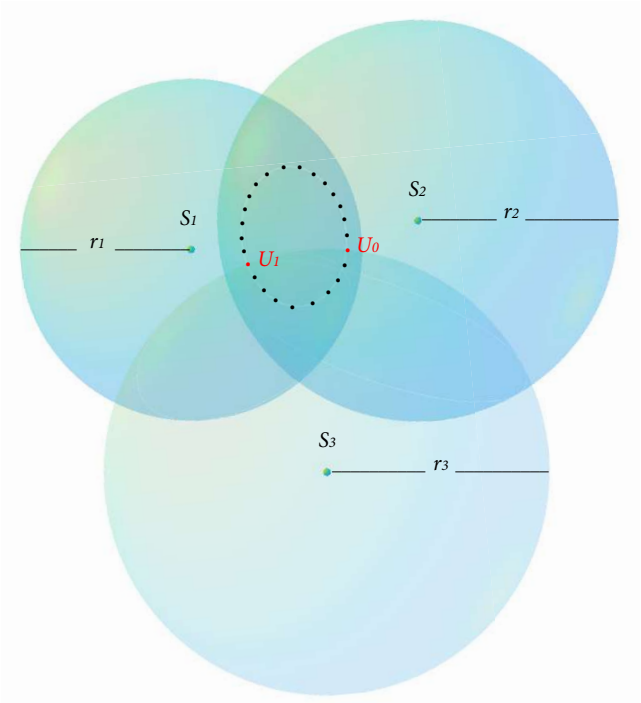


Figure 2.1: Three-dimensional user position.

it is assumed that the distance measured from the user to the satellite is very accurate, and there is no bias error due to the user clock.

In Figure 2.1, the user position U is on the three-dimensional axis (X_u, Y_u, Z_u). If the satellite position S_1 on the three-dimensional axis (X_1, Y_1, Z_1) and the distance r_1 from the user position are both known. The satellite forms a sphere with radius r_1 in three dimensions, which is centred at the satellite self. Therefore, the user location U can be any point outside the sphere which rounding satellite S_1 with radius r_1. Considering a second satellite with known location S_2 (X_2, Y_2, Z_2) and distance r_2 from the user

position, the sphere around satellites S_1 and S_2 intersect to form a circle (dotted circle). The possible user location is located somewhere on this circle. A third satellite with a known location S_3 (X_3, Y_3, Z_3) and distance r_3 from the user position brings down the user location solution to two possible solutions because the circle intersects the third sphere formed by third satellite S_3 to produce two points U_0 and U_1. Because there are three unknowns and three equations, the values of X_u, Y_u and Z_u can be determined from these equations. Theoretically, there should be two sets of solutions as they are second-order equations. To determine which point is the user position, an additional satellite (fourth) is needed, the sphere intersection of the fourth Satellite will bring down the user location solution to a single point (Tsui, 2005). The position of the satellite is determined from the navigation data transmitted by the satellite. The distance from the satellite to the user can be calculated by knowing the signal travel time between the user and the satellite. Therefore, the position of the user is determined.

In fact, the distance measured between the user and the satellite has an unknown bias because the user clock has a lower resolution compared with the satellite clock. Usually, the satellite deploys a very accurate clock, such as an atomic clock, while the user receiver is provided with a less accurate clock owing to cost and size factors. To calculate this bias error, an additional satellite is required. Therefore, five satellites are required to accurately determine the user position, four for a single point position and one for clock bias correction. The user position and bias error due to low receiver clock resolution can be measured by using only four satellites, three for measuring the distance and one for error correction. Using three satellites to measure the position provides two possible user positions, one of which is close to the earth's surface and the other is in space. Because the user position is usually close to the earth's surface, the second possibility can be ignored, this concludes that with four satellites the user position and the clock bias can be solved (Kaplan and Hegarty, 2017).

2.1.1 Basic calculation of user position

In Figure 2.1, assuming that the distance measured from the user to the satellite is very accurate, there are three known points which represent the satellite locations S_1, S_2 and S_3 at (X_1, Y_1, Z_1), (X_2, Y_2, Z_2) and (X_3, Y_3, Z_3), and an unknown user point U at (X_u, Y_u, Z_u). If the distances between the three known points to the unknown point can be measured simultaneously at a certain time instance as ρ_1, ρ_2, and ρ_3, these distances can be written as (Tsui, 2005)

$$\rho_1 = \sqrt{(X_1 - X_u)^2 + (Y_1 - Y_u)^2 + (Z_1 - Z_u)^2}$$
$$\rho_2 = \sqrt{(X_2 - X_u)^2 + (Y_2 - Y_u)^2 + (Z_2 - Z_u)^2}$$
$$\rho_3 = \sqrt{(X_3 - X_u)^2 + (Y_3 - Y_u)^2 + (Z_3 - Z_u)^2}$$

As explained before, there are three unknowns and three equations, there should be two sets of solutions as they are second-order equations. The above equations can be written in a simplified form as

$$\rho_i = \sqrt{(X_i - X_u)^2 + (Y_i - Y_u)^2 + (Z_i - Z_u)^2} \tag{2.1}$$

where $i = 1, 2$, and 3, and X_u, Y_u and Z_u are unknowns. The distance ρ_i and the positions of satellites X_i, Y_i and Z_i are known.

The user position calculated in the above equations is in a Cartesian coordinate system (Tsui, 2005). Usually, it is converted to a spherical system and represented by latitude, longitude, and altitude. The latitude of the earth is the angular distance of a position north or south of the earth's equator, which ranges from $-90°$to $90°$. The longitude is the angular distance east or west of the Greenwich meridian; it ranges from $-180°$ to $180°$ to the Greemwich meridian at $0°$. The altitude is the height in relation to sea level. Considering Earth as a perfect sphere, the user position can be calculated, as shown in Figure 2.2. The distance from the centre of the earth to the user is

$$r = \sqrt{X_u^2 + Y_u^2 + Z_u^2} \tag{2.2}$$

The latitude λ is

$$\lambda = tan^{-1}\left(\frac{Z_u}{\sqrt{X_u^2 + Y_u^2}}\right) \tag{2.3}$$

The longitude ϕ is

$$\phi = tan^{-1}\left(\frac{Y_u}{X_u}\right) \tag{2.4}$$

The altitude h assuming the earth is a perfect sphere is

$$h = r \tag{2.5}$$

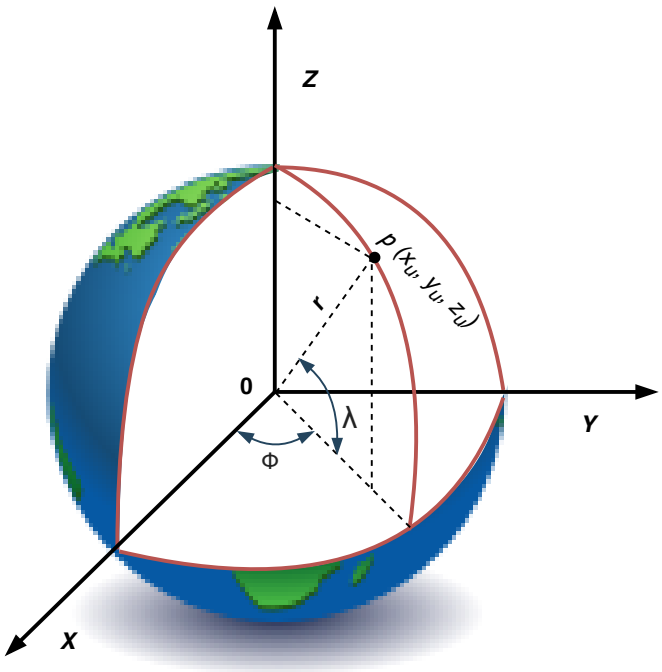


Figure 2.2: Three-dimensional user position.

The distance between the user and the satellite is obtained by calculating the range from the user position and known satellites. Each satellite transmits a signal with an encoded timestamp, this timestamp indicates the signal transmission time from the satellite

t_s. The receiver will receive the signal later at time t_u. The distance ρ between the user and the satellite can be calculated as follows:

$$\rho = c.(t_u - t_s) \tag{2.6}$$

where c is the speed of light.

In fact, it is very difficult to obtain the correct time from the satellite and the user because of satellite and user clock bias errors. Besides the clock error, there are other factors affecting the measurement, such as the satellite position error, tropospheric delay error, ionospheric delay error, and receiver measurement noise error. Some of these errors can be corrected; for example, the tropospheric delay can be modelled, and the ionospheric error can be corrected in a two-frequency receiver. These errors will cause inaccuracies in the user position. However, the user clock error cannot be corrected using the information received. Thus, it remains unknown (Tsui, 2005). Therefore, the equation should be modified as:

$$\rho = c.(t_u - t_s \pm b_{ut}) \tag{2.7}$$

where b_{ut} is the user clock bias error.
As a result, Equation 2.1 must be modified as

$$\rho_i = \sqrt{(X_i - X_u)^2 + (Y_i - Y_u)^2 + (Z_i - Z_u)^2} + b_u \tag{2.8}$$

where b_u is the user clock bias error expressed in distance.
For $i = 4$

$$\begin{aligned}
\rho_1 &= \sqrt{(X_1 - X_u)^2 + (Y_1 - Y_u)^2 + (Z_1 - Z_u)^2} \quad + b_u \\
\rho_2 &= \sqrt{(X_2 - X_u)^2 + (Y_2 - Y_u)^2 + (Z_2 - Z_u)^2} \quad + b_u \\
\rho_3 &= \sqrt{(X_3 - X_u)^2 + (Y_3 - Y_u)^2 + (Z_3 - Z_u)^2} \quad + b_u \\
\rho_4 &= \sqrt{(X_4 - X_u)^2 + (Y_4 - Y_u)^2 + (Z_4 - Z_u)^2} \quad + b_u
\end{aligned}$$

Therefore, to calculate the user position, there is a need for at least four satellites, three for the ranging, and one for the user clock bias correction. Because of this error, the distance obtained by computing the range from unknown user position and known satellite position plus the distance equivalent of the user clock error, is called the "Pseudo Range." This statement reflects the fact that this range is not exact because it includes the receiver clock bias error (Doberstein, 2011).

2.2 GNSS signal structure

The GNSS signals are combined of three components as described in the following subsections.

2.2.1 Carrier

The GNSS signals are transmitted on radio frequencies in the UHF band, more specifically the L band which covers the frequency band from 1 GHz to 2 GHz. In the L band, several sub-frequency bands have been allocated and made available by the international telecommunications union (ITU). Figure 2.3 illustrates the available frequency bands recorded at the ITU. The L band is chosen for satellite navigation because of several advantages

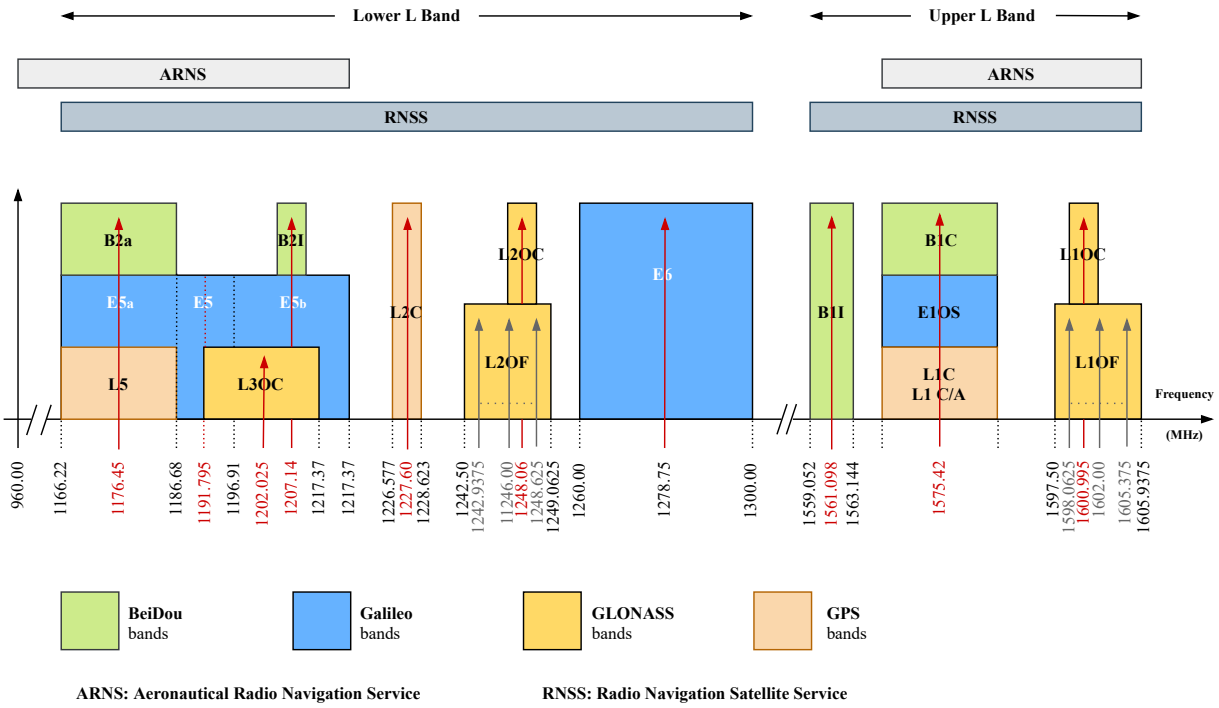


Figure 2.3: GNSS freuency bands as defined by International Telecommunications Union.

(Teunissen and Montenbruck, 2017):

- Acceptable propagation conditions, especially moderate attenuation and impact of atmospheric affects, rain, etc.
- Enable measurements of adequate precision.
- Allow for reasonably simple user equipment.
- The antenna size for the L-band signals is limited.
- A large variety of mature hardware components are available at low cost.

As shown in Figure 2.3,two areas have been established: the upper L-band (1559-1610MHZ) and the lower L-band (1164–1300 MHz) to be utilised for GNSS. These two areas are called the radio navigation satellite service (RNSS) band. A portion of RNSS

is protected against interference; it is used for safety-critical applications such as aeronautical applications, and is called the aeronautical radio navigation service (ARNS) . The GNSS uses right-handed circularly polarised (RHCP) waves for transmitted signals. Circular polarisation is preferred because, unlike the linearly polarised signal, it does not change in polarisation while travelling through the ionosphere due to Earth's magnetic field. A second advantage is that it allows the receiving antenna to discriminate between the direct signal from the satellite and strong multipath signals arriving at the antenna from the underside after reflection. The polarisation of the reflected signal is changed to left-handed circularly polarised (LHCP) , thus allowing the antenna to discriminate it (Rao et al., 2013).

2.2.2 Multiple access

Most satellites transmit navigation signals using the available transmission channel allocated to the GNSS. This is between 1 and 2 GHz and is called the L band in the ultra-high frequency (UHF) band. To utilise this band in an efficient way, multiple access (MA) techniques is deployed. There are three fundamental MA techniques.

- Time division multiple access (TDMA).
- Frequency division multiple access (FDMA).
- Code division multiple access (CDMA).

In the TDMA technique, each channel user utilises the entire available bandwidth, but in different time slots. For example, in Figure 2.4a, three users transmit signals using the same channel, all three utilising the full available bandwidth but transmitting at different times t_1, t_2 and t_3. In the FDMA technique, all channel users transmit at the same time,

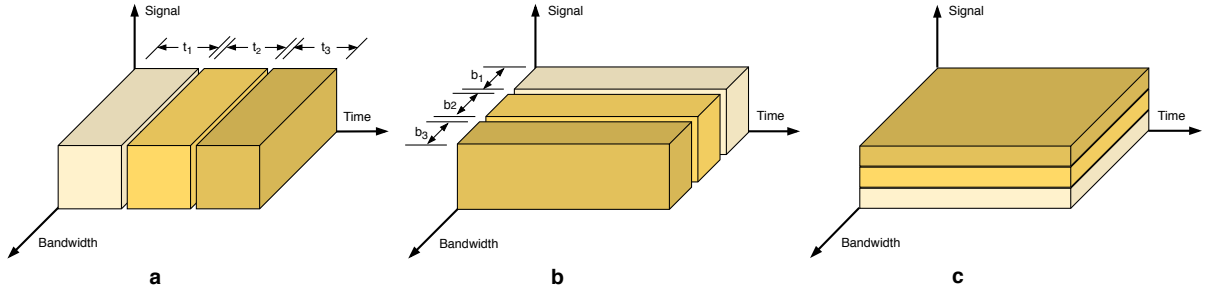


Figure 2.4: Multiple access schemes (a: TDMA, b: FDMA and c: CDMA).

but they transmit at different frequencies and utilise only a part of the full available channel bandwidth. For example, in Figure 2.4b, three users transmit signals using the same channel; all three transmit signals at the same time, but each user utilises only one-third of the full available bandwidths b_1, b_2 and b_3. The disadvantage of FDMA is crosstalk, which can cause interference between frequencies and interrupt the transmission; therefore, guard bands are used between the adjacent signal spectra to minimise crosstalk between the channels which leads to a waste of capacity. Another disadvantage is that it requires radio frequency (RF) filters to meet the stringent adjacent channel rejection specifications. This increases the cost of the system. In the CDMA technique, all channel users transmit at the same time and utilise the full available channel bandwidth, as shown in Figure 2.4c.

There are important advantages of CDMA: the CDMA does not require synchronisation, and a greater number of users can share the same bandwidth and less interference between the user channels owing to the allocated code for each user's efficient utilisation of the channel bandwidth. Most GNSSs employ a direct sequence CDMA (DS-CDMA). The different satellites transmit their signals simultaneously in the same frequency band at the same time. Each satellite uses a different code to transmit its signals. For example all GPS satellites are transmitting there navigation signal utilising the same L1 frequency. The satellites are dedifferentiated from each other by assigning a unique gold sequence to each one. These sequences have a very good auto correlation function and code isolation property. Spectral separation between different GNSSs in the same frequency band can be achieved using different modulation schemes. The GLONASS system uses FDMA, where each visible satellite uses a different frequency slot and uses a code sequence for synchronisation and channel parameter estimation purposes (Teunissen and Montenbruck, 2017).

2.2.3 Spreading codes

In the CDMA technique, spreading codes are deployed to differentiate channel users (satellites) as they transmit at the same time and share the same available transmission bandwidth. These spreading sequences are usually called pseudo-random (PR) binary sequences. While such deterministically generated sequences can never be truly random, their design ensures well-defined properties associated with randomness. These sequences are therefore called PR sequences (Golomb and Gong, 2005). A PR binary sequence deployed in the CDMA can be considered as a sequence of rectangular pulses.

2.2.4 Rectangular pulse

To characterise a single rectangular pulse, consider a pulse $f_{(t)}$ with amplitude A and pulse duration T as shown in Figure 2.5. The rectangular pulse $f_{(t)}$ can be expressed as

$$f_{(t)} = \begin{cases} A, & -T/2 \leq t \leq T/2 \\ 0, & \text{otherwise} \end{cases} \tag{2.9}$$

If the frequency f in Hz and $\omega = 2\pi f$
The Fourier transform of $f_{(t)}$ (Haykin, 2008) is

$$\begin{aligned} F_{(f)} &= \int_{-T/2}^{T/2} A e^{(-j2\pi ft)}\, \mathrm{d}t \\ F_{(f)} &= AT\left(\frac{sin(\pi fT)}{\pi fT}\right) \\ F_{(f)} &= ATsinc\,(fT) \end{aligned} \tag{2.10}$$

The Magnitude spectrum $|F(f)|$ of rectangular pulse is shown in Figure 2.6. The spectrum has zero-crossing occurs at $f = \pm 1/T, \pm 2/T, \pm 3/T, \pm 4/T, \ldots$. As the pulse duration T is decreased, the first zero-crossing moves up in frequency.

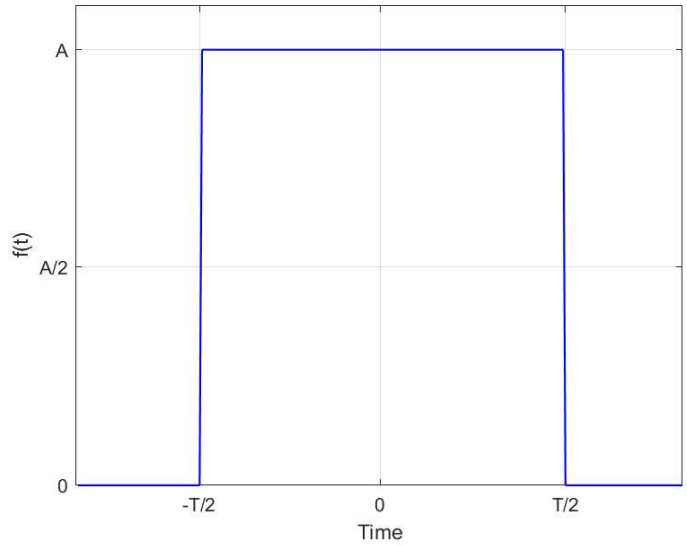


Figure 2.5: Rectangular pulse

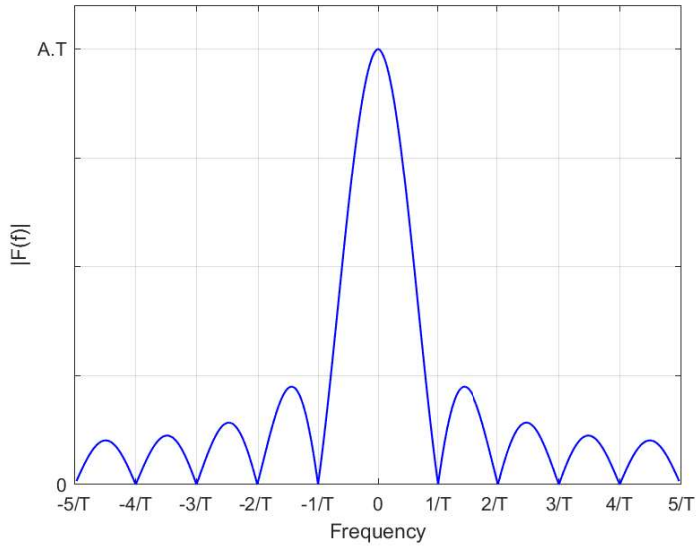


Figure 2.6: Spectrum of rectangular pulse.

As illustrated in Figure 2.7, the auto correlation function (ACF) $rf(\tau)$ for a rectangular pulse has a triangular waveform (Borre et al., 2007).

$$r_{f(\tau)} = \begin{cases} T\left(1 - \frac{|\tau|}{T}\right), & for|\tau| \leq T \\ 0, & \text{otherwise} \end{cases} \tag{2.11}$$

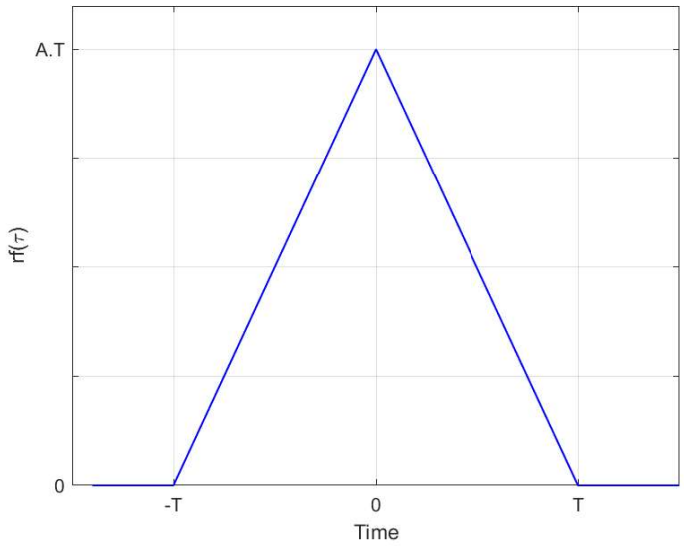


Figure 2.7: Autocorrelation function of the rectangular pulse

2.2.5 Random sequence of pulses

The random binary waveforms shown in Figure 2.8 used in data communication systems are modelled by a random sequence of pulses with the following properties (Shanmugan and Breipohl, 1988):

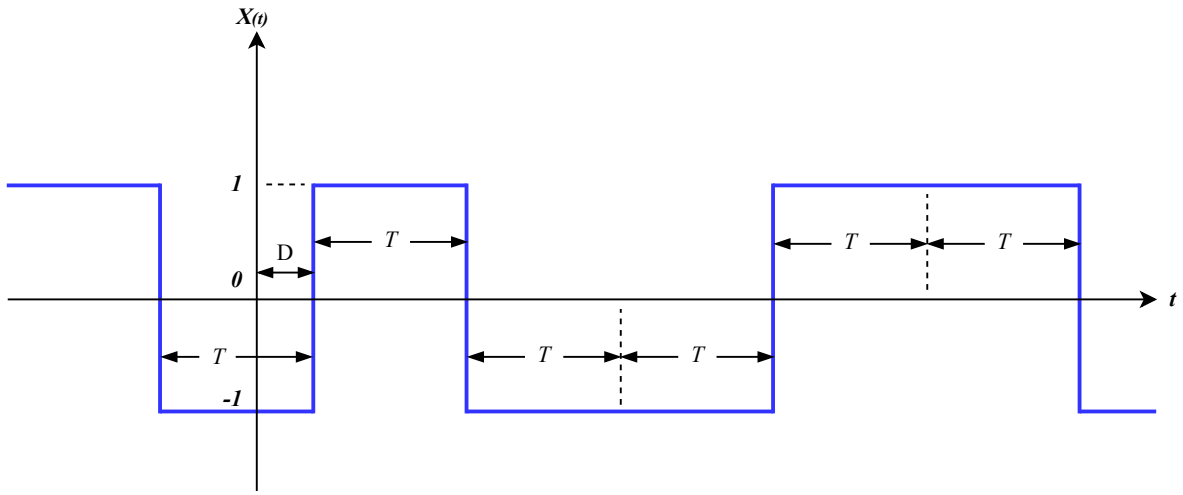


Figure 2.8: Random binary waveform

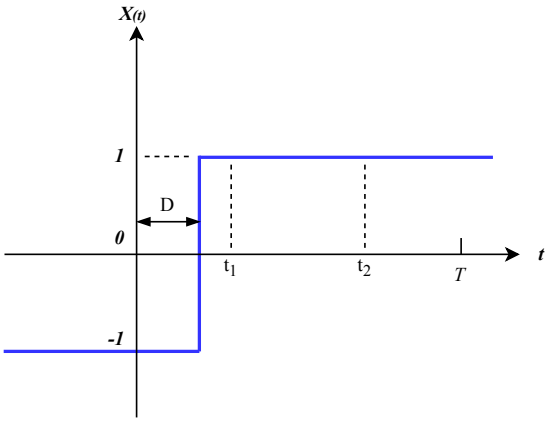
- Each pulse has a rectangular shape with a fixed duration of T and a random amplitude of ±1.
- Pulse amplitudes are equally likely to be ±1 (mean=0 and variance=1).
- All pulse amplitudes are statistically independent.
- The start times of the pulse sequences are arbitrary; that is, the starting time of the first pulse following t = 0 is equally likely to be any value between 0 and T.

The random binary waveform (random sequence of pulses) can be expressed as

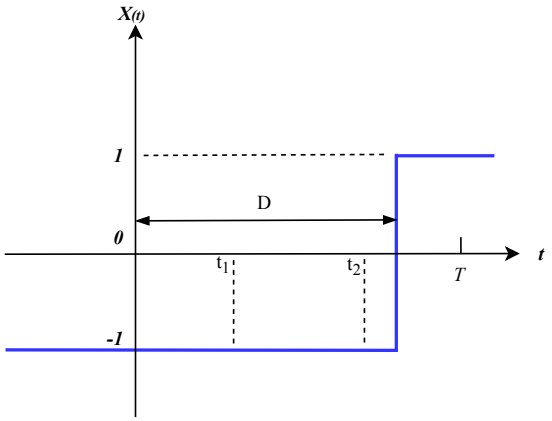
$$X_{(t)} = \sum_{k=-\infty}^{\infty} A_k \; p(t - KT - D) \tag{2.12}$$

where
$p_{(t)}$ is a unit amplitude pulse of duration T,
A_k is K^{th} binary random variable
and D is the random start time in interval $[0, T]$

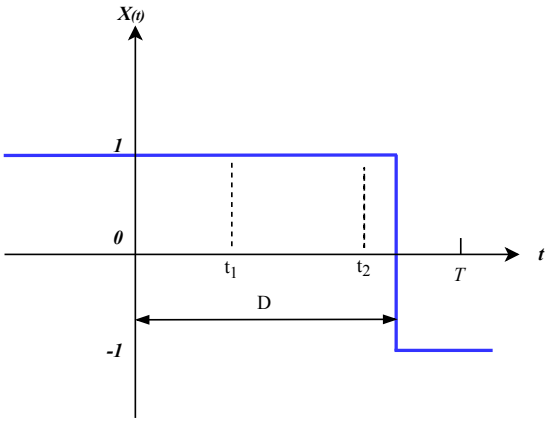
The ACF of the random binary waveform can be calculated by choosing two points in time t_1 and t_2 such that $0 < t_1 < t_2 < T$. From Figure 2.9 it is seen that:



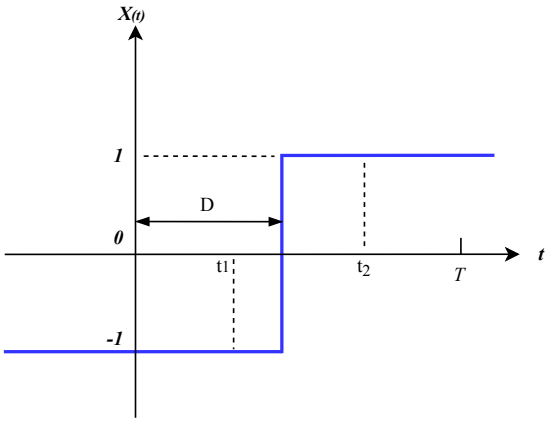
(a) $0 < D < t_1$
t_1 and t_2 belong to the same pulse interval



(b) $t_2 < D < T$
t_1 and t_2 belong to the same pulse interval



(c) $t_2 < D < T$
t_1 and t_2 belong to the same pulse interval



(d) $t_1 < D < t_1$
t_1 and t_2 belong to different pulse interval

Figure 2.9: Calculating ACF of a random binary waveform

- When $0 < D < t_1$ (2.9a) or $t_2 < D < T$(2.9b), t_1 and t_2 lie in the same pulse interval and product of the pulse amplitudes $X(t_1)X(t_2) = 1$.
- When $t_1 < D < t_2$, t_1 and t_2 lie in different pulse intervals and the product of the pulse amplitudes $X(t_1)X(t_2)$ has a value $+1$ or -1 with equal probability.

The ACF of the random binary waveform can be expressed as

$$X_{(t1)}\ X_{(t2)} = \begin{cases} 1, & if\ 0 < D < t_1 \quad or \quad t_2 < D < T \\ \pm 1, & if\ t_1 < D < t_2 \end{cases} \tag{2.13}$$

The general ACF equation for arbitrary values of t_1 and t_2 can be written as (Shanmugan and Breipohl, 1988)

$$R_{xx}(t_1, t_2) = \begin{cases} 1 - \dfrac{|t_2 - t_1|}{T}, & |t_2 - t_1| < T \\ 0, & \text{otherwise} \end{cases} \tag{2.14}$$

If $(t_2 - t_1)$ denoted as τ, equation 2.14 can be written as

$$R_{xx}(\tau) = \begin{cases} 1 - \dfrac{|\tau|}{T}, & |\tau| < T \\ 0, & \text{otherwise} \end{cases} \tag{2.15}$$

For a random binary waveform with amplitude $\pm A$ equation 2.15 can be written as (Haykin, 2008)

$$R_{xx}(\tau) = \begin{cases} A^2 \left(1 - \dfrac{|\tau|}{T}\right), & |\tau| < T \\ 0, & \text{otherwise} \end{cases} \tag{2.16}$$

Equation 2.16 shows that the ACF of a random binary waveform is a function of the time difference τ and has a triangular waveform with maximum amplitude of A^2 at $\tau = 0$ as shown in Figure2.10. The PSD of the random binary sequence can be obtained using the

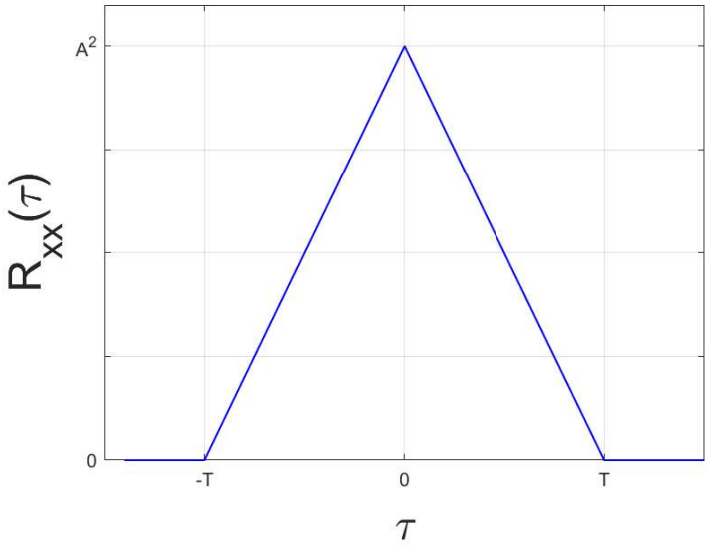


Figure 2.10: Auto correlation function of random binary sequence

Fourier transform of the ACF (Shanmugan and Breipohl, 1988) (Haykin, 2008).

$$S_{xx}(f) = \int_{-T}^{T} A^2 \left(1 - \frac{|\tau|}{T}\right) e^{(-j2\pi f\tau)}\, \mathrm{d}\tau$$

$$S_{xx}(f) = A^2 T\ \frac{sin^2\ \pi fT}{(\pi fT)^2}$$

$$S_{xx}(f) = A^2 T \cdot\ sinc^2(fT) \tag{2.17}$$

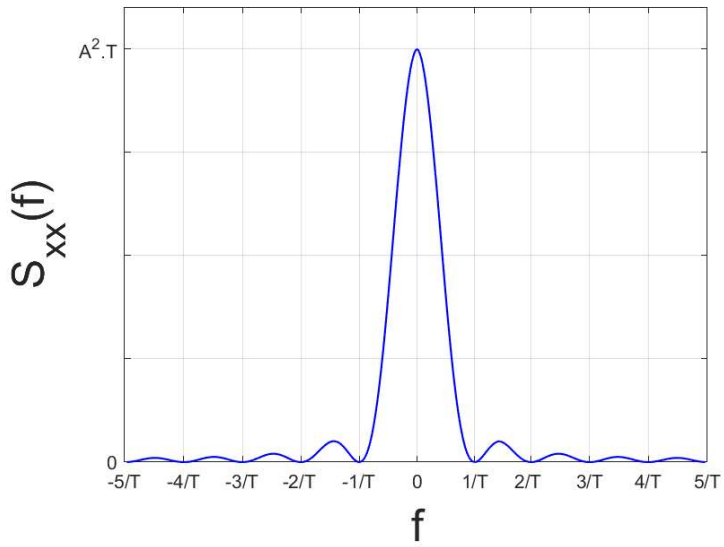


Figure 2.11: Power spectral density of random binary sequence.

Equation 2.17 shows that the PSD of a random binary waveform has a sinc function waveform with a maximum amplitude of A^2T at $f = 0$ as shown in Figure 2.11.
The main lobe of the PSD of a random binary sequence extends from $-\frac{1}{T}$ to $\frac{1}{T}$ Hz, and 90 percent of the signal power is included in the main lobe.

2.2.6 Navigation data

The GNSS signal carries navigation data that the user needs to solve for PNT. These data provide information about the satellite orbits, clock correction, time of week, and other useful information for positioning. Navigation data are uploaded to the satellites from the control segment based on different points on earth. The navigation data are transmitted in with a slow bit rate (between 50 and 250 bit per second) stream, modulo-2 added to the spreading code, and modulated on to a sinusoidal carrier with a constant frequency. For example, the navigation data bits of GPS are formatted into 30-bit words, and the words are grouped into subframes of 10 words that are 300 bits in length and 6 seconds in duration. Frames consist of five subframes of 1500 bits and 30 s in duration (Parkinson et al., 1996).

2.2.7 Modulation scheme

In the CDMA technique, the navigation data signal (in binary form) is modulo-2 added to the spreading codes, converted from unipolar to bipolar (from 0's and 1's to -1 and +1) and transmitted over the allocated satellite channels in L-Band by modulating the bipolar signal onto a sinusoidal carrier wave. The first generation of GNSS systems such as GPS deploying phase shift keying (PSK), while GLONASS is deploying PSK and FDMA. Modernised systems such as Galileo, Beido, and new GPS generation deploying beside PSK, the BOC technique.

2.2.7.1 Phase shift keying

The first generation of the satellites deploy PSK modulation technique. The PSK modulation corresponds to keying (changing) the carrier phase according to the binary modulated

data values. The most commonly used PSK modulation types, are either binary phase shift keying (BPSK) or quadrature phase shift keying (QPSK). In general, the two modulation types are denoted as $BPSK_{(m)}/QPSK_{(m)}$ which indicates the modulation of the navigation data signal with a spreading code rate of $m \times 1.023$ Mega chip per second. The BPSK is a two-phase modulation technique. In this the 0's and 1's in a binary signal are represented by two different phase states in the carrier signal. The binary data is represented with a bipolar code and then modulated with the carrier. The BPSK is described by (Haykin, 2008):

$$S_1(t) = A_c\ cos(2\pi f_c t) \qquad \text{for symbol 1} \tag{2.18}$$

$$S_0(t) = -A_c\ cos(2\pi f_c t) \qquad \text{for symbol 0}$$

$$S_0(t) = A_c\ cos(2\pi f_c t + \pi) \qquad \text{for symbol 0} \tag{2.19}$$

where
$S_1(t)$ denotes the carrier signal when information bit 1 is transmitted,
$S_0(t)$ stands for the carrier signal when information bit 0 is transmitted,
A_c is the amplitude of sinusoidal signal,
f_c is the carrier frequency (Hz),
t is the time in seconds.

QPSK is a quad-phase modulation technique. In this case, two binary signals are simultaneously modulated onto the carrier. Four different possible values from the two bits are represented by four different phase states in the carrier signal. For example, the four possible pairs of bits 10, 00, 01, and 11 can be represented as (Haykin, 2008):

$$S_0(t) = A_c\ cos(2\pi f_c t + \frac{\pi}{4}), \qquad \text{for bit pair 11} \tag{2.20}$$

$$S_1(t) = A_c\ cos(2\pi f_c t + \frac{3\pi}{4}), \qquad \text{for bit pair 01} \tag{2.21}$$

$$S_2(t) = A_c\ cos(2\pi f_c t + \frac{5\pi}{4}), \qquad \text{for bit pair 00} \tag{2.22}$$

$$S_3(t) = A_c\ cos(2\pi f_c t + \frac{7\pi}{4}), \qquad \text{for bit pair 10} \tag{2.23}$$

where, $S_0(t)$, $S_1(t)$, $S_2(t)$ and $S_3(t)$ denote the carrier signals when information bit pairs 11, 01, 00 and 10 are transmitted.
A QPSK system can be seen as a two-BPSK system operating in parallel, with two carrier waves in the phase quadrature. GPS L1 signal is a QPSK modulation, the C/A code is modulated into in-phase while the P(Y) code is into quadrature-phase. Galileo E1 signal is another QPSK modulation example, the E1 open service is modulated into in-phase while the E1A is modulated into quadrature-phase.

2.2.8 Binary offset carrier

The binary offset carrier (BOC) is another spread spectrum technique adopted by a new generation of GNSS satellites. BOC modulation provides flexibility in shifting the signal power around the main carrier in the allocated band to reduce interference with other signals.

In BOC modulation, each single rectangular pulse, as illustrated in Figure 2.12a of the binary sequence, is formed as a bipolar sine (as illustrated in Figure 2.12c and 2.12e) or cos (as illustrated in Figure 2.12d and 2.12f) square wave with a specific number of

cycles determined by the BOC order. BOC is the product of a random binary waveform (random sequence of pulses) in a bi-polar form (ones and zeros) with a sin-or cos-shaped square wave sequence. The sine shaped square wave term is usually called a sine-phased, and the cos shaped square wave term is usually called a cos-phased sub-carrier.
The random binary waveform (random sequence of pulses) $c(t)$ can be expressed as

$$c(t) = \sum_{k} c_k \; h(t - KT_c) \tag{2.24}$$

where
c_k is K^{th} binary random variable
$h_{(t)}$ is the unit amplitude pulse of duration T_c in bipolar form.
The sin-phased and co-phased sub-carriers can be written as

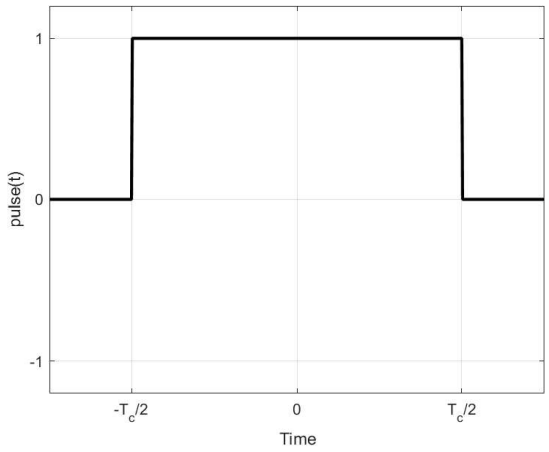
$$s_{(sin)}(t) = sgn\left[\sin(2\pi n_s f_r t)\right] \tag{2.25}$$
$$s_{(cos)}(t) = sgn\left[\cos(2\pi n_s f_r t)\right] \tag{2.26}$$

where $s_{(sin)}(t)$ and $s_{(cos)}(t)$ define the sin-phased and cos-phased sub-carrier respectively, n_c is the chip rate and n_s the sub-carrier rate and f_r is the reference frequency. For GNSS signals, for example, for GPS (Navstar, 2013) or the European Galileo system (Union, 2010), $f_r = 1.023MHz$.

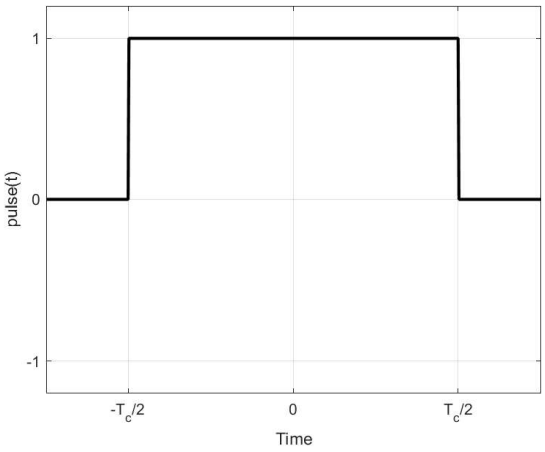
In general, BOC signals are denoted as $BOC(n_s, n_c)$ or $BOC_{cos}(n_s, n_c)$ for the BOC signals with sine or cosine-phased sub-carriers, respectively (Betz, 2001), and are given as

$$x_{BOC_{sin}(t)} = c(t) \cdot sgn\left[\sin(2\pi n_s f_r t)\right] \tag{2.27}$$
$$x_{BOC_{cos}(t)} = c(t) \cdot sgn\left[\cos(2\pi n_s f_r t)\right] \tag{2.28}$$

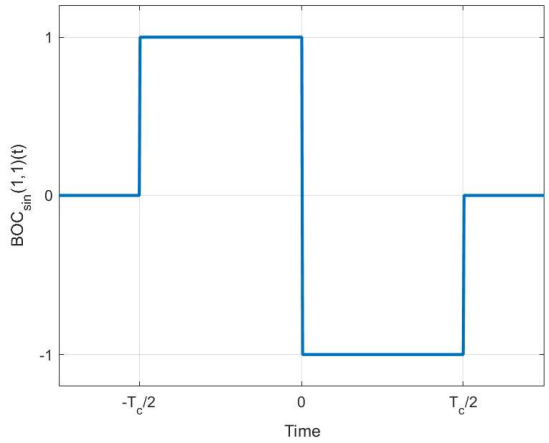
Figures 2.12c and 2.12d illustrate the chip pulse shapes for a $BOC_{sin}(1,1)$ signal and a $BOC_{cos}(1,1)$ signal respectively, one chip is formed with one sub-carrier cycle. When the sub-carrier rate is increased to six cycles, i.e. $n_s = 6$, the chip pulse shape is formed with six subcarrier cycles for $BOC_{sin}(6,1)$ and $BOC_{cos}(6,1)$ as illustrated in Figures 2.12e and 2.12f.



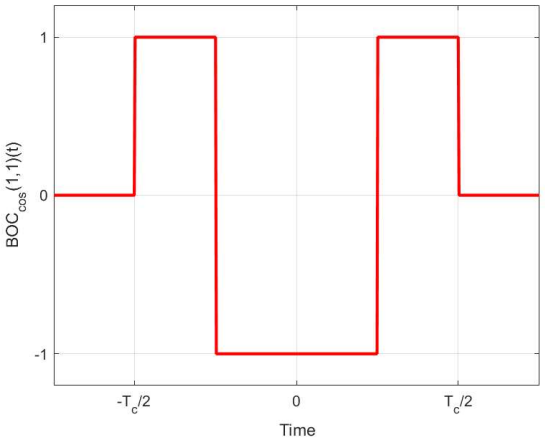
(a) single chip pulse



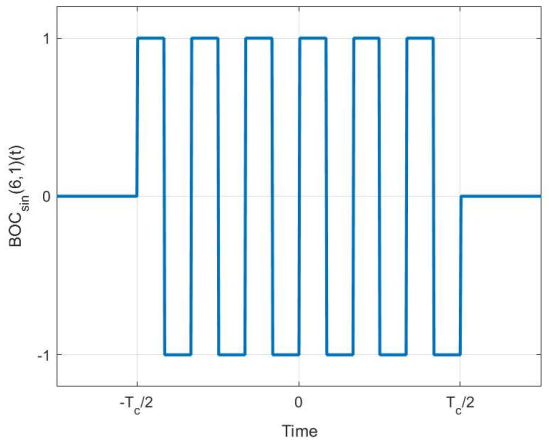
(b) single chip pulse



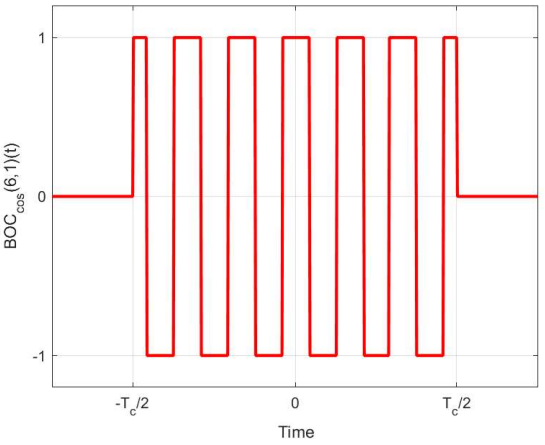
(c) chip pulse shape for $BOC_{sin}(1,1)$ signal



(d) chip pulse shape for $BOC_{cos}(1,1)$ signal



(e) chip pulse shape for $BOC_{sin}(6,1)$ signal



(f) chip pulse shape for $BOC_{cos}(6,1)$ signal

Figure 2.12: A chip pulse modulated with *sin* and *cos−phased* $BOC(1,1)$ and $BOC(6,1)$

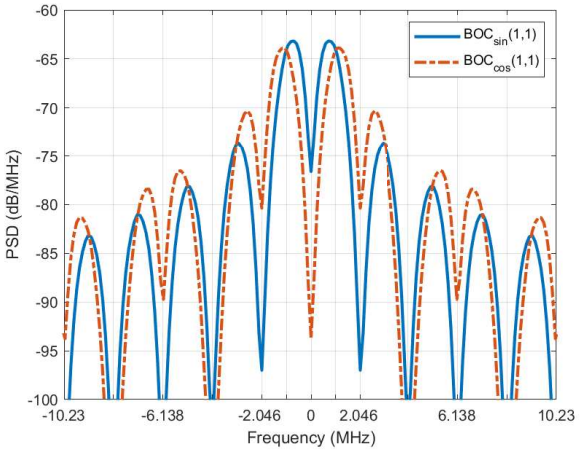
BOC signals are applied in GNSS to fulfil the spectral separation requirements between different non-interoperable signals of different GNSSs and to enhance synchronisation performance. By multiplying the pulse chips with a sin-phased or cos-phased sub-carrier the spectrum of the signal is divided into two parts; therefore, BOC modulation is also known as split-spectrum modulation (Betz, 2001). The general formula for $BOC_{sin}(f_s, f_c)$

and $BOC_{cos}(f_s, f_c)$ can be written as (Rodríguez and Ángel, 2008):

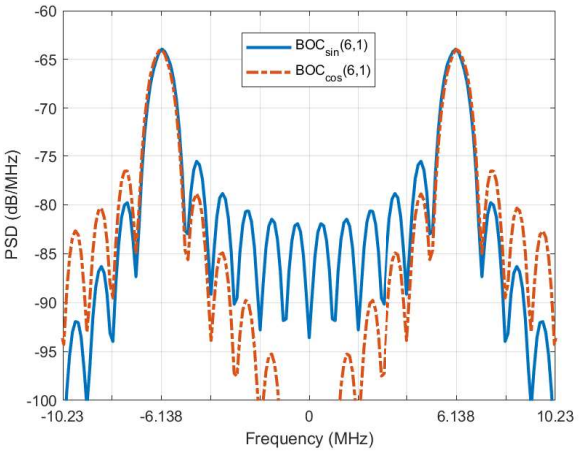
$$G_{BOC_{sin}(f_s,f_c)} = f_c \frac{\sin^2(\frac{\pi f}{f_c})}{(\pi f)^2} \tan^2\left(\frac{\pi f}{2f_s}\right) = f_c \left[\frac{\sin(\frac{\pi f}{f_c})\sin(\frac{\pi f}{2f_s})}{\pi f cos\left(\frac{\pi f}{2f_s}\right)}\right]^2 \tag{2.29}$$

$$G_{BOC_{cos}(f_s,f_c)} = 4f_c \frac{\cos^2(\frac{\pi f}{f_c})\sin^4(\frac{\pi f}{4f_s})}{(\pi f)^2 cos^2\left(\frac{\pi f}{2f_s}\right)} = f_c \left[\frac{2\cos(\frac{\pi f}{f_c})\sin^2(\frac{\pi f}{4f_s})}{\pi f cos\left(\frac{\pi f}{2f_s}\right)}\right]^2 \tag{2.30}$$

The PSDs of a $BOC_{sin}(1,1)$ and a $BOC_{cos}(1,1)$ signals are illustrated in Figure 2.13a. The larger the sub-carrier rate n_s is chosen, the further the two main lobes of the split spectrum signal are shifted apart. This effect can be seen in BOC(6,1) as illustrated in Figure 2.13b when compared with the BOC(1,1) modulation.



(a) PSD of $BOC_{sin}(1,1)$ and $BOC_{cos}(1,1)$.

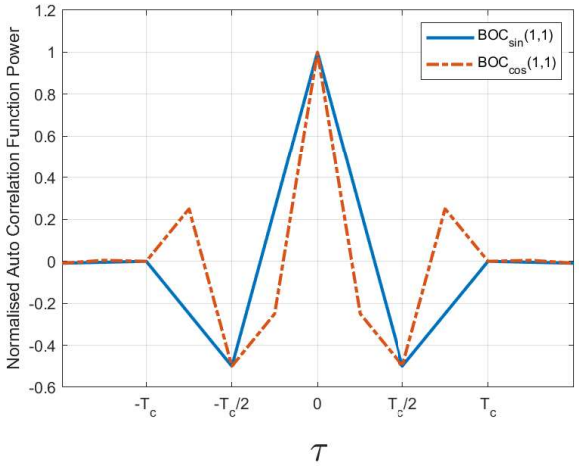


(b) PSD of $BOC_{sin}(6,1)$ and $BOC_{cos}(6,1)$.

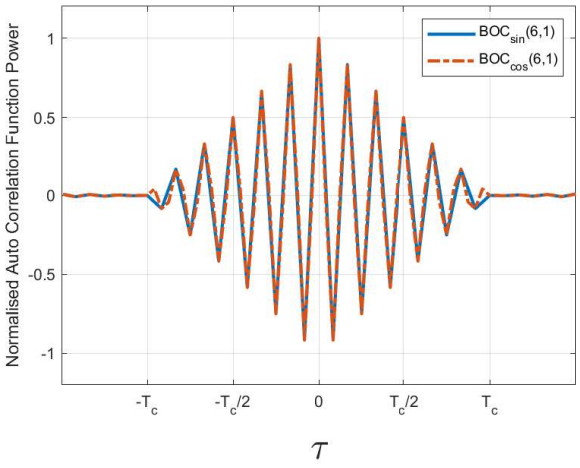
Figure 2.13: PSD of sin and cos-phased sub-carrier BOC(1,1) and BOC(6,1).

The difference between the sin-phased and cos-phased BOC modulation can be seen more significantly in $BOC_{sin}(6,1)$ and $BOC_{cos}(6,1)$, in case of sin-phased the power is more distributed in the inner lobes while in cos-phased BOC the power is more distributed in the outer lobes as illustrated in Figure 2.13b.

The ACF of the BOC signals has significant side lobes. The side lobe in a $BOC_{sin}(1,1)$ signal and a $BOC_{cos}(1,1)$ signal is approximatly half of the main correlation peak as illustrated in Figure 2.14a. For higher sub-carrier rates n_s, the ACF of BOC-modulated signals has even higher side lobes. The ACF of a $BOC_{sin}(6,1)$ and a $BOC_{cos}(6,1)$ has six side-lobes, the next to the main lobe has a power around 90 percent of the main lobe. The high side-lobes of the autocorrelation function, cause less robust time-delay estimation. Figure 2.15 shows a comparison between autocorrelation function of three signal with equal chipping rate and different modulation scheme BPSK, BOC(1,1) and BOC(6,1). It can be seen that the BPSK has an single ACF peak within one chip, the BOC(1,1) has a sharper main peak within half chip with two side peaks and BOC(6,1) has the most sharpest peak within $\frac{1}{12}$ chip with 12 side peaks, the second adjacent peak has approximately 90% of the main peak power. However, BOC modulation presents some



(a) ACF of $BOC_{sin}(1,1)$ and $BOC_{cos}(1,1)$.



(b) ACF of $BOC_{sin}(6,1)$ and $BOC_{cos}(6,1)$.

Figure 2.14: ACF of sin and cos-phased sub-carrier BOC(1,1) and BOC(6,1).

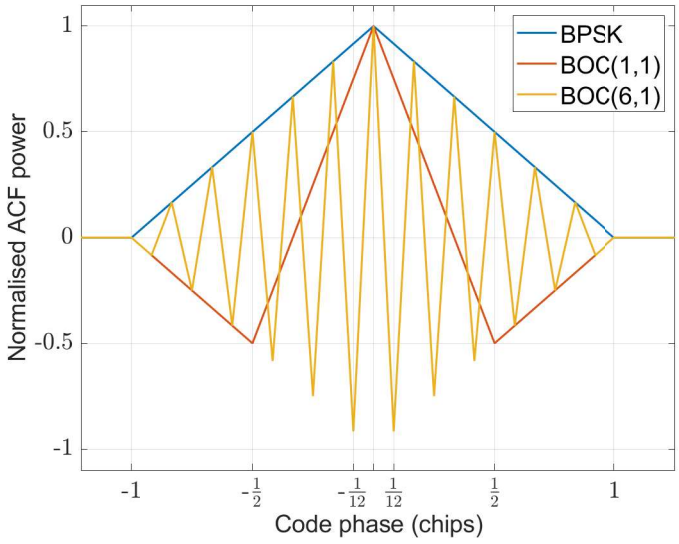
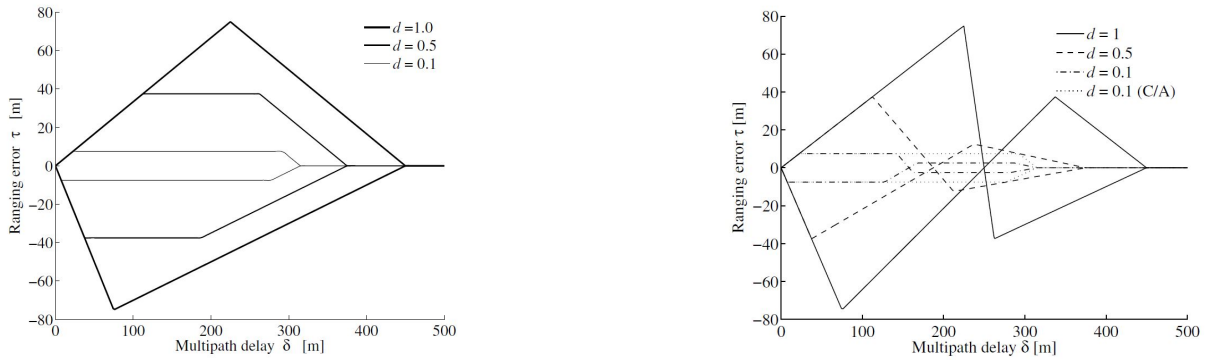


Figure 2.15: Normalised autocorrelation function comparison of BPSK(1), BOC(1,1) and BOC(6,1) calculated over 24 MHz bandwidth in MATLAB.

drawbacks, the most severe being the ambiguity problem in tracking. Since the sine-BOC modulated signal has a sawtooth-like, piecewise linear autocorrelation function which has multiple peaks, the receiver may lock onto one of the side peaks. This would result in intolerable bias in measurements.

Figure 2.16a and 2.16b, shows multipath error envelope for noncoherent early/late detector for C/A code (BPSK) and BOC(1,1) respectively. The positive multipath error corresponds to constructive interference while negative multipath error corresponds to destructive interference. For correlator spacing d = 0.1, it can be seen that the C/A code multipath error envelop is sensitive for multipath signals with a relative path delay up to 300 m. For BOC(1,1) the corresponding value is 150 m. This demonstrates that BOC(1,1) signals are better to handle multipath signals than C/A code signals (Borre et al., 2007).



(a) BPSK. (b) BOC(1,1).

Figure 2.16: Multipath error envelope for noncoherent early/late detector. The positive multipath error corresponds to constructive interference while negative multipath error corresponds to destructive interference.

(Borre et al., 2007)

2.3 Summary

The fundamentals of positioning using navigation satellites are based on trilateration. The user location is calculated using three or more satellites in the user's view with known positions in space (three-dimensional axis). The position of the satellite is determined from the navigation data transmitted by the satellite. The distance from the satellite to the user is calculated by determining the signal travel time between the user and the satellite. In general, the user position is measured using four satellites, three for measuring the distance, and one for error correction due to the user clock bias. The GNSS signal structure consists of three components: carrier, spreading code, and navigation data. The GNSS signals are transmitted on radio frequencies in the L band (1–2 GHz), this band has good propagation conditions including moderate attenuation and less impact of atmospheric effects, rain etc, L band has less expensive hardware and requires smaller antennas for communication. To differentiate the transmitted navigation data from each satellite, a multiple-access technique is used.

Most navigation systems employ the CDMA technique. In CDMA, all users (satellites) use the same bandwidth, but they are differentiated with different allocated codes, which are denoted as spreading codes.

The GNSS signal carries navigation data, which provides the user with information about the satellite orbits, clock correction, time of week, and other useful information for positioning. Navigation data are uploaded to the satellites from the control segment based on different points on earth.

The legacy signals of the satellites deploy BPSK and QPSK modulation techniques. PSK modulation corresponds to keying (changing) the carrier phase according to the binary-modulated data values. The two modulation types are denoted as $BPSK_{(m)}/QPSK_{(m)}$ which indicates the modulation of the navigation data signal with a spreading code rate of $m \times 1.023$ Mega chip per second.

The modernised GNSS satellite signals deploy BOC which is another spread spectrum technique. BOC modulation provides flexibility in shifting the signal power around the main carrier in the allocated band to reduce the interference with other signals, that is, better use of the spectrum, which makes the separation with GNSS signals easier. BOC signal provide better multipath mitigation than BPSK modulated signals.

Chapter 3

Satellite navigation systems

This chapter presents an overview of the four main global positioning systems, the space segment is described. Then, an overview of the control segment is given. This is followed by an overview of the signal structure, as well as a description of the navigation data content.

3.1 Global positioning system

GPS is a satellite navigation system maintained by the United States Air Force. The system consists of a nominal constellation of 24 satellites in a medium earth orbit (MEO) and a worldwide control segment to track and control satellites. The GPS program began in the early 1970s. The initial operational capability for civil use is declared at the end of 1993, and FOC is attained by the end of 1994 (Parkinson et al., 1996).

3.1.1 Space segment

The GPS constellation consists of six orbital planes in MEO with a radius of 26,560 km (at an altitude of approximately 20.200 km). The orbital planes are inclined at 55° with respect to the equatorial plane and separated by 60° to cover the complete 360°. In each orbital plane, there are four asymmetrically spaced slots, each slot containing a satellite, which provides more reliability against probable satellite failures (Tsui, 2005). Previously, the GPS baseline 24-slot constellation consisted of 24 slots in six orbital planes with four slots per plane. In recent years, three of the 24 slots have become expandable. Each of the three slots, B1, D2, and F2, split into two slots to accommodate up to 27 total satellites in the constellation. Surplus satellites (satellites that do not occupy a slot in the GPS constellation, may be "young" recently launched satellites waiting to move into a slot, or they may be "old" nearly worn out satellites providing their last few months or years of navigation service before they finally expire) are typically placed in locations adjacent to satellites (Grimes, 2008) .

3.1.2 Control segment

The GPS control segment (CS) consists of a global network of ground stations that track the GPS satellites, monitor their transmissions, perform analyses, send control commands to the constellation, and uploads the satellites with navigation data. The CS includes a master control station, a global network of monitoring stations, and ground antennas.

The master control station provides control commands to the GPS constellation, analyses global monitoring station data to keep the satellites in precise locations,uploads the satellites with navigation messages, monitors satellite signals to ensure constellation health and accuracy, and performs satellite maintenance and anomaly resolution, including repositioning satellites to maintain optimal constellation.

The global network of monitor stations includes high-precision GPS receivers that track the available GPS satellites in view, collects navigation signals from satellites, range/carrier measurements, and atmospheric data to send them to the master control station. The ground antennas are high-gain antennas used to read telemetry data from GPS satellites, send commands, navigation data uploads, and processor program loads to the GPS satellites.

3.1.3 L1 and L2 signal structure

The oldest operational satellites, block II/IIA and IIR, broadcast only what is now known as legacy GPS signals. The legacy GPS signals are composed of a coarse/acquisition (C/A) code signal on L1 with a carrier component of 1575.42 MHz and the Precision (P) code signal on L1 and L2. The P-code is in- phase quadrature with C/A-code on L1. The L2 signal with a carrier component of 1227.6 MHz is only modulated with P-code (in-phase), while the P-code on L2 might be a modulo-2 sum with legacy navigation (LNAV) data or without LNAV data selected by a ground command. The C/A code is open to the public. The P-code signal is intended only for authorised (military) use and is normally encrypted. Both legacy GPS signals are generated using direct sequence spread spectrum (DSSS) modulation. The signals are formed as a product of three components:

3.1.3.1 Navigation data

The legacy navigation (LNAV) data $D_{(t)}$ have a bit rate of 50bps, it provides information about the satellite vehicle (SV) ephemerides, clock correction, time of week, and other useful information for positioning. This information is uploaded from the GPS control segment based on different points on earth. The LNAV data is modulo-2 added to the P(Y)- and C/A-codes, and the resultant bit trains are used to modulate the L1 and L2 carriers. All of the modernised civil GPS signals provide navigation data content similar to that provided by the C/A-code and P(Y)-code signals. L5 transmits the navigation data at 50 bps and L2C at 25 bps. The navigation data format for L2C and L5 is referred to as civil navigation (CNAV), and the format for L1C is referred to as CNAV-2. Block IIR-M, Block IIF, and subsequent blocks of SVs transmit CNAV data, Dc(t), as well as information about the SV ephemerides, clock correction, time of week, etc. Dc(t) is a 25 bps data stream which is encoded by a rate 1/2 convolutional encoder resulting 50 sps symbol stream. The 50 sps stream is modulo-2 added to the L2 CM code.

Compared to the LNAV, the CNAV message provides six additional parameters (two orbit parameters and four inter-signal correction (ISC) parameters for prospective civil users. Using the precise products of the international global navigation satellite system service (IGS), the authors in (Wang et al., 2019) evaluated the precision of satellite orbit, clock and ISCs of the CNAV. Additionally, the contribution of the six new parameters to GPS single point positioning (SPP) is analysed using data from 22 selected multi-global navigation satellite system experiment (MGEX) stations from a 30-day period. The results indicated that the CNAV/LNAV signal-in-space range error (SISRE) and orbit-only SISRE from January 2016 to March 2018 was of 0·5 m and 0·3 m respectively, which

is improved in comparison with the results from an earlier period. The ISC precision of L1 C/A was better than 0·1 ns, and those of L2C and L5Q5 were about 0·4 ns. Remarkably, ISC correction has little effect on the single-frequency SPP for GPS users using civil signals (for example, L1C, L2C), whereas dual frequency SPP with the consideration of ISCs results had an accuracy improvement of 18·6%, which is comparable with positioning accuracy based on an ionosphere-free combination of the L1 P(Y) and L2 P(Y) signals.

3.1.3.2 Spreading codes

There are two different spreading codes for each satellite, transmitted through the L1 channel, the first spreading codes called Gold code, and used for coarse acquisition (C/A), which are also called coarse acquisition codes. Gold codes are a class of sequences, which provide reasonably large sets of codes with good periodic cross-correlation nearly reaching Welch's bound as well as good autocorrelation properties. The C/A code has a chipping rate of 1.023 MHz with a sequence length of 1023 chips, and each sequence is repeated every millisecond. C/A-code is the modulo-2 sum of two 1023-bit linear patterns, G1 and G2i. As illustrated in Figure 3.1, the G2i sequence is formed by effectively delaying the G2 sequence by an integer number of chips. The G1 and G2 sequences are generated by 10-stage shift registers with the following polynomials (Dunn and DISL, 2012):

$$G1 : X^{10} + X^3 + 1$$
$$G2 : X^{10} + X^9 + X^8 + X^6 + X^3 + X^2 + 1$$

The relationship between shift register taps and the exponents of the corresponding polynomial, referenced to the shift register input. The effective delay of the G2 sequence to form the G2i sequence is obtained by combining the output of two stages of the G2 shift register by modulo-2 addition. The G1 and G2 registers are clocked at 1.023 MHz, which is obtained from the 10.23 MHz main satellite atomic clock by dividing this clock by 10. The G1 and G2 registers are initialised with all ones. The G1 and G2 shift registers are initialised at the P-coder X1 epoch.

The second spreading code is called the encrypted precision code P(Y) or P code before it has been encrypted. The P code is a longer code with a chipping rate of 10.23 MHz, repeats itself each week. The C/A code is only modulated onto the L1 carrier, while the P(Y) code is modulated onto both the L1 and L2 carriers, as illustrated in Figure 3.2. Block IIR-M, Block IIF, and subsequent blocks of SVs, introduced two new navigation signals, a new military signal on L1 and L2 referred to as the M code, and a new civil signal on L2, referred to as L2C (Dunn and DISL, 2012).

L2 consists of two carrier components which are in phase quadrature. The in-phase component modulated with M-code and P-code. The P-code might be a modulo-2 sum with LNAV data or without LNAV data (this is selected with a ground command). The quadrature-phase component is modulated by a new L2C civilian signal.

L2C is a chip-by-chip time multiplex combination of two spreading codes known as civil moderate (CM) and civil long (CL) . The CM-code is modulo-2 added to the navigation data, while the CL code is not added, that is, data less.

A data-less component (also referred to as a pilot) is a GNSS signal that is not modulated by the navigation data. The benefit of adding a pilot component is to make the tracking process of the signal more robust by a receiver in low signal-to-noise conditions. The receiver can track the pilot component using a pure phase-locked loop (PLL) , whereas

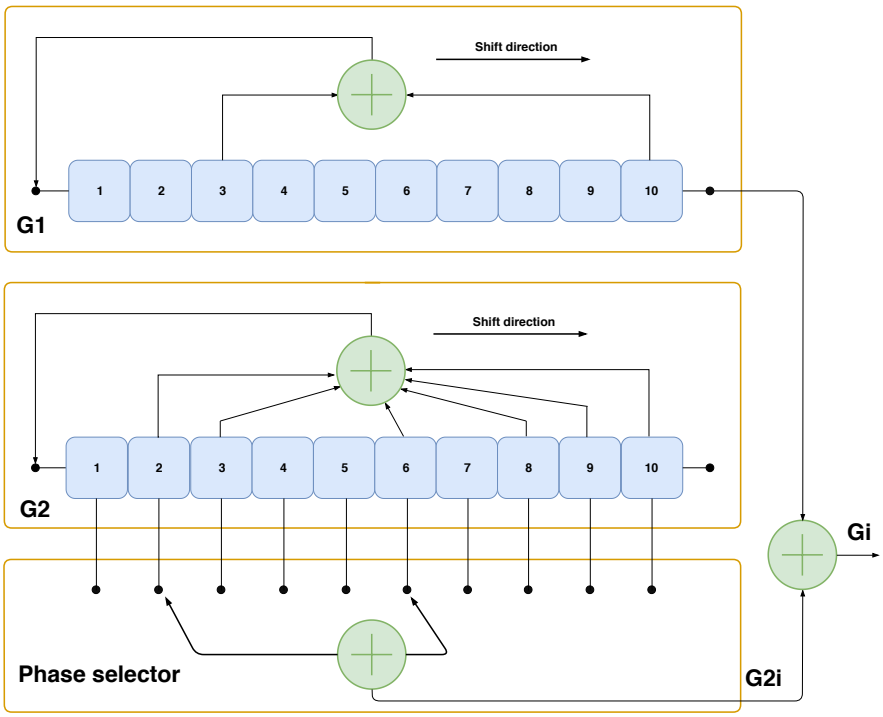


Figure 3.1: Gold code generator.

a Costas loop is required to track a signal modulated by unknown binary data. A PLL can track a signal with approximately one-quarter of the signal-to-noise ratio necessary for a Costas loop. Even though only one-half of the total power in the transmitted L2C signal is devoted to a data-less component, there is still a net 3 dB tracking robustness (Fontana and Cheung).

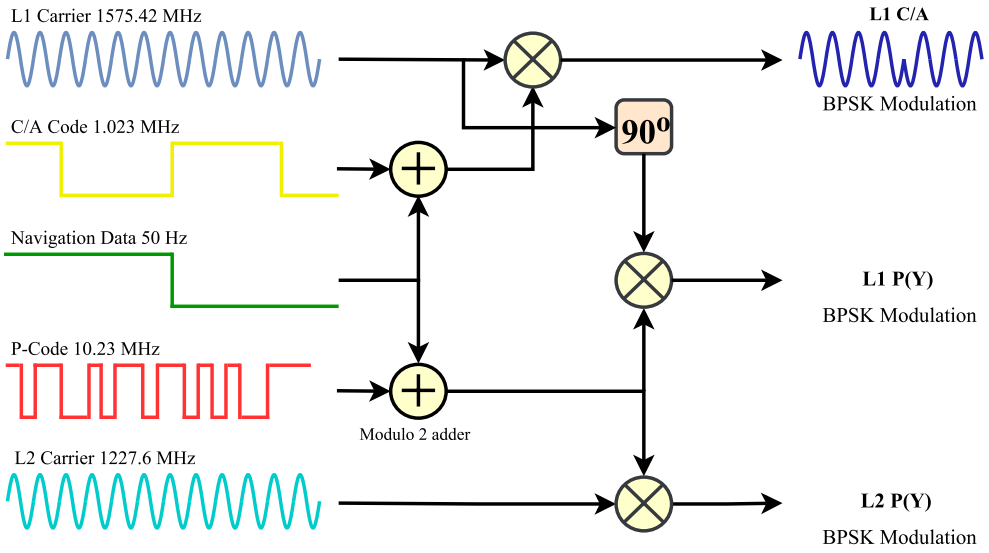


Figure 3.2: GPS L1, L2 legacy signal modulation.

Both L2 CM-code and L2 CL-code patterns are generated using the same code generator polynomial, each clocked at 5.115 MHz. The CM-code sequence is 10230 chips long with a duration of 20ms, while the CL-code sequence is 767250 chips long with an aduration of 1.5 Second. The details of the L2C code generator can be found in the GPS interface

control document (Dunn and DISL, 2012).

Block IIF satellites introduced a third civil signal on a new carrier frequency. Both the carrier and the signal are referred to as L5 (Van Dierendonck, 2000).

3.1.3.3 Carrier

The carrier component has a frequency of 1575.42 MHz in L1 and 1227.6 MHz in the L2 band. All signals use binary BPSK modulation. The GPS L1 C/A signal $S_{(t)}$ travels down from the satellite is composed from the L1 carrier, spreading code and the Navigation data.

$$S_{(t)} = AD_{(t)}X_{(t)}\cos(2\pi f_{L1}t) \tag{3.1}$$

where
A is the signal amplitude
$D_{(t)}$ is the navigation data
$X_{(t)}$ is the spreading code
f_{L1} is the L1 carrier frequency.

3.1.4 L5 signal structure

Block IIF and subsequent blocks of SVs introduced a new navigation signal, known as L5. L5 is composed of two carrier components that are in the phase quadrature. The in-phase carrier component (I5) is BPSK modulated with a modulo-2 sum of I5 code, L5 civil navigation data referred to as L5 CNAV, and synchronisation sequence. The in-quadrature carrier component (Q5) is BPSK modulated with a modulo-2 sum of Q5 code without navigation data and synchronisation sequence which is different from the I5 component sequence (Dunn and DISL, 2012). The L5 signal is generated using the DSSS modulation. The L5 signal is a product of three components.

3.1.4.1 Navigation data

L5 transmits L5 CNAV data at 100 sps, which is modulated into the in-phase component I5. The L5 CNAV data, D5(t), contains SV ephemerides, GPS system time, SV clock correction data, etc. The 100 sps data is the result of 50 bps navigation data encoded in a rate 1/2 convolution encoder and clocked out at 100 sps. The resulting 100 sps data stream is modulo-2 added to the I5-code only and BPSK modulated into the L5 in-phase (I) carrier (Dunn and DISL, 2012).

3.1.4.2 Spreading codes

There are two different spreading codes for each satellite, transmitted through the L5 channel, the first spreading code called (I5-code), used to modulate the I5 component, and the second spreading code called (Q5-code), which is used to modulate the Q5 component. Both code patterns are generated by a modulo-2 summation of two pseudo-random noise (PRN) codes. The I5-code is the result of the modulo-2 summation of XA(t) and XBIi(nIi, t), whereas the Q5-code is the result of the modulo-2 summation of XA(t) and XBQi(nQi, t), where nIi and nQi are the initial states of XBIi and XBQi for satellite i. The patterns XA(t), XBI(t), and XBQ(t) generated using three 13 bit shift registers clocked at 10.23 MHz as illustrated in Figure 3.3.

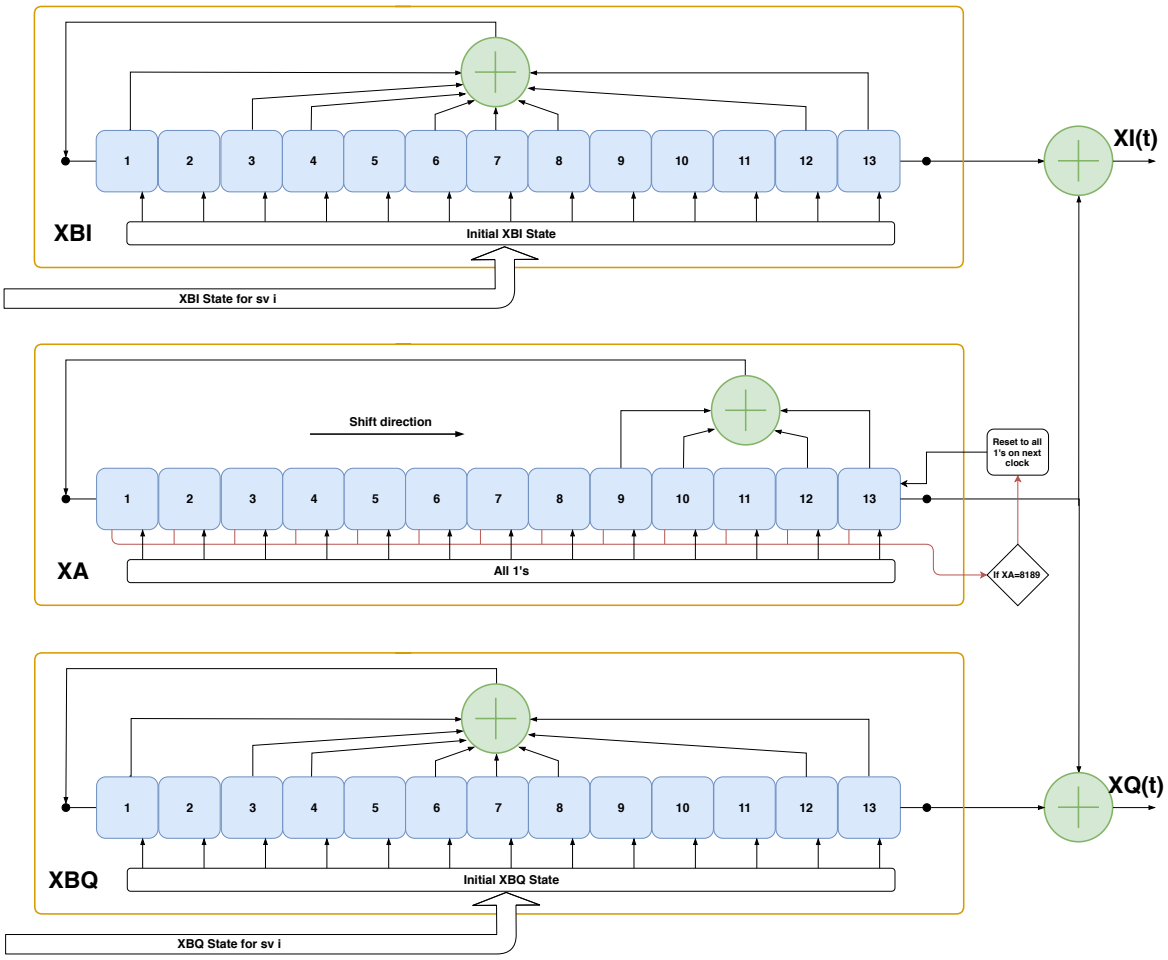


Figure 3.3: GPS L5 shift register based code generator.

The 13 bit shift register generates 8191 different states, which are short-cycled by one bit to obtain a code length of 8190 chips, with a register pre-set value of all ones at the start and before roll-over by one chip. The XA register also re-started over each one millisecond (synchronised with the L1 frequency C/A-code) to extend the code to 10230 chips. The XBIi and XBQi, with different initial conditions for each satellite, are 8191 chip length codes that are not restarted before roll-over. They are restarted at their natural roll-over with their initial conditions. Both registers are re-started each one millisecond (synchronised with the L1 frequency C/A-code) to extend the generated codes to 10230 chips. The polynomials for XA and XBIi or XBQi codes, as referenced to the shift register input, are:

$$XA : 1 + x^9 + x^{10} + x^{12} + x^{13}$$
$$XBIi \quad or \quad XBQi : 1 + x^1 + x^3 + x^4 + x^6 + x^7 + x^8 + x^{12} + x^{13}$$

The relationship between the shift register taps and the exponents of the corresponding polynomial, referenced to the shift register input, are as shown in 3.3.

3.1.4.3 Carrier

The carrier component has a frequency of 1176.45 MHz in the L band. L5 deploys quadrature phase-shift keying (QPSK) modulation. Figure 3.4 illustrates the L5 signal

modulation scheme. In the data channel, that is, I5, 276 bits of L5 CNAV at a 50 bps rate are added to a 24 bit parity block referred to as cyclic redundancy check (CRC) to form a 300 bit navigation message, 6 s long. CRC parity provides protection against burst as well as random errors (Directorate, 2011). After adding the CRC bits to the navigation message, the resulting bit trains at 50 bps streams through a convolution encoder known as forward error correction (FEC) . The navigation message is FEC encoded in a continuous process independent of message boundaries. The L5 CNAV bit train at 50 bps is the rate 1/2 convolution encoded and clocked at 100 sps. In addition, the 100 sps symbols are modulated with a 10-bit Neuman-Hofman code that is clocked at 1 kHz. The 10-bit Neumann–Hofman code is defined as 0000110101. The 10 bits are modulo-2 added to the symbols at the PRN code epoch rate of 1 kHz starting at the 100 sps symbol transitions to extend the code by a factor of 10. The result is that a "1" data symbol is replaced by 1111001010, and a "0" data symbol is replaced by 0000110101. The resultant symbol sequence is then used to modulate the L5 in-phase carrier. The Q5 channel is data-less

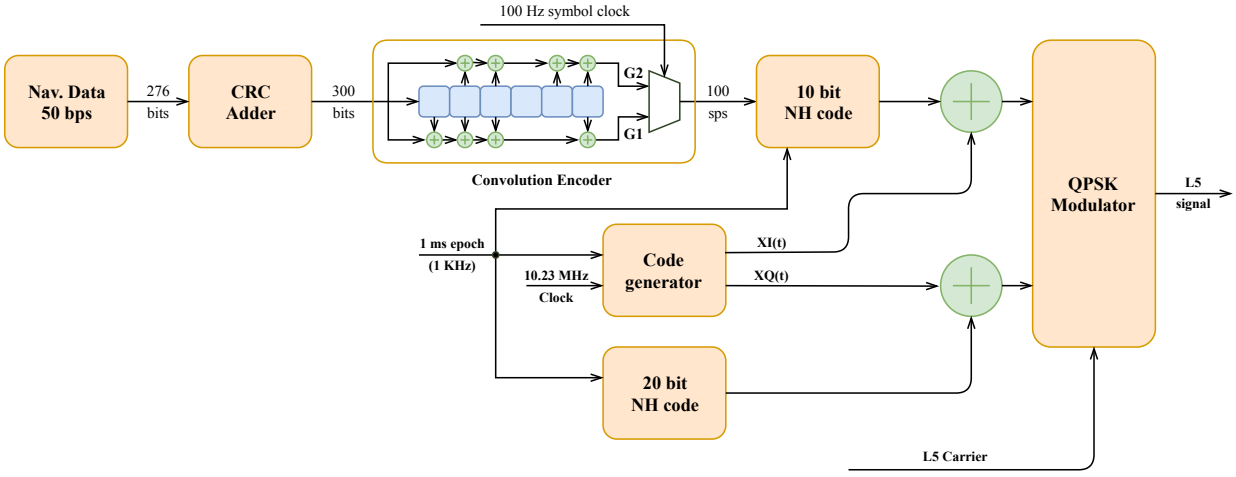


Figure 3.4: GPS L5 signal modulation scheme.

i.e. not modulated with L5 CNAV. The Q5-code is modulated using a 20-bit Neuman-Hofman code clocked at 1 kHz. Each of the 1 ms Q5-code blocks is further encoded using a 20-bit Neuman-Hofman code. The 20 bits are modulo-2 added to the Q5 code chips at the PRN code epoch rate of 1 kHz to extend the code by a factor of 20. Both I5 and Q5 signals QPSK modulated to form an L5 signal.

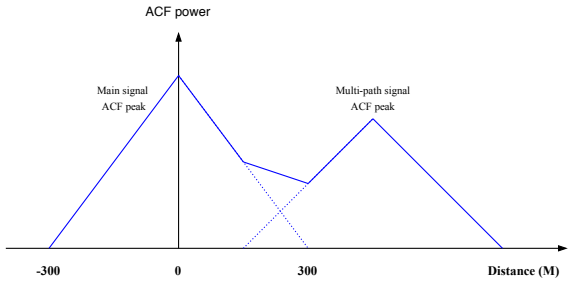
L5 spreading codes have a faster chipping rate and longer sequence compared to L1 C/A codes. The L5 chipping rate is 10.23 Mcps, while the L1 C/A chipping rate is 1.023 Mcps. The chip-length (wavelength) for both codes can be calculated as follows:

$$\lambda_{L1} = \frac{c}{f_{C/A}} = \frac{299792458m/s}{1.023 * 10^6} = 293.05m \tag{3.2}$$

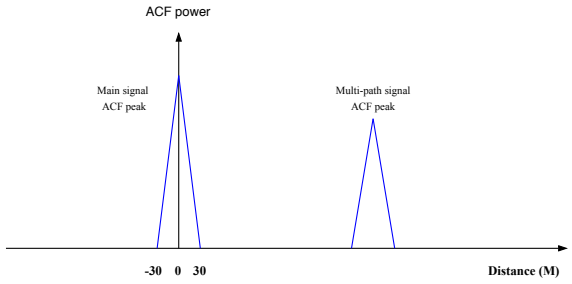
$$\lambda_{L5_I} = \frac{c}{f_{L5_I}} = \frac{299792458m/s}{10.23 * 10^6} = 29.305m \tag{3.3}$$

Where λ_{L1} and λ_{L5_I} are the chip length (wavelength) of the L1 C/A and $L5_I$ chipping rates respectively; c is the speed of light; $f_{C/A}$ is the chipping rate of L1 C/A code; and f_{L5_I} is the chipping rate of the $L5_I$ code. Assuming the time of arrival can be measured

with accuracy of approximately 0.1% in a receiver, this corresponds to a range precision of approximately $0.3m$ using C/A code (Misra and Enge, 2006). For the same receiver the range precision will be approximately $0.03m$ using L5 codes. The faster chipping rate also provides a sharper ACF peak, which results in better ranging precision and better multipath resolution, as shown in Figure 3.5.



(a) The C/A code can resolve multipath from 300m or more.



(b) The I5 code can resolve multipath from 30m or more.

Figure 3.5: Effect of chipping rate on resolving the multipath.

In summary, L5 has several advantages over L1: signal structure, wide bandwidth, pilot codes. The use of L5 signals provides the capability of delivering 10x higher precision than a legacy L1 GNSS receiver in an open environment, as well as very noticeable benefits in multipath environments.

3.2 GLONASS

GLONASS is a global navigation satellite system developed by the Russian Federation. In 1988, GLONASS provided air safety services for civil use, details of the GLONASS system, and signals are released to the international organization for air safety (ICAO). Because of the limited operational lifetime of the early generation of GLONASS satellites and insufficient replacement, the number of active satellites gradually reduced to seven active satellites in 2001. Since the end of 2001, the GLONASS Federal Program secured a budget that enabled significant step by step performance improvements on the system. Regularly launching the extended lifetime of the new GLONASS-M satellites helped to gradually increase the number of operational satellites. The 24-satellite constellation required for a fully global service is ultimately re-established in 2011. On 17 May 2007 the Russian Federation declared the GLONASS open service available to all national and international users without any limitations (Kaplan and Hegarty, 2017).

GLONASS provides two types of service:

- An open service with unencrypted signals in three frequency bands L1, L2, and L3, which is globally available for all users without any regulations referred to as standard accuracy (ST), accessible to any user.
- A service for authorised users, using encrypted signals in presently two frequency bands L1 and L2, referred to as pinpoint accuracy (W), accessible only to special users.

Each GLONASS satellite transmits signals for both open and authorised services. The early version of GLONASS satellites launched from 1982 through 2005 deployed FDMA

modulation, and signals transmitted by a specific satellite utilise the same ranging code, but different carrier frequencies in the L1 and L2 sub-bands to keep the signals from different satellites differentiated. The GLONASS satellites launched in 2003, known as GLONASS-M, are the modernised versions of the original legacy GLONASS satellites, which started to transmit additional FDMA signals for all users without any regulations on L2. The GLONASS-M SVs, launched in 2014, added an additional open CDMA signal on the L3 sub-band. At the beginning of 2011, Russia launched a new generation of satellites known as GLONASS-K1. This generation transmits the legacy signals and the open CDMA signal which is transmitted for the first time by GLONASS-M SVs in 2014. In 2018, Russia launched GLONASS-K2 satellites, which continued to carry legacy FDMA signals and the CDMA on L3, and also transmitted CDMA signals on L1, L2, and L3 (Kaplan and Hegarty, 2017). Table 3.1 shows an overview of the GLONASS signal evolution, where the signals are identified by the frequency band (first two characters), the service type O stands for open, S stands authorised, and the modulation type F for FDMA, C for CDMA.

Table 3.1: GLONASS signals overview.

Satellites	**FDMA**	**FDMA**	**CDMA**	**CDMA**	**CDMA**
Launch / Status	L1	L2	L1	**L2**	**L3**
GLONASS	L1OF	L2SF			
1982-2005 / (out of service)	L1SF				
GLONASS-M	L1OF	L2OF			L3OC*
2003- / (in service)	L1SF	L2SF			
GLONASS-K1	L1OF	L2OF			L3OC
2011- / (in service)	L1SF	L2SF			
GLONASS-K2	L1OF	L2OF	L1OC	L2OC	L3OC
2018- / (in test mode)	L1SF	L2SF	L1SC	L2SC	L3OC

3.2.1 Space segment

The GLONASS constellation consists of 24 satellites in three orbital planes in the MEO at an altitude of 19.100 km. The orbital planes are inclined 64.8° with respect to the equatorial plane and separated by 120° to cover the complete 360°, and each satellite completes the orbit in approximately 11 h 15 min. In each orbital plane, there are eight symmetrically spaced slots, each slot contains a satellite, and the constellation populates 24 operational satellites. Each GLONASS satellite is identified by its slot number, which defines the orbital plane and its location within the plane. Slot numbers 1–8 belong to orbital plane 1, while slot numbers 9–16 belong to orbital plane 2 and slot numbers 17–24 belong to orbital plane 3 (GLONASS, 2008).

3.2.2 Control segment

The control segment includes the system control centre and the network of the command and tracking stations that are located throughout Russia. The control segment is as follows:

- Monitoring of GLONASS constellation status.
- Correction to the orbital parameters.
- Navigation data uploading.

3.2.3 L1OF and L2OF signal structure

In this section, structure and performances of navigation radio signal ST in sub-bands L1 and L2, transmitted by "Glonass" and "Glonass M" satellites known as L1OF and L2OF are considered.

The first-generation operational GLONASS satellites only broadcast what is now known as the legacy GLONASS signals using only FDMA modulation utilising L1 and L2 bands for both public and authorised users. The open service signal is only transmitted on the L1 frequency in the first generation of GLONASS satellites, but it is made available at both frequencies starting with the GLONASS-M satellite generation. Despite the ongoing system evolution and the introduction of new CDMA signals, GLONASS will continue to provide legacy FDMA signals in the future to provide backward compatibility with the user equipment already in use. The GLONASS L1 and L2 signals are products of three components.

3.2.3.1 Navigation data

The navigation message transmitted at 50 bps and 100 Hz auxiliary meander sequences. The navigation message includes data related to the satellite, which transmits the given navigation signal and data related to all satellites within the GLONASS constellation (GLONASS almanac).

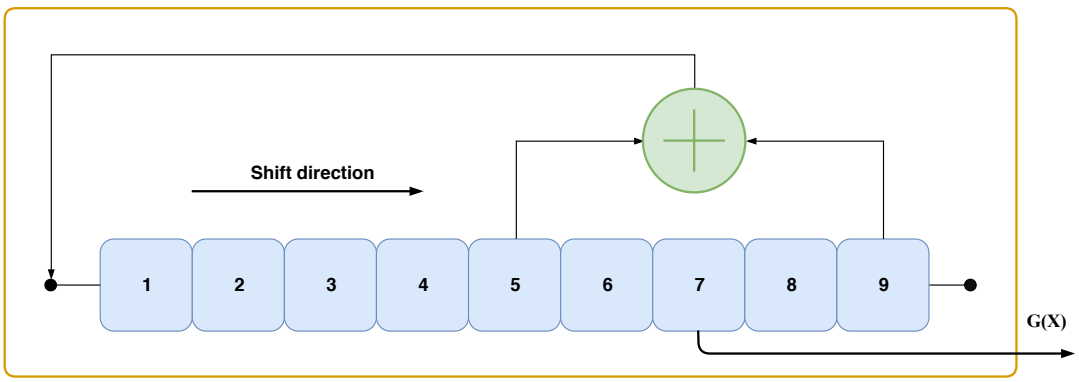


Figure 3.6: GLONASS L1 and L2 ranging code generator.

3.2.3.2 Spreading codes

The spreading code G(X) is a sequence of the maximum length of the 9-stage shift register clocked at 0.511 MHz, as illustrated in Figure 3.6. The period of one sequence is 1 ms and the chip rate is 511 kbps. G(X) ranging code is tapped out at the 7th stage of the 9-stage shift register. The shift register is initialised to all the ones at the beginning to generate this sequence. The first character of the ranging code is the first character in group 111111100, and it is repeated every 1 ms. The generating polynomial, which corresponds to the 9-stage shift register, is (GLONASS, 2008):

$$G(X) = 1 + X^5 + X^9$$

3.2.3.3 Carrier

Unlike GPS, GLONASS deploys the FDMA technique to transmit signals in L1 and L2. Satellite generations up to GLONASS-K1 deploy FDMA, while the GLONASS-K2 and subsequent generation launched after 2018 (they are not in service yet) start introducing CDMA beside the existing FDMA signals. A set of carriers with different frequencies is required to differentiate the satellite signal in the constellation. This set of carriers allocated in the L1 and L2 sub-bands and the nominal values of these carrier frequencies are defined as follows:

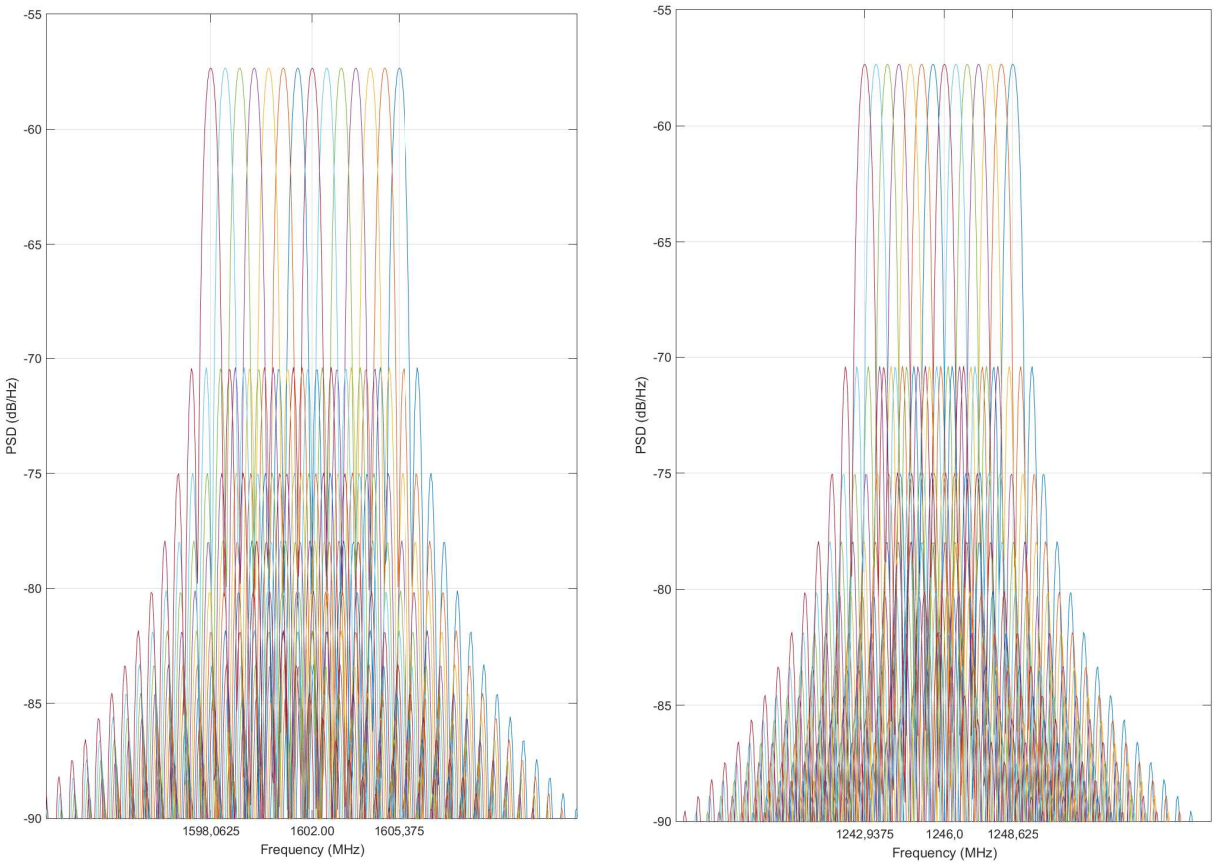
$$f_{K1} = f_{01} + K\Delta f_1 \tag{3.4}$$
$$f_{K2} = f_{02} + K\Delta f_2 \tag{3.5}$$

where
K is the frequency channel number used by GLONASS satellites in the L1 and L2 sub-bands to transmit the signal.
f_{01} = 1602 MHz; Δf_1 = 562,5 kHz, for sub-band L1
f_{02} = 1246 MHz; Δf_2 = 437,5 kHz, for sub-band L2
The channel number K for any particular GLONASS satellite is provided in almanac (non-immediate navigation message data). All GLONASS SVs launched after 2005 use numbers of frequencies K = (-7... +6), where the channel numbers K = +5 and K = +6 may be used for only technical purposes over the Russian Federation (for instance, when performing replacements within the space segment) (GLONASS, 2008).

GLONASS constellation populates 24 satellites; however, it utilises only 14 frequency channels in the L1 and L2 sub-bands, which is achieved by sharing the same frequency channels between antipodal (opposite side of planet in orbit) satellite pairs, as these satellites are never both in view of an earth-based user at the same time. Each satellite is given a unique spreading code to be identified in acquisition process. Figure 3.7 illustrates the PDS for L1OF and L2OF signals. The navigation signal being transmitted at a specific carrier frequency of the L1 and L2 sub-bands is formed from BPSK modulation of the carrier with a binary bit train which is the product of the modulo-2 sum of the binary navigation data and ranging code.

3.2.4 L3OC signal structure

The first open CDMA signal introduced from the GLONASS system transmitted by GLONASS-K1 satellites, launched in 2011, is also transmitted by the latest version of GLONASS-M satellites launched in 2014. This signal is called L3OC and is transmitted



(a) L1OF signal PSD.

(b) L2OF signal PSD.

Figure 3.7: The legacy GLONASS signals transmitted from 24 satellites using only FDMA modulation utilising only a set of 14 channels (carriers) inside the L1 and L2 sub-bands, by sharing the same frequency channels between antipodal (opposite side of planet in orbit) satellite pairs, each channel is a BPSK modulation of the carrier with a unique spreading code per each satellite.

at a carrier frequency of 1202.025 MHz in the L band. Similar to the GPS L5 signal, L3OC is composed of two BPSK(10) components with equal power strength added together to form a QPSK. The first component is the product of the modulo-2 sum of navigation data and PRN sequence, modulated with the in-phase (I) component of the carrier, referred to as the data component (L3OCd), and a data-less PRN sequence modulated with the quadra-phase (Q) component of the carrier, referred to as the pilot component (L3OCp). These two components are in phase quadrature, and L3OCd is in the in-phase (I) and L3OCP in the quadrature-phase (Q).
The L3OC signal is generated using DSSS modulation, and the signal is a product of three components:

3.2.4.1 Navigation data

The navigation message of the L3OCd signal is transmitted at 100 bps. The 100 bps data stream of the L3OCd signal is encoded by a rate 1/2 convolution encoder (CE) producing a 200 sps stream as shown in Figure 3.8. The encoder switches the first half (lower position) of a 10-millisecond data bit period first. The GLONASS navigation message contains immediate (ephemeris and clock corrections), non-immediate (almanac, Earth's

rotation parameters, ionosphere model parameters, etc.) information (GLONASS, 2016).

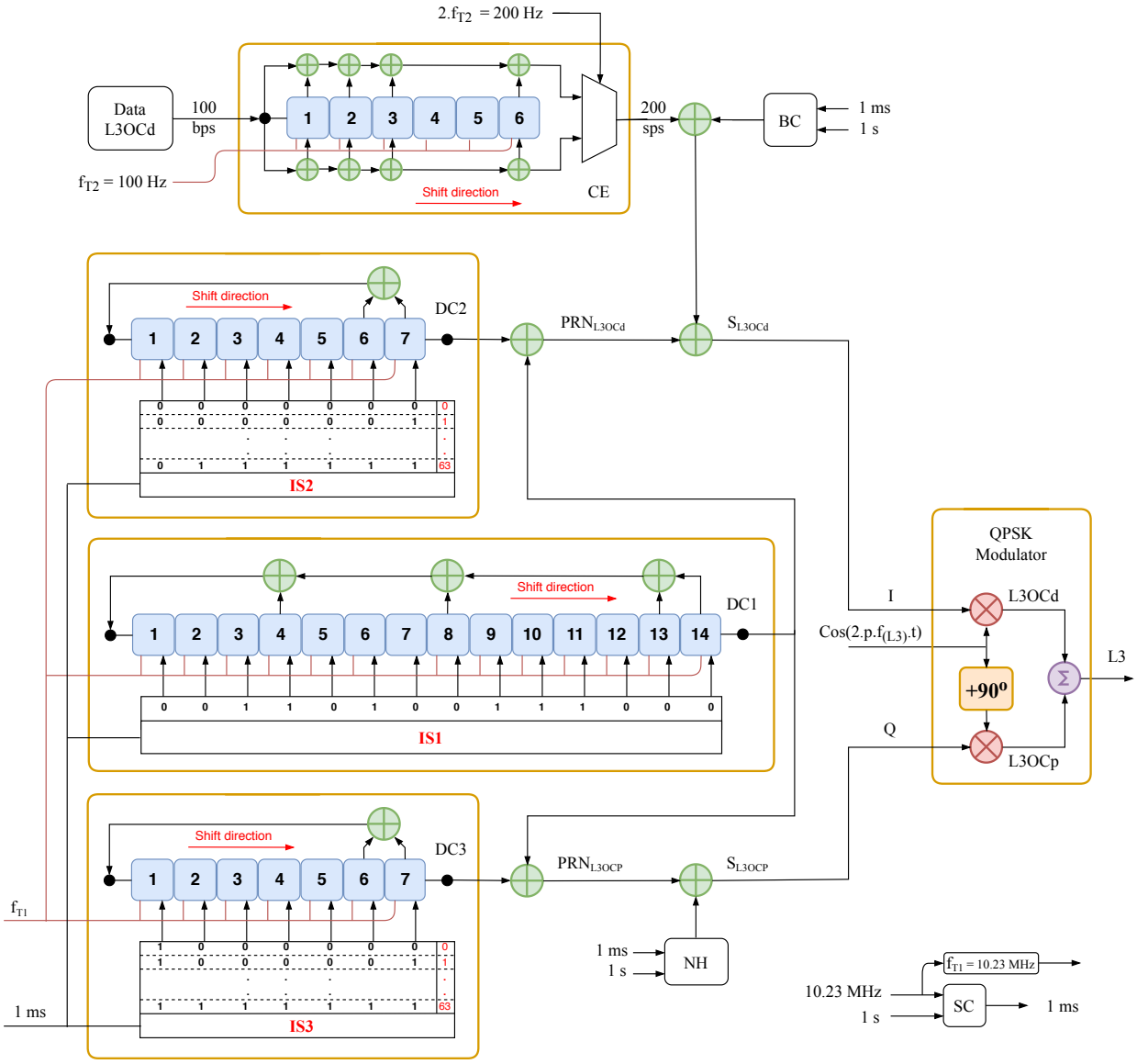


Figure 3.8: GLONASS L3OC modulation scheme.

3.2.4.2 Spreading codes

There are two different spreading codes for each satellite, transmitted through the L3OC channel, the first spreading code referred to as (PRN_{L3OCd}),used to modulate the I component of the carrier and (PRN_{L3OCd}), used to modulate the Q component. Both code patterns are truncated Kasami sequences of 10230 chips with a period of 1 ms. The two sequences are generated by a modulo-2 summation of two PRN codes. The PRN_{L3OCd} is the result of modulo-2 summation of the DC1 and DC2 sequences, while the PRN_{L3OCp} is the result of modulo-2 summation of the DC1 and DC3 sequences. The Pattern DC1 generated using a 14 bit shift register clocked at 10.23 MHz, DC2 and DC3 are generated using three 7 bit shift registers clocked at 10.23 MHz as illustrated in Figure 3.8. The polynomials for DC1, DC2 or DC3 codes, as referenced to the shift register input, are as

follows:

$$DC1 : 1 + x^4 + x^8 + x^{13} + x^{14}$$
$$DC2 \quad or \quad DC3 : 1 + x^6 + x^7$$

The registers restarted every 1 ms with the following initial states(IS):
IS1 = 00110100111000 for DC1;
IS2 = j = 0000000, 0000001...0111111 for DC2, where j is the SV ID number;
IS3 = j + 64 = 1000000, 1000001...1111111 for DC3, where j is the SV ID number.
IS1, IS2 and IS3 (binary numbers) are loaded in DC1, DC2 and DC3 in such a way that the least significant bit enters the last stage of register. Pulses at 1 ms intervals to restart and load DC registers with IS values are generated in a synchronous counter (SC), as shown in Figure 3.8 based on 10.23 MHz signal and 1 s pulses.

3.2.4.3 Carrier

The Carrier frequencies of all transmitted signals and clock rates of modulating sequences are coherently derived from the on-board atomic frequency standard (AFS). The nominal value of the AFS is 5.115 MHz. The carrier component has a frequency of $f_{L3} = 1175 \cdot f_b = 235 \cdot 5.115MHz = 1202.025MHz$ in the L band. L3OC deploys quadrature phase shift keying (QPSK) modulation.

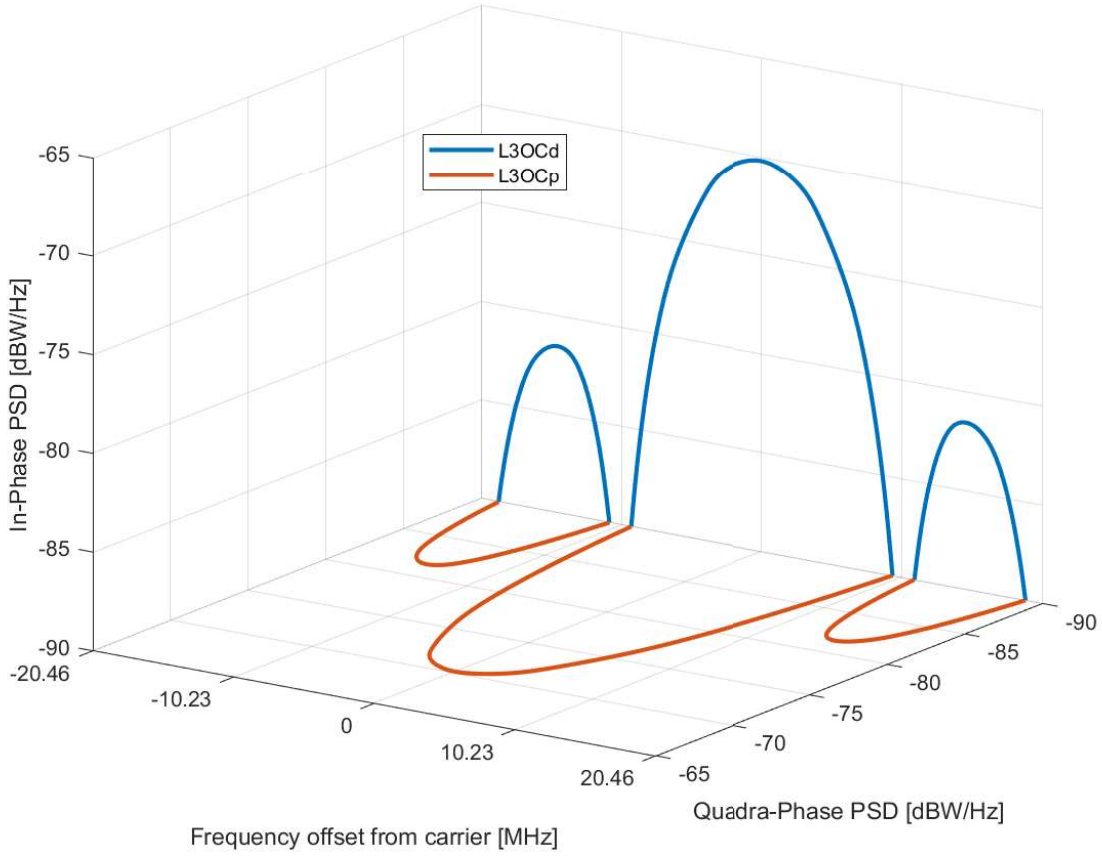


Figure 3.9: GLONASS L3OC PSD.

As illustrated in Figure 3.8 the signal (SL3OCd) is the modulo-2 sum of PRN_{L3OCd} chip stream clocked at $f_{T1} = 10.23MHz$, Barker code (BC) symbol stream clocked at 1000

sps, and convolution encoder (CE) symbol stream clocked at 200 sps. Sequence S_{L3OCd} is used to phase-shift key I-component of the carrier frequency in L3. The signal S_{L3OCp} is the modulo-2 sum of PRN_{L3OCp} chip stream clocked at $f_{T1} = 10.23MHz$ and Newman-Huffman (NH) code symbol stream clocked at 1000 sps. Sequence S_{L3OCp} is used to phase-shift key Q-component of the carrier frequency in L3.

BC is a periodic 5-symbol code 00010 clocked at 1000 sps (1 KHz), synchronised with CE symbols ($T_{CE} = 5ms$) and transmitted with the most significant bits first. NH (Newman-Huffman code) is a periodic 10-symbol code 0000110101 clocked at 1000 sps (1 KHz), synchronised with data bits ($T_{data} = 10ms$) and transmitted with the most significant bits first (the first symbol of NH for a data bit duration is 0) (GLONASS, 2016). The PSD of the L3OC signal is shown in Figure 3.9. The signal consists of two BPSK(10) components of the same power L3OCd (data component) and L3OCp (pilot component). These components are in phase quadrature with each other and L3OCd is delayed by 90° with a main lobe bandwidth of 20.46 MHz.

3.3 Galileo

Galileo is a global navigation satellite system developed by European space agency (ESA) and co-funded by ESA and the European Union. Galileo is designed as an independent system with respect to satellite constellation, ground segment, and operation for world-wide services. Galileo is compatible and interoperable with other radio navigation satellite systems, mainly with GPS as an example. The fully deployed Galileo system consists of 24 operational satellites and up to six active spares populated in three circular Medium Earth Orbit planes with an altitude of 23,222 km. Three independent CDMA signals are transmitted by all the Galileo satellites E1, E5, and E6. The E5 signal is further subdivided into two signals, denoted E5a and E5b (Galileo, 2008). Galileo provides enhanced distress localisation and call features for the provision of a search and rescue (SAR) service. The Galileo system started its initial services on 15 December 2016 and reached its FOC by 2021.
The Galileo system offers five global high-performance services (Galileo, 2019):

- Open service (OS): A free user charge service at E1-B/C, E5a, and E5b, set up for PNT services.
- Public regulated service (PRS) : A service restricted government-authorised users on E1-A and E6-A for sensitive applications that require a high level of service continuity. This service is more robust against interference and jamming.
- High accuracy service: A free access service complements the OS, delivers high accuracy data, and provides better ranging accuracy, enabling users to achieve sub-meter level positioning accuracy.
- Commercial authentication service: A service complementing the OS through the data-pilot pair E6-B/C, providing a controlled access and authentication function to the users.
- Search and rescue service (SAR): Europe's contribution to COSPAS-SARSAT, an international satellite-based search and rescue distress alert detection system.

3.3.1 Space segment

The Galileo constellation consists of 24 satellites and up to six active spares in three orbital planes in MEO at an altitude of 23,222 km. The orbital planes are inclined at 56° with respect to the equatorial plane and separated by 120° to cover the complete 360°. In each orbital plane, there are eight symmetrically spaced slots, and each slot contains a satellite. Two further satellites in each plane will be a spare, and on standby, any operational satellite fails. It takes approximately 14 h for a satellite to orbit the Earth. The reason for choosing such a structure for the Galileo constellation because of the very high probability that anyone anywhere in the world will always be in sight of at least four satellites and hence will be able to determine their position from the ranging signals broadcast by the satellites.

3.3.2 Ground segment

The transmitted signal from the satellite must be continuously monitored to guarantee signal quality. Calibration activities must be performed urgently if necessary. These functions are ensured by the ground control segment. The Galileo ground segment consists of the following parts (Galileo, 2019):

- Two Galileo control centres, ground control segment (GCS) and ground mission segment (GMS). GCS provides support for the management and control of satellite constellations, such as control and monitoring of satellites and payload. GMS determines the navigation and timing data part of the navigation messages and transmits it to the satellite via its C-band ground stations (e.g. satellite orbit and time synchronisation data, ionospheric correction model data, and other information transmitted in the navigation signals).
- A global network of Galileo sensor stations (GSS) collects and forwards Galileo signal measurements and data to the GCC in real time.
- A worldwide network of Galileo uplink stations (ULS) distributes and uplinks mission data to the Galileo constellation.
- Aworldwide network of Telemetry, tracking, and control stations (TT&C stations), which collect and forwards (receive and transmit) telemetry data generated by the Galileo satellites, and distribute and uplink the control commands required to maintain the Galileo satellites and constellation in nominal operational conditions.

3.3.3 Galileo signals

Galileo transmits several navigation signals at four different carrier frequencies. The frequency range allocated for Galileo is from 1.1 to 1.6 GHz in L band as follow:

- E1 signal, with carrier frequency centred at 1575.42 MHz
- E5 signal, with carrier frequency centred at 1191.795 MHz. E5 consist of two subsignals E5a and E5b, at 1176.45 and 1207.14 MHz respectively, multiplexed together through an Alternative BOC (AltBOC) scheme.
- E6 signal, with carrier frequency centred at 1278.75 MHz

Galileo OS signals are provided in either the single-frequency (SF) or dual-frequency (DF) mode. In the SF usage mode, the user tracks and processes the information provided by only one of the three OS frequencies, E1, E5a, or E5b. In the SF mode, there is a need to compensate for errors due to broadcast group delay (BGD) to compensate for specific satellite systematic delays between the different frequencies, and ionospheric compensation, to compensate for atmospheric delays of the signals. In the DF usage mode, the user tracks and processes the information provided by two of the three OS frequencies. The different OS dual-frequency usage modes are E1/E5a and E1/E5b. In dual-frequency usage modes, BGD compensation is not required, and ionospheric delays are measured directly by Galileo OS receivers, making use of the dispersive property of the ionosphere.

3.3.3.1 Galileo open service navigation data characteristics

The Galileo OS navigation data are provided in two different message types: free navigation (F/NAV) and integrity navigation (I/NAV) messages. The navigation data are transmitted only on the data component as a sequence of frames. A frame is a combination of several sub-frames, and sub-frames are the number of pages which is the basic structure for building navigation messages. Both E1-B and E5b-I signal components are modulated by the I/NAV navigation message type, whereas the E5a-I component is modulated by the F/NAV navigation message type. The navigation message contains the necessary information to enable the OS users to calculate the PVT solution. This includes (Kaplan and Hegarty, 2017):

- Galileo system time (GST). This includes two parameters, the time of week (TOW) , which is defined as the number of seconds. The TOW covers an entire week from 0 to 604799 seconds and is reset to zero at the end of each week. The second parameter is the Week number, which is an integer counter that counts the number of successive weeks from the GST start epoch. This parameter covers 4096 weeks (approximately 78 years).

- Clock correction parameters. Each Galileo satellite broadcasts its own clock correction data for all signals through the relevant signal, which includes the clock correction data for the TOW and clock drift correction coefficients.

- Ephemeris information. This provides precise information for computing the satellite positions in the sky (coordinates) with respect to the user receiver.

- Satellite signal health information.

- Ionospheric error correction. This includes the parameters are required for single-frequency receivers to correct the effect of the ionosphere on the transmitted signals.

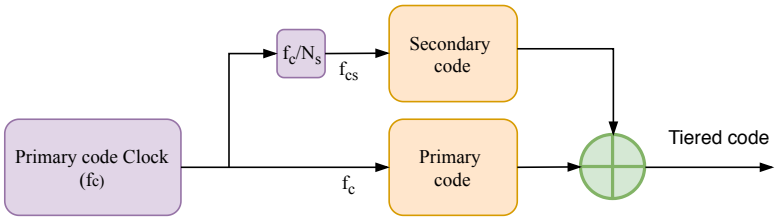
- Almanacs. This includes coarse information with reduced accuracy regarding the position of all active satellites in the constellation to assist the initial acquisition of the signals by the receiver. A predicted satellite health status is provided for each of these satellites, providing indications of the satellite's health and navigation data health. A reduced set of clock correction parameters is also provided in the almanac.

Likewise, the modernised signal in GPS and GLONASS, Galileo navigation message, employs a CRC technique to detect corruption in the received data. The CRC checksum is used to ensure errorless reception of the transmitted navigation bits. If the CRC checksum is not passed successfully, the received data must be discarded, and once a navigation message is received with a successful CRC, the user can then proceed with the data. The full details regarding the Galileo OS navigation message can be found in the Galileo OS SIS ICD document (Galileo, 2008).

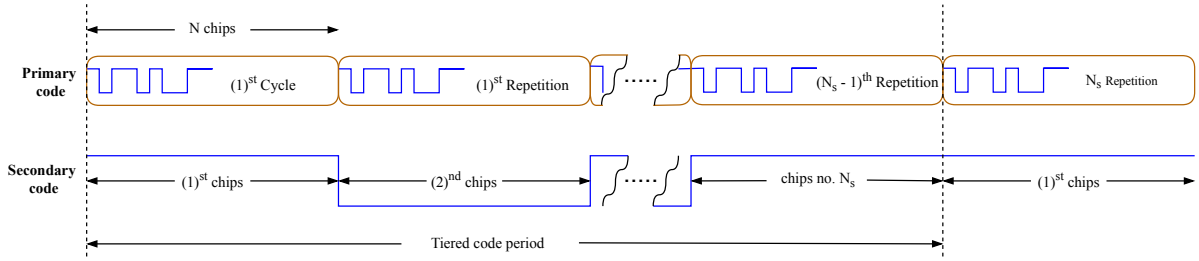
3.3.3.2 Galileo spreading codes

Galileo deploys very long codes to improve the code isolation of the CDMA technique. Longer codes produced better cross-correlation performance and robustness. However, longer codes also delay the acquisition process. To search for very long code lengths proposed for the new signals would be impractical, so the codes have been designed with escape routes. The most common one is called tiered code. This means that it is built in layers so that in case of a strong signal, the user can acquire on a simple layer with less acquisition time, only switching to the full-length code when required (Borre et al., 2007).

Tiered codes are the modulo-2 sum of the primary code with a secondary code. Usually, the primary codes are truncated Gold codes, with a greater number of chips in one sequence but shorter duration, while the secondary codes are pre-defined codes with fewer chips and longer durations. The secondary code sequence is used to modify successive repetitions of a primary code. Figure 3.10 illustrates a primary code of length N chips with chipping rate f_c, and a secondary code of length N_s chips with chipping rate $f_{cs} = f_c/N$.The secondary code chips are sequentially exclusive-OR ed with the primary code, and always one chip of the secondary code covers one period of the primary code. The primary codes can be LFSR-based sequences or optimised code sequences. Optimised codes need to be stored in memory and are therefore often called 'memory codes' (Galileo, 2008). Register-based codes used in Galileo are similar to gold codes, generated as combinations of two sequences, and are truncated to the desired length. These codes can be generated either using LFSR or stored in memory.



(a) Logical representation.



(b) Timing diagram.

Figure 3.10: Tiered codes generation.

3.3.3.3 E1 open service signal

The E1 OS signal consists of two components modulated by the CBOC(6,1,1/11) scheme. A data component (E1-B) modulated with the I/NAV navigation message at a data rate of 125 bps and a symbol rate (after FEC) of 250 sps, modulated with the CBOC(6,1,1/11,+) scheme. The second component is a data-less (not carrying navigation data) component (E1-C) modulated with a CB0C(6,1,1/11,-) scheme. The E1 signal deploys the QPSK modulation technique. Both E1-B and E1-C are added together and modulated into the in-phase component of the E1 carrier at 1575.42 MHz. On the other hand, the public restricted signal E1-A is modulated into the quadrature-phase component of the carrier. E1 OS deploys optimised primary memory codes. The data channel deploys primary

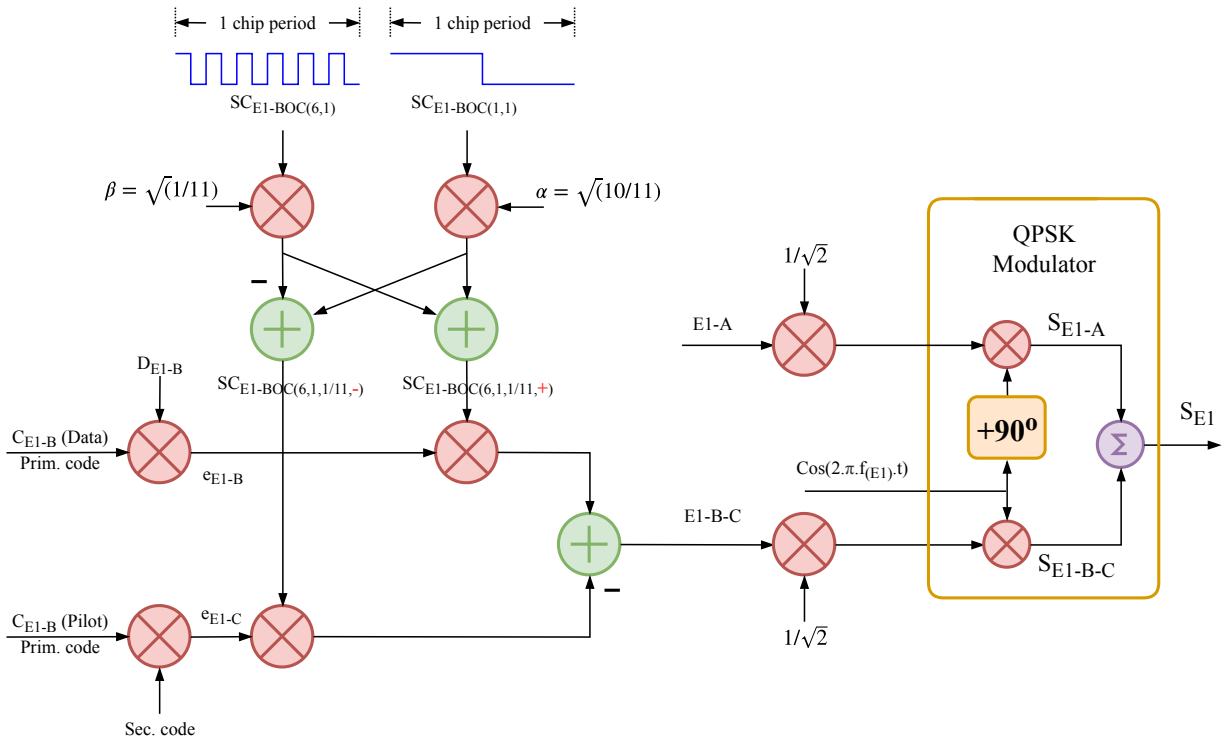


Figure 3.11: Galileo E1 signal modulation scheme.

codes of 4092 chips long with a 4 ms period at 1.023 Mcps. The pilot channel deploys tiered codes of 25 x 4092 chips, a 100 ms period with 1.023 Mcps, and the tiered codes are generated using 4092 chip long primary codes at 1.023 Mcps with 25 chips secondary code at a chipping rate of 250 cps. Pilot channels facilitate long integrations and improve the sensitivity, and tiered codes simplify the synchronization. The E1 OS modulation scheme is denoted as a composite binary offset carrier (CBOC) (Rodríguez and Ángel, 2008). CBOC is a weighted composition of BOC(1,1) and BOC(6,1), which is usually used for the CBOC signal as CBOC(6,1,ρ), where the parameter ρ is related to the power ratio of the BOC(6,1) in the CBOC signal. The E1 OS CBOC modulation scheme results from multiplexing a wideband signal BOC(6,1) and a narrowband signal BOC(1,1) with two different power levels; only 1/11 of the power is allocated, on average, to the BOC(6,1) component.

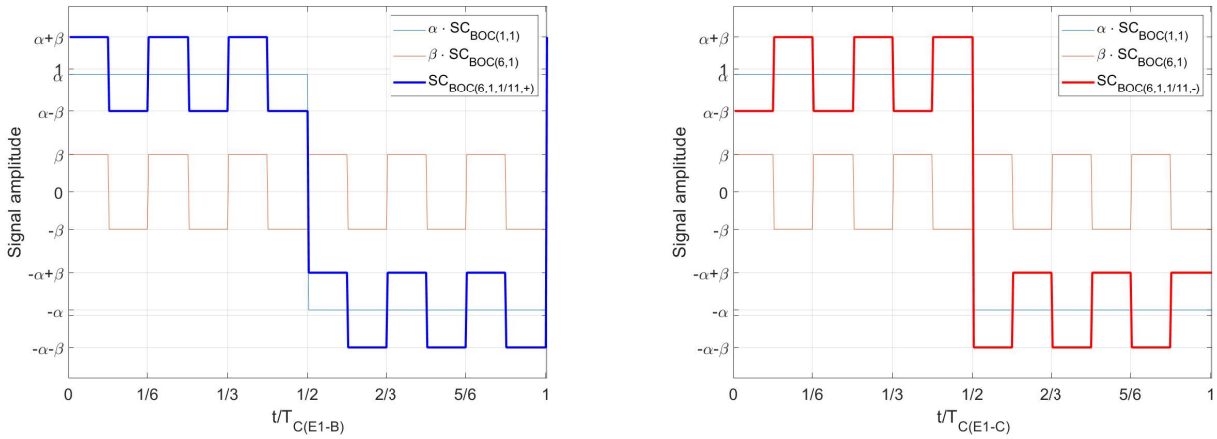
Figure 3.11 illustrates the Galileo E1 signal modulation scheme, where the E1 CBOC signal components are generated as follows:
The composite sub-carrier $SC_{BOC(6,1,1/11)}$ is a weighted combination of a $SC_{BOC(1,1)}$ and

a $SC_{BOC(6,1)}$ subcarriers and expressed as (Dovis et al., 2008):

$$SC_{BOC(1,1)}(t) = \begin{cases} sign\left[\sin\left(\frac{2\pi t}{T_C}\right)\right], & 0 \le t \le T_C, \\ 0, & \text{otherwise}, \end{cases} \tag{3.6}$$

$$SC_{BOC(6,1)}(t) = \begin{cases} sign\left[\sin\left(\frac{12\pi t}{T_C}\right)\right], & 0 \le t \le T_C, \\ 0, & \text{otherwise}, \end{cases} \tag{3.7}$$

where $T_C = 1/(1.023 \cdot 10^6)$ sec, is one chip duration. The in-phase composed sub-carrier



(a) In-Phase composite sub-carrier. (b) Anti-Phase composite sub-carrier.

Figure 3.12: One period of the CBOC sub-carrier.

$SC_{BOC(6,1,1/11,+)}(t)$ driven as:

$$SC_{BOC(6,1,1/11,+)}(t) = \alpha \cdot SC_{BOC(1,1)}(t) + \beta \cdot SC_{BOC(6,1)}(t) \tag{3.8}$$

where (+) denotes to in-phase
The anti-phase composed sub-carrier $SC_{BOC(6,1,1/11,-)}(t)$ driven as:

$$SC_{BOC(6,1,1/11,-)}(t) = \alpha \cdot SC_{BOC(1,1)}(t) - \beta \cdot SC_{BOC(6,1)}(t) \tag{3.9}$$

where (-) denotes to anti-phase.
One period of the in-phase composite sub-carrier for the E1-B signal component and One period of the anti-phase composite sub-carrier for the E1-C signal are shown in Figure 3.12. The parameters α and β are chosen such that the combined power of the $SC_{BOC(6,1)}$ on E1-B and the $SC_{BOC(6,1)}$ on the E1-C subcarrier components equals 1/11 of the total power of E1-B plus E1-C, before the application of any bandwidth limitation(Galileo, 2008). This yields
$\alpha = \sqrt{\frac{10}{11}}$ and $\beta = \sqrt{\frac{1}{11}}$
The binary signal e_{E1-B} (data component) is generated from the I/NAV navigation data stream D_{E1-B} and the primary ranging code C_{E1-B}, then modulated with the in-phase composite sub-carrier. The binary signal e_{E1-C} (pilot component) is generated from the primary ranging code C_{E1-B} including its secondary code, and then modulated with the anti-phase composite sub-carrier. During the modulation logic signal 1 is assigned to -1

at the signal level and 0 to +1.
The baseband composite signal E1-BC (E1 OS) is then generated according to equation 3.10 as follows:

$$s_{E1-BC}(t) = \frac{1}{\sqrt{2}} \left(e_{E1-B}(t) \cdot \left(SC_{BOC(6,1,1/11,+)}(t)\right) - e_{E1-C}(t) \cdot \left(SC_{BOC(6,1,1/11,-)}(t)\right) \right) \tag{3.10}$$

The pilot and data components are modulated onto the same carrier component, with a power sharing of 50 percent. Figure 3.13 shows the first six chips of Galileo E1 OS CBOC modulated baseband signal in the time domain, which belongs to the SV number one. The ACF of $CBOC_{(6,1,1/11)}$ can be written in terms of $BOC_{(1,1)}$ and $BOC_{(6,1)}$ as

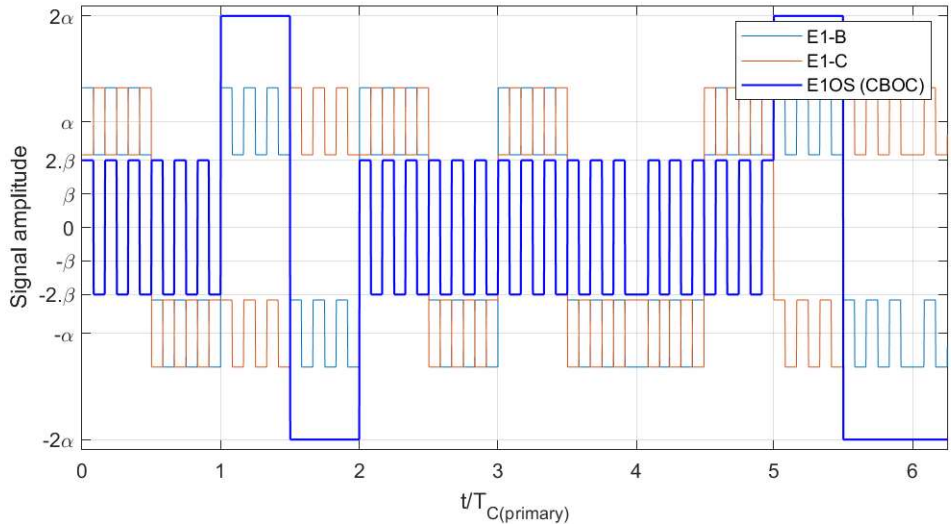


Figure 3.13: Galileo E1 OS baseband signal in time domain (first 6 chips SV 1).

(Rodríguez and Ángel, 2008) :

$$R_{CBOC(6,1,1/11)}(\tau) = \frac{10}{11} \cdot R_{CBOC(1,1)}(\tau) + \frac{1}{11} \cdot R_{CBOC(6,1)}(\tau) \tag{3.11}$$

As shown in Figure 3.14 the ACF of $BOC(1,1)$ has a sharper main peak than GPS C/A code BPSK(1), resulting in an improvement of the tracking error standard deviation of 2.4 dB approximately. The improvement is still more at the CBOC as the ACF peak has a sharper peak considering the data and pilot together. This represents an improvement of the tracking performance of approximately 1.03 dB with respect to BOC(1,1) and of 3.41 dB with respect to BPSK(1) (Rodríguez and Ángel, 2008).

The total E1 OS power spectral density can be defined as the linear combination of the PSDs of the two waveforms comprising the CBOC signal, namely BOC(1, 1) and BOC(6,1), weighted by the percentage of power that is put on the BOC(6,1) component and can be written as (Rodríguez and Ángel, 2008).

$$G_{OS_D+OS_P}(f) = 10/11.G_{BOC(1,1)}(f) + 1/11.G_{BOC(6,1)}(f) \tag{3.12}$$

Figure 3.15 illustrates the PSD of CBOC(6,1,1/11). The main envelope is the sum of 10/11.BOC(1,1) PSD and 1/11.BOC(1,1).

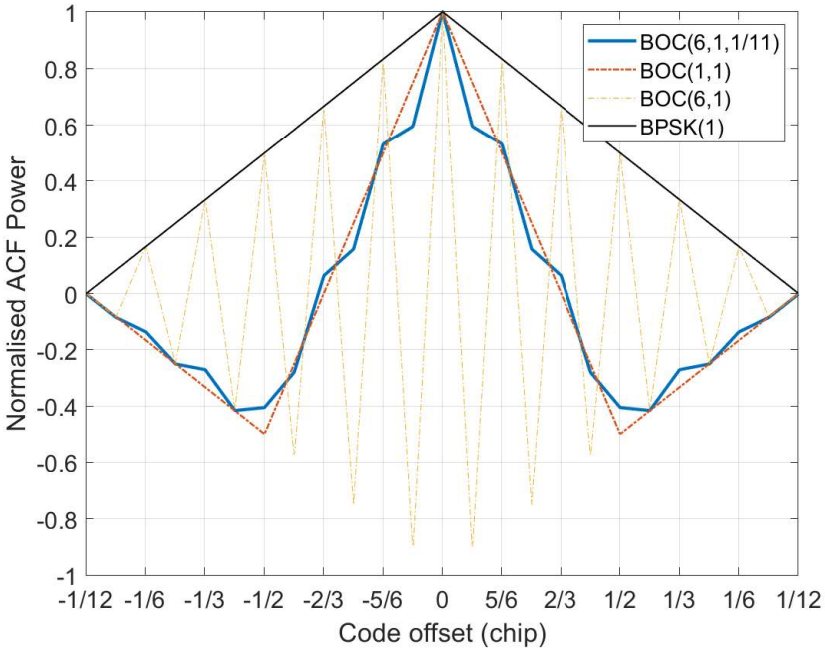


Figure 3.14: Normalised autocorrelation function comparison of CBOC(6,1,1/11), BOC(1,1), CBOC(6,1) and BPSK(1) computed over an infinite bandwidth.

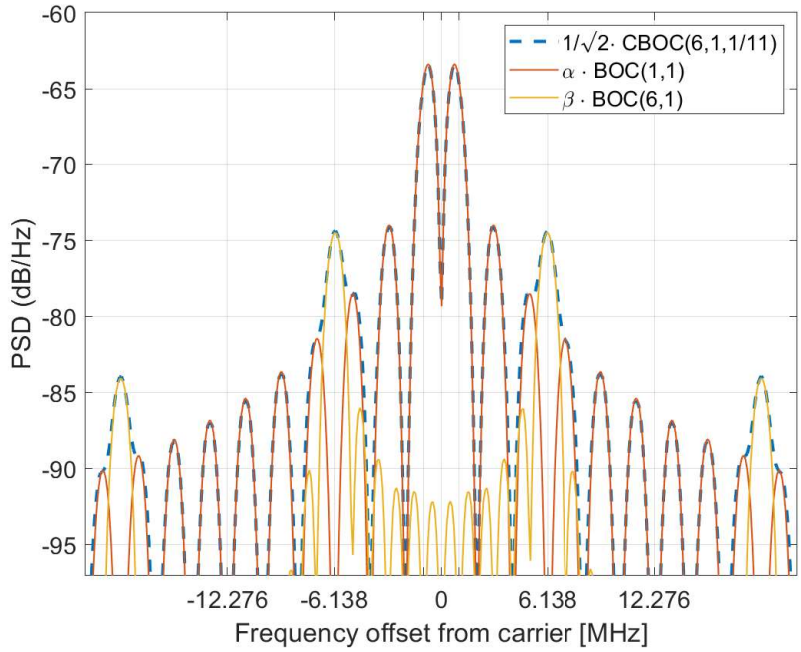


Figure 3.15: Galileo E1 OS baseband signal PSD as a function of the normalised offset from carrier frequency.

The E1 signal provides two services: the open service E1 OS and the public restricted service (PRS), known as E1-A. The PRS deploys the BOC(15,2.5) modulation scheme; both services are multiplexed into the E1 carrier using coherent adaptive subcarrier modulation (CASM) in order to maintain a constant power envelope, as shown in Figure 3.11. The transmitted signal can be written as (Ucar, 2010):

$$S_{E1}(t) = \sqrt{(2P)}.a1(s_{E1b} - s_{E1c})\cos(2\pi f_{E1}t) - \sqrt{(2P)}\left[a2.s_{E1a}(t) + a3.s_{IM}\right]\sin(2\pi f_{E1}t) \tag{3.13}$$

where $IM = s_{E1a}.s_{E1b}.s_{E1c}$
$a_1 = \sqrt{(2/3)}, a_2 = 2/3 and a_3 = 1/3$ which ensures a constant power envelop.
Approximately 11% of the total power is spent on the intermodulation product. Figure 3.16 shows the Galileo E1 signal PSD as a function of the normalised offset from the carrier frequency. As illustrated, the E1 Os with CBOC(6,1,1/11) modulated into the in-phase component of the carrier and the PRS signal with BOC(15,2.5) modulated into the quadrature-phase.

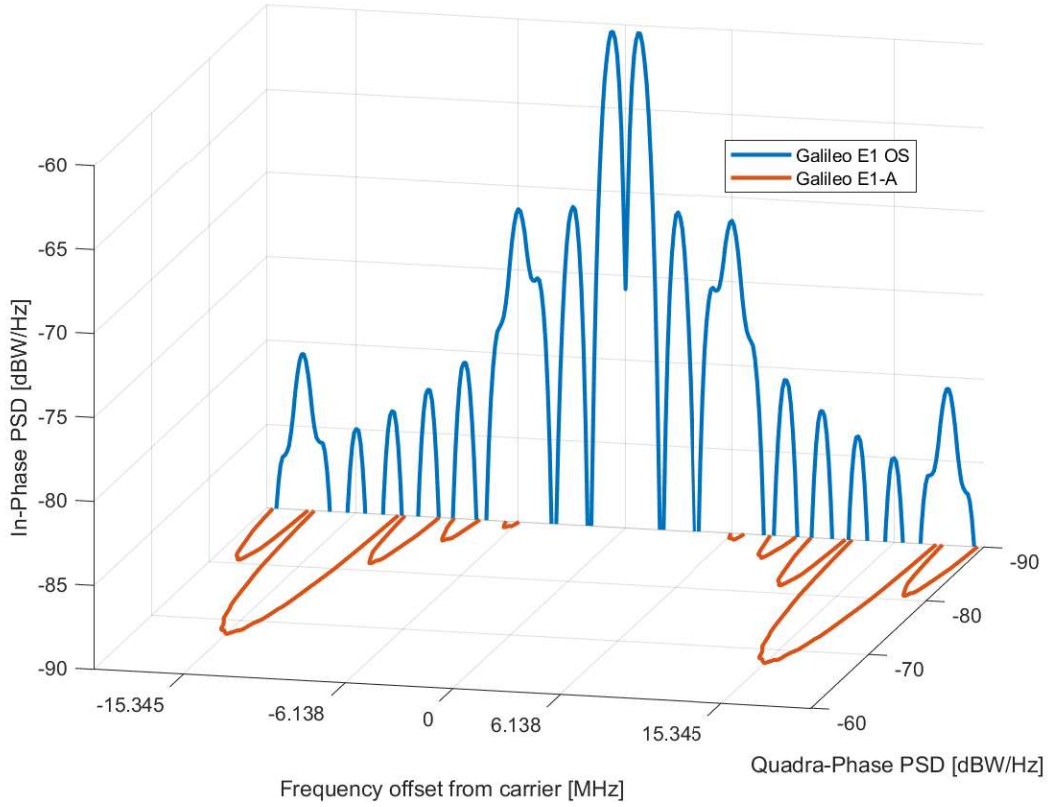


Figure 3.16: Galileo E1 signal PSD as a function of the normalized offset from the carrier frequency.

3.3.3.4 E5 signal

Galileo E5 deploys an AltBOC modulation scheme, which can be considered as a modified version of the BOC. In BOC modulation, the signal spectrum is split depending on the sub-carrier frequency. The BOC signal could not be correlated with a BPSK replica having a chip rate of the spreading code; to obtain the autocorrelation, the replica spreading code should be multiplied with a replica sub-carrier frequency of the original BOC signal, thereby increasing the complexity of the correlation engine and increasing power consumption. AltBOC solves the drawback of BOC modulation by providing spectral isolation between the two upper and lower components of the same composite signal; in other words, the signal spectrum is not split up, but only shifts to higher or lower frequencies and the BPSK correlation features are maintained, that is, each lobe can be

considered as an independent signal and its centre frequency is the amount of shift from the main carrier. The idea of AltBOC modulation is to perform the same process as BOC modulation, but the sub-carrier used is a "complex" sub-carrier. In this way, the signal spectrum is not split up, but only shifts to higher and lower frequencies (Rebeyrol et al., 2005). Similar to the BOC modulation, for simplicity, the AltBOC modulation is generally denoted as $AltBOC(m,n)$ where $m = f_s/1.023MHz$, $n = f_c/1.023MHz$, f_s is the subcarrier frequency and f_c is the chipping rate.

As presented in (Ries et al., 2001) the AltBOC signal is defined as the product of a spreading code sequence with a complex subcarrier. The AltBOC signal can be formed by two (only data signals) or four codes (data and pilot with 90° phase shift between the data and pilot components). In the case of four codes the signal is composed of data and pilot, expressed as follows:

$$S_{AltBOC}(t) = (C_{L^D} + jC_{L^P}).C_s(t) + (C_{U^D} + jC_{U^P}).C_s^*(t) \tag{3.14}$$

where C_L^D and C_L^p are the lower-side band data and pilot code respectively (in complex form) C_U^D and C_U^p are the upper-side band data and pilot code respectively (in complex form) $C_s(t)$ is the complex sub-carrier, complex sum of the rectangular cosine and sine-phased rectangular waveform, which can be written as follows:

$$C_s(t) = sign\left[\cos(2\pi f_s t)\right] + jsign\left[\sin(2\pi f_s t)\right] = C_r(t) + jS_r(t) \tag{3.15}$$

The aim of multiplying the lower complex code and the upper complex code by the complex sub-carrier and its conjugate is to shift the lower code by $+f_s$ and the upper code by $-f_s$. The composite signal obtained from complex multiplication in equation 4.14 does not provide a constant envelope and may be distorted due to nonlinear amplification.
A constant envelope-modified version of AltBOC modulation is proposed in (Godet, 2001). This is achieved by introducing a new signal called the intermodulation (IM) product. The IM component does not contain any information to be utilised by signal users. It is only added to bring the phase points of the constellation back to the circle so that the amplitude of the envelope remains constant. The modified constant envelope signal $S_{AltBOC}(t)$ is defined (Soellner and Erhard, 2003) as:

$$S_{AltBOC}(t) = \begin{cases} (C_{L^D} + jC_{L^P})\left[SC_{as}(t) - jSC_{as}(t - \frac{T_s}{4})\right] + \\ (C_{U^D} + jC_{U^P})\left[SC_{as}(t) + jSC_{as}(t - \frac{T_s}{4})\right] + \\ \left(\overline{C_{L^D}} + j\overline{C_{L^P}}\right)\left[SC_{ap}(t) - jSC_{ap}(t - \frac{T_s}{4})\right] + \\ \left(\overline{C_{U^D}} + j\overline{C_{U^P}}\right)\left[SC_{ap}(t) + jSC_{ap}(t - \frac{T_s}{4})\right] \end{cases} \tag{3.16}$$

where:
T_s is the period of the sub-carrier and the IM components are defined as follows:

$$\begin{cases} \overline{C_{L^D}} = C_{U^P}.C_{U^D}.C_{L^P} \\ \overline{C_{L^P}} = C_{U^D}.C_{U^P}.C_{L^D} \\ \overline{C_{U^D}} = C_{L^D}.C_{U^P}.C_{L^P} \\ \overline{C_{U^P}} = C_{U^D}.C_{L^D}.C_{L^P} \end{cases} \tag{3.17}$$

The signal and product sub-carrier components defined as follows:

$$SC_{as}(t) = \begin{cases} \frac{\sqrt{2}}{4} sign\left[\cos\left(2\pi f_s t - \frac{\pi}{4}\right)\right] + \frac{1}{2} sign\left(\cos\left(2\pi f_s t\right)\right) + \\ \frac{\sqrt{2}}{4} sign\left[\cos\left(2\pi f_s t + \frac{\pi}{4}\right)\right] \end{cases} \tag{3.18}$$

$$SC_{ap}(t) = \begin{cases} \frac{-\sqrt{2}}{4} sign\left[\cos\left(2\pi f_s t - \frac{\pi}{4}\right)\right] + \frac{1}{2} sign\left(\cos\left(2\pi f_s t\right)\right) - \\ \frac{\sqrt{2}}{4} sign\left[\cos\left(2\pi f_s t + \frac{\pi}{4}\right)\right] \end{cases} \tag{3.19}$$

Figure 3.17 shows the sub-carrier structure for the signal and product component. The

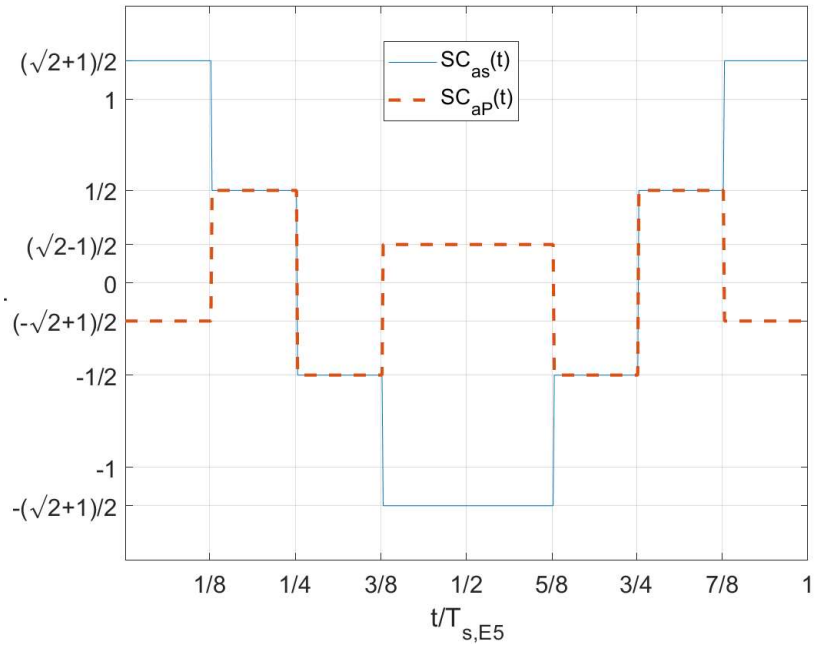


Figure 3.17: sub-carrier structure for the signal and product component

transmitted E5 signal from Galileo satellites is made of four components multiplexed together through an AltBOC(15,10) scheme (Figure 3.18) generated as follows:

- The binary signal e_{E5a-I} (data component) is generated from the F/NAV navigation data stream D_{E5a-I} with a data rate of 25 bps modulated with the ranging code C_{E5a-I}.
- The binary signal e_{E5a-Q} (pilot component) is generated from the ranging code C_{E5a-Q}. This signal dos not contain any navigation messages.
- The binary signal e_{E5b-I} (data component) is generated from the I/NAV navigation data stream D_{E5b-I} with a data rate of 125 bps, which is converted to a symbol rate of 250 sps modulated with the ranging code C_{E5a-I}.
- The binary signal e_{E5b-Q} (pilot component) is generated from the ranging code C_{E5b-Q}. This signal does not contain any navigation messages.

The signal E5 is generated with the AltBOC modulation using side-band square wave sub-carrier rate $SC_{E5} = (15x1.023MHz)$ according to the expression in equation 3.16. The codes belonging to the I and Q channels for each E5a and E5b can be found in the

Galileo interface control document (ICD) (Galileo, 2008). Finally the complex E5 base-band signal modulated into a complex carrier at 1191.795 MHz. The data channels of E5a and E5b are modulated into the in-phase carrier component, while the pilot channels of E5a and E5b are modulated into the quadrature-phase carrier component. The modulated (band-pass) signal $S_{E5}(t)$ is described in terms of its in-phase $s_{E5-I}(t)$ and quadrature $s_{E5-Q}(t)$ base-band components by the following generic expression (Galileo, 2008):

$$S_{E5}(t) = \sqrt{(}2P_{E5}) \left[s_{E5-I}(t).\cos(2\pi f_{E5} t) - s_{E5-Q}(t).\sin(2\pi f_{E5} t) \right] \tag{3.20}$$

where f_{E5} is the E5 carrier frequency and P_{E5} is the RF signal power.
The E5a and E5b signals can be processed independently by the user receiver as they are two separate QPSK signals with a carrier frequency of 1176.45 MHz and 1207.14 MHz respectively. Figure 3.18 illustrates the Galileo E5 signal modulation scheme. The power

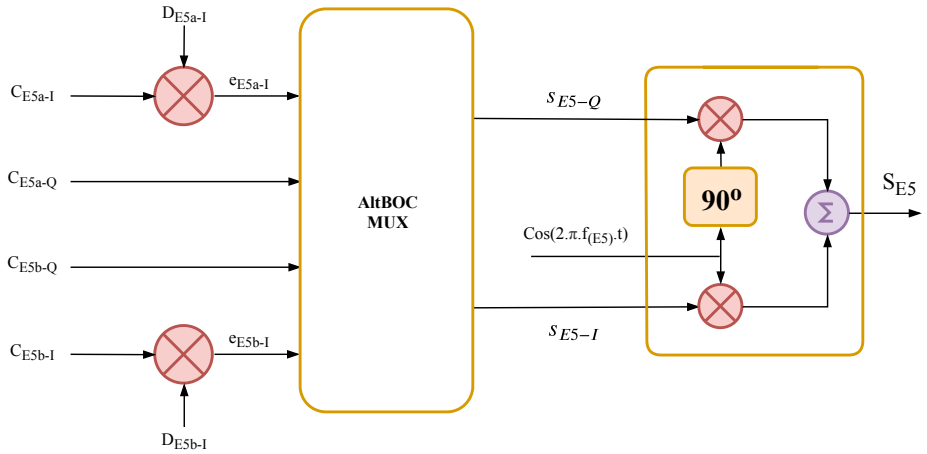


Figure 3.18: Galileo E5 signal modulation scheme.

spectral density of the modified even AltBOC modulation with constant envelope can be written as (Rodríguez and Ángel, 2008):

$$G_{AltBOC}^{even}(f) = \frac{4f_c}{\pi^2 f^2} \frac{\sin^2\left(\frac{\pi f}{f_c}\right)}{\cos^2\left(\frac{\pi f}{2f_s}\right)} \left[\cos^2\left(\frac{\pi f}{2f_s}\right) - \cos\left(\frac{\pi f}{2f_s}\right) - 2\cos\left(\frac{\pi f}{2f_s}\right)\cos\left(\frac{\pi f}{4f_s}\right) + 2 \right] \tag{3.21}$$

While for the odd case is:

$$G_{AltBOC}^{odd}(f) = \frac{4f_c}{\pi^2 f^2} \frac{\cos^2\left(\frac{\pi f}{f_c}\right)}{\cos^2\left(\frac{\pi f}{2f_s}\right)} \left[\cos^2\left(\frac{\pi f}{2f_s}\right) - \cos\left(\frac{\pi f}{2f_s}\right) - 2\cos\left(\frac{\pi f}{2f_s}\right)\cos\left(\frac{\pi f}{4f_s}\right) + 2 \right] \tag{3.22}$$

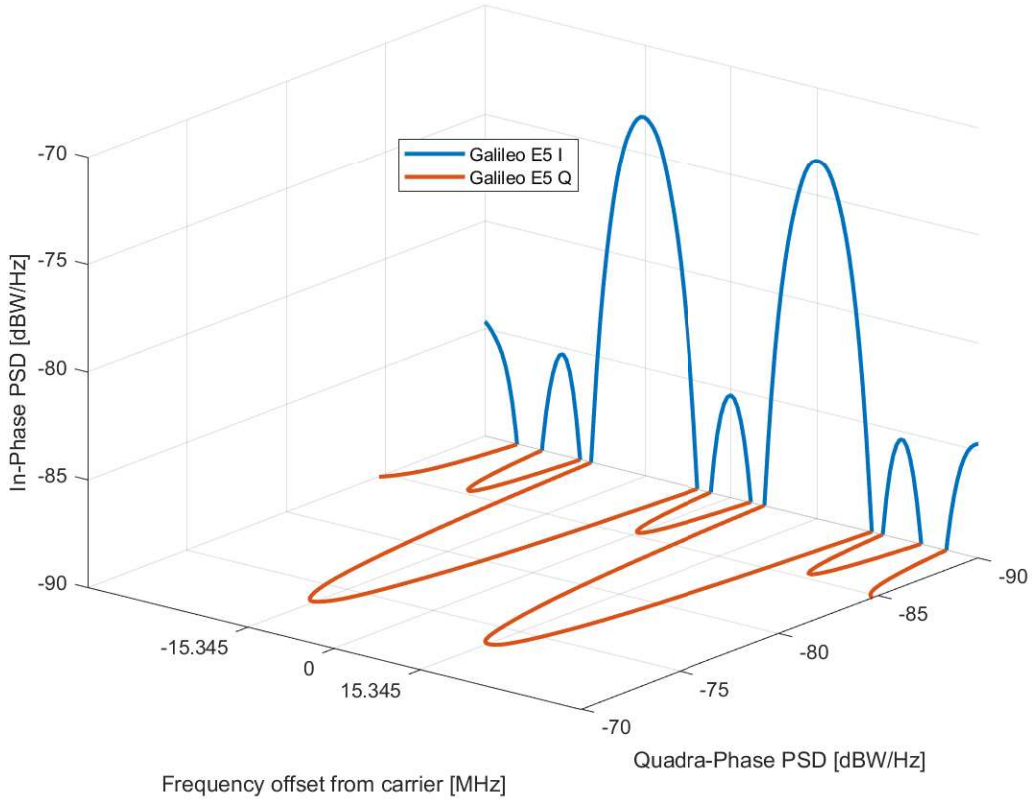


Figure 3.19: Galileo E5 signal PSD.

3.3.3.5 E6 signal

The E6 signal consists of two main components. A BPSK(5) commercial signal (CS) multiplexed with a $BOC_c(10,5)$ PRS signal. Likewise, E1 signals the IMP component used to ensure a constant power envelope (Ucar, 2010).

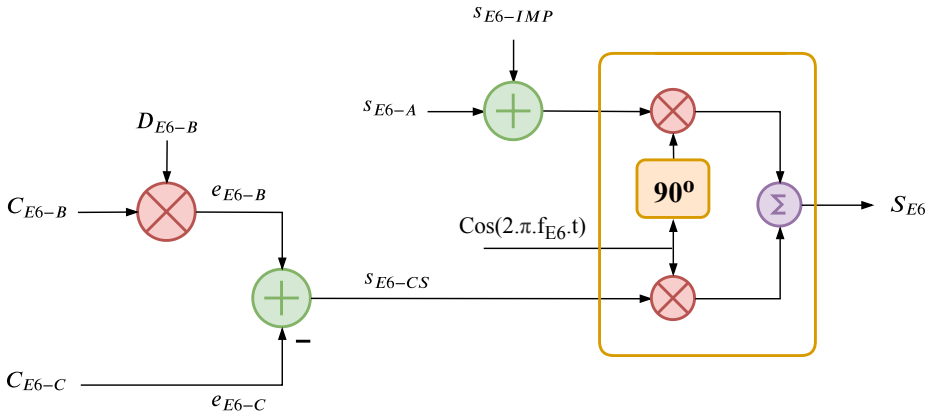


Figure 3.20: Galileo E6 signal modulation scheme.

As illustrated in Figure 3.20 the E6 CS signal is composed of two signals, B and C, generated as follows:

- The binary signal e_{E6-B} (data component) is a modulation product of the C/NAV navigation data stream D_{E6-B} at 1000 sps with the encrypted ranging code C_{E6-B} at 5.115 MChip/s.

- The binary signal e_{E6-C} (pilot component) generated from the ranging code C_{E6-C}

The primary code used in the B and C channels is memory codes defined according to the Galileo ICD document (Galileo, 2008). The data channel (B) primary code is encrypted with a licensed code, it cannot be accessed without a license key. The E6 CS signal is the sum of the binary signal (base-band) components $e_{E6\text{-}B}(t)$ and $e_{E6-C}(t)$ in anti-phase as defined in equation 3.23. Both the pilot and data components are combined on the same carrier component, with a power sharing of 50 percent.

$$s_{E6-CS}(t) = \frac{1}{\sqrt{2}}\left[e_{E6-B} - e_{E6-C}\right] \tag{3.23}$$

The power spectrum of the Galileo E6 signal is shown in Figure 3.21. The commercial

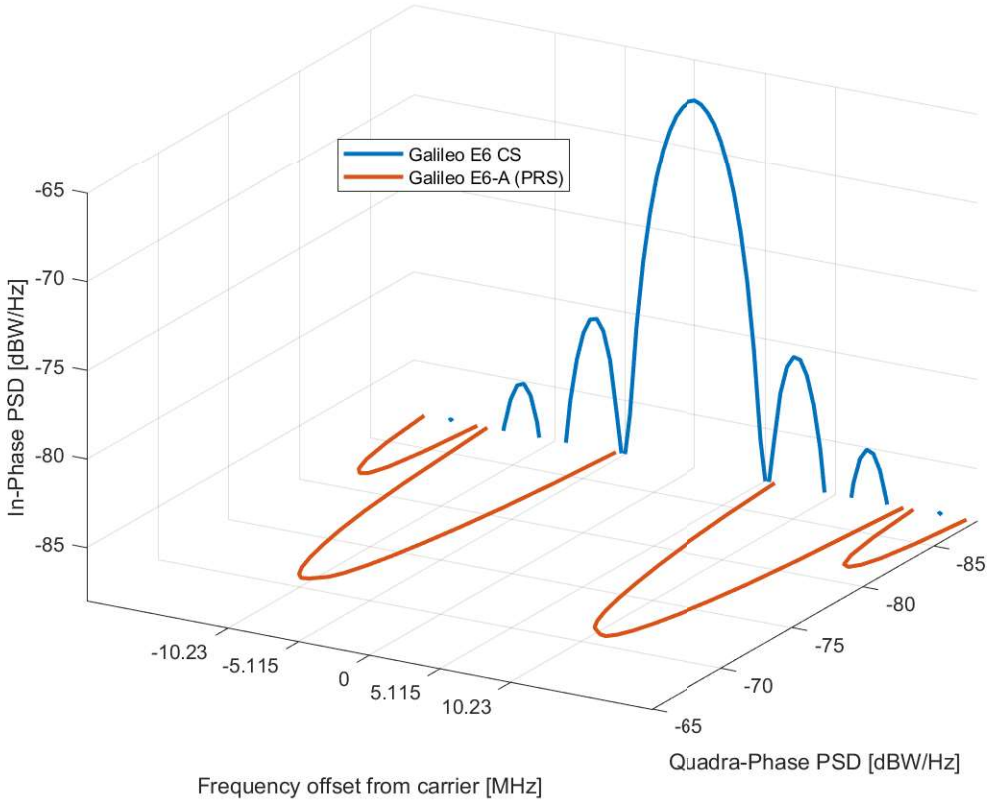


Figure 3.21: Galileo E6 signal PSD.

signal with a BPSK(5) occupies the in-phase component, the main lobe has a bandwidth of 5.115 MHz. The PRS signal with a $BOC_{cos}(10,5)$ is on the quadrature-phase, and it can be seen that the main lobe is offset from the centre frequency by 10x1.023 MHz carrier. The two signals are modulated into the E6 carrier at 1278.75 MHz. To ensure a constant power envelope, the IMP component is used.

3.4 BeiDou navigation satellite system

BDS is a system developed by China. BDS is the third independently developed GNSS after GPS and GLONASS. BDS is designed to be compatible and interoperable with other radio navigation satellite systems. BDS provides open service for PNT to users worldwide, while for the Asian-Pacific, BDS provides a more enhanced service. Smiler to

other GNSS BDS consists of a space, control, and user segment. The fully deployed BDS constellation consists of 30 satellites with backup satellites. The ground control system consists of a master control station, an upload station, and a montoring station (BeiDou, 2018a). The development of the BDS involves three steps (BeiDou, 2019a).

- BeiDou-1
  In 1994, the BeiDou navigation satellite experimental (demonstration) system started to develop a radio demonstration satellite service (RDSS) (Rothblatt, 1987) for China in 2000. This system is called the BeiDou-1 system (BDS-1). Two BDS-1 version satellites are allocated in GEO orbit using an active positioning scheme provided two-way short massage, positioning, and timing to the users in China (Bian et al., 2005). The system quality is enhanced by allocating a third satellite in GEO in 2003.

- BeiDou-2
  In 2004, the regional BeiDou navigation satellite system started, called BDS-2. The first MEO orbit satellite is launched in 2007, and the system is developed in 2012 with a constellation of 14 satellites, five satellites in GEO, five satellites in an inclined geosynchronous orbit (IGSO), and four satellites in MEO. The system later included a passive positioning technique. The scope of the system reached beyond China by providing users in the Asia-Pacific region with PVT solutions and SMS services.

- BeiDou-3
  BDS-2 is extended to a global service by deploying a third-generation satellite called BDS-3, which started in 2009 with full operational capability on 2020. BDS-3 satellite versions are compatible with BDS-2, providing users with continuous services. By using both active and passive positioning schemes, BDS-3 can provide global users with positioning, navigation, and timing, global short message communication, and international search and rescue services, and offer users in China and surrounding areas with regional short message communication, satellite-based augmentation, ground augmentation, and precise point positioning services.

3.4.1 Space segment

The space segment is a hybrid constellation consisting of satellites in three types of orbits, GEO, IGSO and MEO. The fully operational BDS consists of a multi-orbit constellation, including five GEO satellites, three IGSO satellites and 27 MEO satellites. The GEO satellites are allocated in an orbit with an altitude of 35,786 km and positioned at $58.75°E$, $80°E$, $110.5°E$, $140°E$ and $160°E$ respectively, covering China and the surrounding area. The IGSO satellites arepopulated in three planes with an inclination of 55° with respect to the earth equator line at an altitude of 35,786 km with one satellite per each plane. The MEO satellites are populated in three planes at an altitude of 21,528 km with 55° inclination to the equatorial plane (BeiDou, 2016).

3.4.2 Ground segment

The ground control segment of the BDS consists of master control station (MCS), time synchronisation/upload stations (TS/US), and monitor stations (MS). The MCS performs the following (BeiDou, 2018a):

- Process navigation signal recorded from TS/US and MS and generate navigation messages;
- Management and control of the constellation;
- Calculate the the satellites clock biases;
- Monitor the payloads and analyse abnormalities.

The TS/US continuously measurs satellite clock biases and uploads the satellites with navigation messages. The MS are responsible for recording navigation signals transmitted from the satellites and sending real-time recorded data to the MCS.

3.4.3 BeiDou signals

Similar to other GNSS signals BeiDou transmits navigation signals in three frequency bands, B1, B2 and B3 within the L-band as follows:

- Navigation signal B1, transmitted with a carrier frequency of 1561.098 MHz. This signal consists of to components in quadrature with respect to each other, B1I, provides an open service and B1Q provides an authorised service.
- To make BeiDou interoperable with Galileo and GPS, China announced the B1C signal. This signal will be transmitted from BDS-3 satellites with a frequency centred at 1575.42 MHz, employing MBOC modulation similar to the future GPS L1C and Galileo's E1.
- Navigation signal B2, transmitted with a carrier frequency of 1191.79 MHz, is composed of two sub-band signals referred to as B2a and B2b. The sub-band B2a centred at 1176.45 MHz, composed from a data component referred to as B2a-data and the pilot component B2a-pilot. The sub-band B2b, centred at 1207.14 MHz, is composed of a component referred to as B2b-I. This signal is similar to that of the Galileo E5 signal.
- Navigation signal B3, transmitted with a carrier frequency at 1268.52 MHz;

Table 3.2 shows the corresponding signals to each satellite version.

Table 3.2: Satellite version corresponding signal and navigation message.

Satellite Vesion	Transmitted Signal	NAV Message Types	NAV Data Rate	NAV Frame Length
BDS-2M BDS-2I	B1I, B3I	D1	50 bps	-
BDS-2G	B1I, B3I	D2	500 bps	-
BDS-3M	B1I, B3I	D1	50 bps	-
	B1C	B-CNAV1	100 sps	1800 symbols
	B2a	B-CNAV2	200 sps	600 symbols

3.4.4 BeiDou open service navigation message characteristics

The BDS open service provides four types of navigation messages; Table 3.2 shows the navigation messages carried by each signal for different versions of satellites and their corresponding data/symbol rate and the navigation frame length. The BDS open-service navigation message contains the following information:

- Satellite ephemeris parameters;
- Satellite clock bias parameters;
- Time correction parameters;
- Ionospheric correction parameters;
- Satellite health (if the signal provides service or not);
- BDS time-universal time coordinated (BDT-UTC) time synchronisation parameters;
- Almanac (coarse information regarding the constellation status).

3.4.5 BeiDou B1I signal

The BDS B1I signal is transmitted from all BDS satellite generations (BDS-2 and BDS-3) providing an open service on B1I and an authorised service on B1Q. The B1I signal is composed of a navigation message, ranging code and carrier.

3.4.5.1 Navigation data

The B1I signals transmitted from MEO/IGSO satellites provide the D1 navigation message format. The B1I signals transmitted from the GEO satellites provide the D2 navigation message format. The D1 navigation message rate is 50 bps, while the D2 navigation message rate is 500 bps. The D1 navigation message includes basic navigation information which defines the week number (WN), user range accuracy index (URAI) , satellite health flag (SatHl) , clock correction parameters, ephemeris parameters, and ionosphere parameters. D2 includes basic navigation and wide-area differential information (BDS integrity, differential, and ionospheric information) (BeiDou, 2019b).

3.4.5.2 Spreading code

Similar to the GPS C/A code, the ranging code C_{B1I} is a gold code, truncated before the last chip in the sequence. C_{B1I} has a chipping rate of 2.046 Mcps with a code length of 2046 chips repeated each millisecond. C_{B1I} is the modulo-2 sum of G1 and G2 linear patterns, obtained from two 11-bit shift registers with the following polynomials:

$$G1(X) : 1 + X + X^7 + X^8 + X^9 + X^{10} + X^{11}$$
$$G2(X) : 1 + X + X^2 + X^3 + X^4 + X^5 + X^8 + X^9 + X^{11}$$

with the initial values of G1 and G2 at 01010101010.

The relationship between the shift register taps and the exponents of the corresponding polynomial is referenced to the shift register input. As illustrated in Figure 3.22 the $G2i$

sequence is obtained by adding the output of two stages in the G2 shift register to a modulo-2 adder. Each specific taping of G2 is assigned to a satellite where i is the satellite number. The tap assignment of the G2 register for each satellite is defined in (BeiDou, 2019b).

The G1 and G2 registers are clocked at 2.046 MHz, reset at each 1 ms and restarted with the initial values of the G1 and G2 registers.
It is very important to know that both B1I and B2I signal which transmitted from the same satellite, utilise the same ranging code.

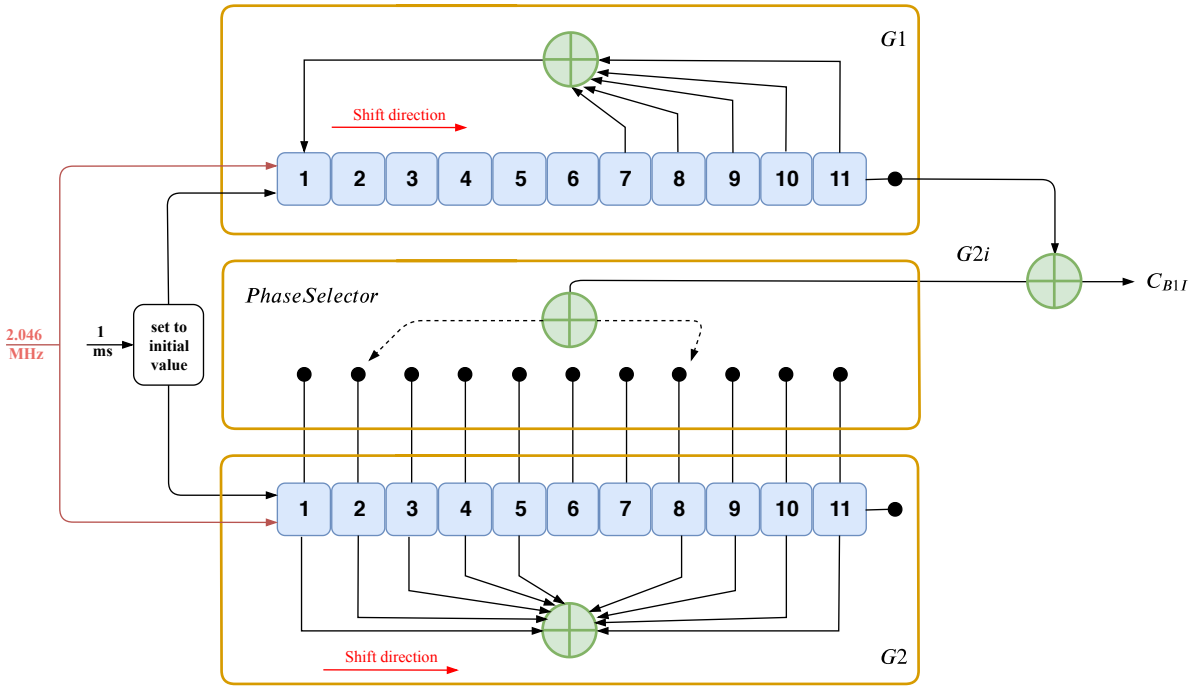


Figure 3.22: BDS C_{B1I} code generator.

3.4.5.3 Carrier

The carrier frequency is derived from the on-board satellite common reference clock. The carrier frequency of the B1I signal is 1561.098 MHz with a bandwidth of 4.092 MHz. Signal B1 consists of two components, I and Q which are in phase quadrature with respect to each other. The B1 signal employs the CDMA multiplexing mode, and the ranging code and NAV message are the sum of modulo-2 addition modulated into the carrier using the QPSK modulation scheme. The open service signal B1I is BPSK modulated into the I component of the B1 carrier, whereas the authorised service signal B1Q is BPSK modulated into the Q component of the B1 carrier. Both the I and Q components of the carrier are added together to form a QPSK modulation. Signal B1 can be defined as follows:

$$S^i_{B1} = A_{B1I}.C^i_{B1I}(t).D^i_{B1I}(t).\cos(2\pi f_1 t) + A_{B1Q}.C^i_{B1Q}(t).D^i_{B1Q}(t).\sin(2\pi f_1 t) \quad (3.24)$$

where:
superscript i represents the satellite number;
A_{B1I} and A_{B1Q} are the amplitudes of the B1I and B1Q signals respectively;
C_{B1I} and C_{B1Q} are the ranging codes of the B1I and B1Q signals respectively;

D_{B1I} and D_{B1Q} are data modulated on the ranging code of the B1I and B1Q signal respectively and f_1 is the carrier frequency of B1.

Figure 3.23 shows the B1 signal PSD, the two component sharing the signal power equally.

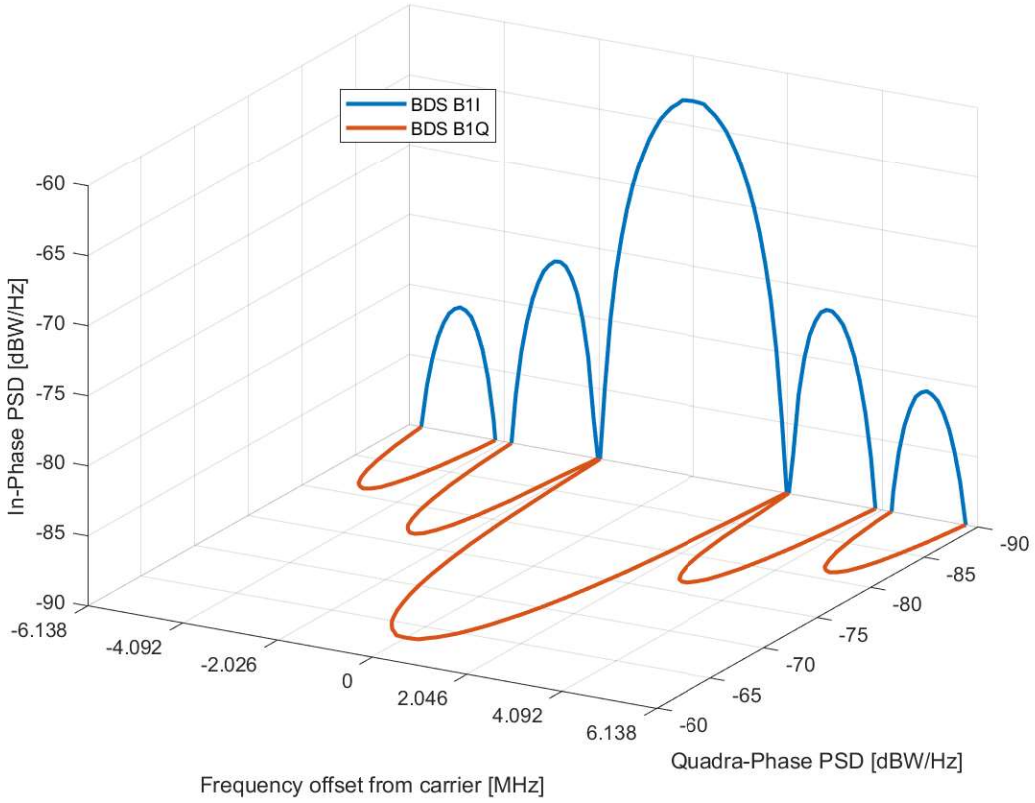


Figure 3.23: BeiDou $B1$ signal power spectrum density.

3.4.6 The new BeiDou B1C signal

To make BeiDou interoperable with Galileo and GPS, China announced the B1C signal. The B1C signal is transmitted by MEO satellites and the IGSO satellites of BDS-3 to provide open services and should not be transmitted by the GEO satellites. B1C has a bandwidth of 32.736 MHz and a centre frequency of 1575.42 MHz (BeiDou, 2017a). B1C deploys MBOC modulation similar to the future GPS L1C and Galileo E1. Similar to the other signals, B1C is composed of a navigation message, ranging code, and carrier.

3.4.6.1 Navigation data

The B1C signal carries a navigation message denoted as B-CNAV1. The B-CNAV1 navigation message is encoded using a cyclic redundancy check (CRC) for error correction. The length of each frame is 1800 symbols, and its symbol rate is 100 sps, so the transmission duration of one frame is 18 s. Each frame is divided into three subframes. Subframe 1 contains the PRN (satellite ID) and the second contains our (SOH). Subframe 2 provides information regarding the system time parameters, data issues, ephemeris parameters, clock correction parameters, and group delay differential parameters. Subframe 3 is divided into multiple pages, providing information regarding ionospheric delay correction

model parameters, earth orientation parameters (EOP), BDT-UTC time offset parameters, BDT-GNSS time offset (BGTO) parameters, midi almanac, reduced almanac, satellite health status, satellite integrity status flag, signal in space accuracy index, signal in space monitoring accuracy index, etc.. The system time parameters in B-CNAV1 contain Seconds Of Hour (SOH), HOW, and week number (WN)(BeiDou, 2017a).

3.4.6.2 sreading code

The B1C speading codes are tiered codes. As discussed in 3.3.3.2, tiered codes are products of the modulo-2 sum of a primary code with a secondary code. The duration of one chip of the secondary code has the same length as one period of the primary code. The starting time of the first secondary code chip is aligned with the start of the first chip of the secondary code.

The B1C primary codes for both the data and pilot components are truncated Weil codes. Weil codes obtained by modulo-2 addition of two unique circulating Legendre sequences $L(t)$ at different points, the code is truncated to obtain the desired code length. Each unique Weil code obtained by modulo-2 addition of $L(t)$ with $L(t+w)$ fixed Legendre sequence, where w is the Weil Index which represents the amount of shift. In general, a Weil code sequence of length N is defined as (BeiDou, 2017a):

$$W(t;w) = L(t) \oplus L\left((t+w)\ mod\ N\right)\ , t = 0, 1, 2, ..., N-1 \tag{3.25}$$

where
$L(t)$ is a Legendre sequence of length N, and w is the phase shift between two Legendre sequences. A legendre sequence $L(t)$ of length N is defined as

$$L(t) = \begin{cases} 0, & t = 0, \\ 1, & t \neq 0, \text{and if there exists an integer } x \text{ which makes } t = (x^2 \bmod \mathrm{N}), \\ 0, & \text{else}, \end{cases} \tag{3.26}$$

where, mod is a modulo division operation. The B1C primary codes (data and pilot) are 10230 chips long with a chipping rate of 1.023 Mcps, obtained by truncating a Weil code with a length of 10243 chips at the point p as given

$$c(n;w;p) = W\left((n+p-1)\ mod\ N;w\right)\ , n = 0, 1, 2, ..., N_0 - 1 \tag{3.27}$$

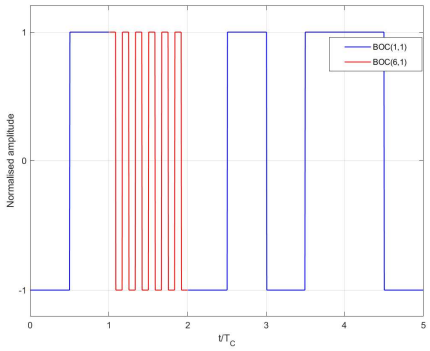
where, p is the truncation point in the range of $1 to N$,
N is the length of the Weil code and N_0 is the length of the truncated Weil code. The phase difference w and truncation point p for each satellite are defined in the BDS ICD (BeiDou, 2017a)

3.4.6.3 Carrier

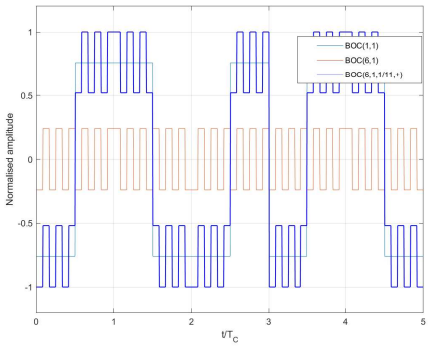
BDS employes a B1C signal for interoperability reasons, and the base band signal provides the same envelope as Galileo E1 OS and GPS L1C signals. All three signals (from different constellations) share a carrier frequency of 1575.42 MHz. The nominal bandwidth of the B1C is 32..736 MHz (BeiDou, 2017a). It is important to note that the same system can be used to process the B1C, L1C and E1 OS signals.

3.4.6.4 BeiDou B1C signal structure

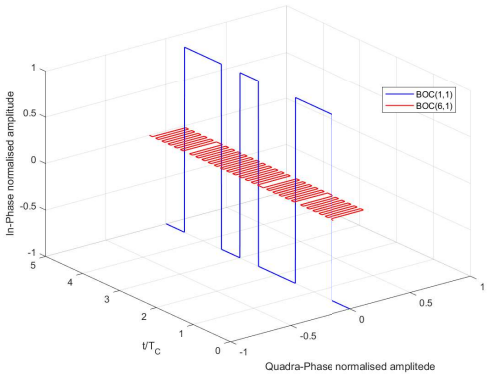
B1C signal deploys quadrature multiplexed BOC modulation (QMBOC) scheme. This scheme makes B1C compatible with other signals modulated into the same carrier frequency and better interoperability with L1C and E1 OS while maintaining independent intellectual property rights. QMBOC has the same receiving performance as E1 OS CBOC and L1C TMBOC. As shown in Figure 3.24, in the TMBOC scheme, BOC(1,1)



(a) Time multiplexed binary offset carrier.



(b) Composed binary offset carrier.



(c) Quadrature multiplexed binary offset carrier.

Figure 3.24: Normalised amplitude pilot component in time representation for time multiplexed, composed and quadrature multiplexed binary offset carrier.

is time multiplexed with BOC(6,1), while CBOC is the product of a weighted summation of BOC(1,1) and BOC(6,1). In the QMBOC scheme, BOC(1,1) is in phase quadrature modulated by BOC(6,1) (Yao et al., 2010).
The B1C baseband signal $s_{B1C}(t)$ is defined in a complex form as follows:

$$s_{B1C}(t) = s_{B1C-data}(t) + js_{B1C-pilot}(t) \tag{3.28}$$

In the following equations all modulo-2 additions are performed by multiplication after assigning each logic 1 to -1.0 in signal level and each logic 0 to 1.0 in signal level and $s_{B1C-data}(t)$ is the data component, modulo-2 addition of the navigation message $D_{B1C-data}(t)$ and the ranging code $C_{B1C-data}(t)$ modulated with BOC(1,1) subcarrier $SC_{B1C-data}(t)$. The data component power ratio is 1:3 with respect to the pilot component and can be expressed as:

$$s_{B1C-data}(t) = \frac{1}{2}\left[D_{B1C-data}(t) \cdot C_{B1C-data}(t) \cdot SC_{B1C-data}(t)\right] \tag{3.29}$$

The B1C data component sub-carrier $SC_{B1C-data}(t)$ is a BOC(1,1) subcarrier and is expressed as:

$$SC_{B1C-data}(t) = sign\left(\sin(2\pi f_{SC_{BOC11}} t)\right) \tag{3.30}$$

where, $f_{SC_{BOC11}} = 1.023$ MHz
$s_{B1C-pilot}(t)$ is the pilot component, generated from the ranging code $C_{B1C-pilot}(t)$ modulated with QMBOC(6, 1, 4/33) sub-carrier $SC_{B1C-pilot}(t)$. The pilot component can be expressed as:

$$s_{B1C-pilot}(t) = \frac{\sqrt{3}}{2}\left[C_{B1C-pilot}(t) \cdot SC_{B1C-pilot}(t)\right] \tag{3.31}$$

The B1C pilot component sub-carrier $SC_{B1C-pilot}(t)$ is a QMBOC(6,1,4/33) sub-carrier composed of a BOC(1,1) subcarrier, in phase quadrature with a BOC(6,1) subcarrier with a power ratio of 29:4 between the two components and is expressed as:

$$SC_{B1C_{pilot}}(t) = \sqrt{\frac{29}{33}}\left[sign\left(\sin(2\pi f_{SC_{BOC11}} t)\right)\right] - j\sqrt{\frac{4}{33}}\left[sign\left(\sin(2\pi f_{SC_{BOC61}} t)\right)\right] \tag{3.32}$$

where, $f_{SC_{BOC61}} = 6.138$ MHz, the equation 3.28 can be rewritten as follows:

$$s_{B1C}(t) = \begin{cases} \frac{1}{2}\left[D_{B1C-data}(t) \cdot C_{B1C-data}(t) \cdot sign\left(\sin(2\pi f_{SC_{BOC11}} t)\right)\right] + \\ \sqrt{\frac{1}{11}}\left[C_{B1C-pilot}(t) \cdot sign\left(\sin(2\pi f_{SC_{BOC61}} t)\right)\right] + \\ j\sqrt{\frac{29}{44}}\left[C_{B1C-pilot}(t) \cdot sign\left(\sin(2\pi f_{SC_{BOC11}} t)\right)\right] \end{cases} \tag{3.33}$$

Equation 3.33 is composed of three main parts; the first part represents the data component $s_{B1C-data}(t)$. The second part represents the real part of the complex pilot component $s_{B1C-pilot61}(t)$, obtained by modulating the pilot ranging code with a BOC(6,1) sub-carrier. The third part represents the imaginary part of the complex pilot component $s_{B1C-pilot11}(t)$, obtained by modulating the pilot ranging code with a BOC(1,1) subcarrier. The complex pilot component is a weighted component, in other words the power is not shared equally between the in-phase and quadrature-phase components. In general Equation 3.33 can be written as

$$s_{B1C}(t) = \frac{1}{2}\left[s_{B1C-data}(t)\right] + \sqrt{\frac{1}{11}}\left[s_{B1C-pilot61}(t)\right] + j\sqrt{\frac{29}{44}}\left[s_{B1C-pilot11}(t)\right] \tag{3.34}$$

Table 3.3 shows BDS B1C signal components, modulation, phase relationship and power ratio of each component accordingly.

Table 3.3: B1C signal Modulation characteristics

Component	Modulation scheme		Phase	Power ratio
$s_{B1C-data}(t)$	BOC(1,1)		0°	1/4
$s_{B1C-pilot11}(t)$	QMBOC(6,1, 4/33)	BOC(1,1)	90°	29/44
$s_{B1C-pilot61}(t)$		BOC(6,1)	0°	1/11

Figure 3.25 illustrates the PSD of QMBOC. The main envelope is the sum of the data

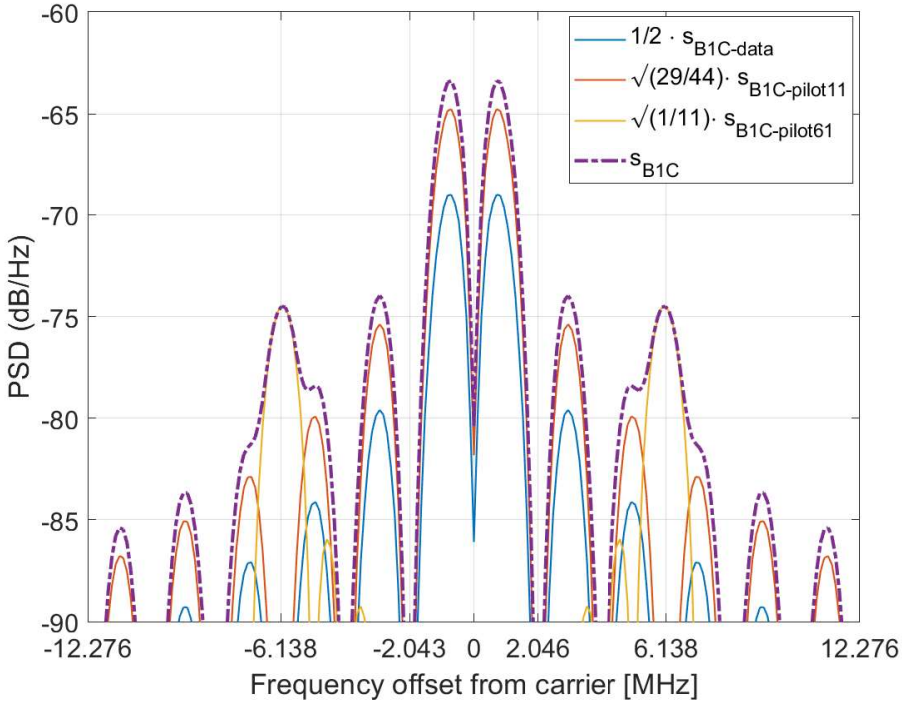


Figure 3.25: BeiDou $B1C$ signal power spectrum density as a function of the normalised frquency offset from carrier.

and pilot component PSD (Rodríguez and Ángel, 2008). The main envelop is a sum of $\frac{1}{2}.s_{B1C-dat}$, $\sqrt{29/44}.s_{B1C-pilot11}$ and $\sqrt{1/11}.s_{B1C-pilot61}$ The new B1 baseband signal $s_{B1}(t)$ which is transmitted with a 1575.42 MHz carrier composed of two components, the open service $s_{B1C}(t)$ and the authorised service $s_{B1A}(t)$. The open service is QMBOC modulated, while the authorised is BOC(14, 2) modulated. The power-normalised complex envelope of the modulated base-band signals can be expressed as:

$$s_{B1}(t) = s_{B1C}(t) + js_{B1A}(t) \tag{3.35}$$

The BDS B1C base-band signal $s_{B1C}(t)$ is modulated into the in-phase component of the carrier at $1575.42MHz$ while the authorised base-band signal $s_{B1A}(t)$ is modulated into the quadrature component. The carrier-modulated bandpass signal $S_{B1}(t)$ can be expressed as:

$$S_{B1}(t) = \sqrt{2P_x}\left[s_{B1C}(t)\cos(2\pi f_{B1}t) - s_{B1A}(t)\sin(2\pi f_{B1}t)\right] \tag{3.36}$$

where, f_{B1} is the carrier frequency and P_x is the signal power. It is important recognise that the expression $S_{B1}(t)$ representing the modulated base-band signal with the carrier while $s_{B1}(t)$ is the base-band signal.

Figure 3.26 illustrates the BeiDou $B1$ signal power spectrum density as a function of the normalised offset from the carrier frequency. The in-phase component represents QMBOC envelope which consists mainly of two components, BOC(1,1) and BOC(6,1) with different power levels, and it can be seen that the BOC(1,1) lobes shifted from the centre by 1.023 MHz while the BOC(6,1) lobes shifted from the centre by 6.138 MHz. The B1A signal populated in the quadrature phase, and the main lobe can be seen to shift from the centre frequency by 14.322 MHz.

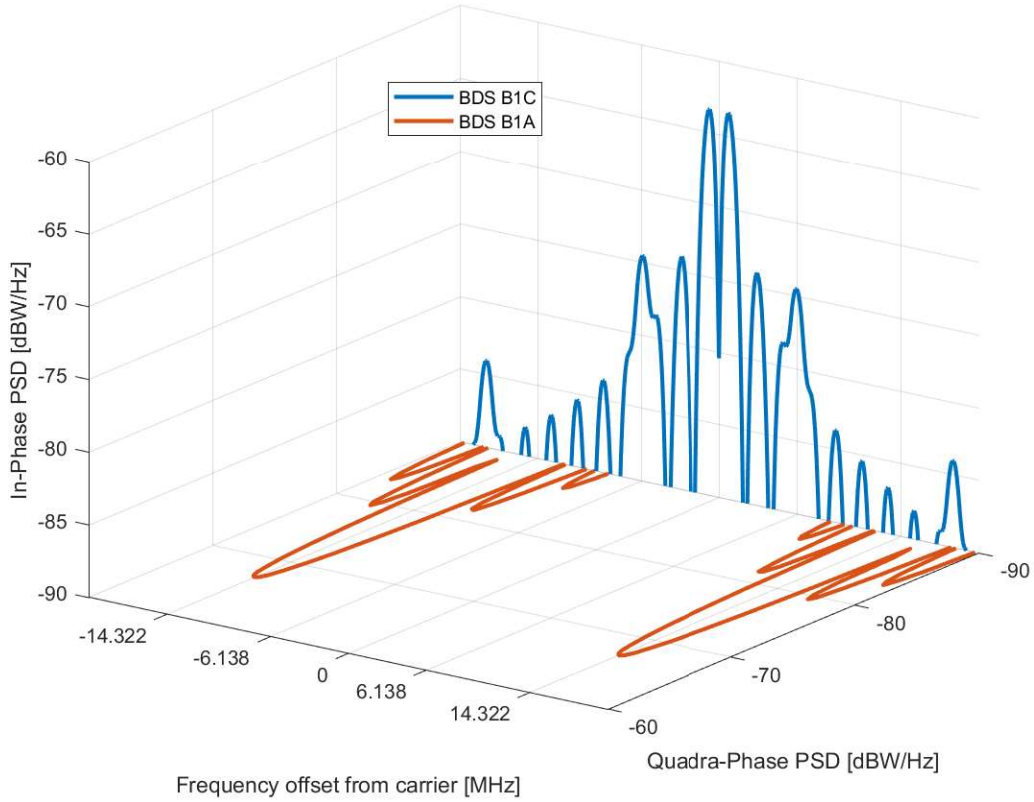


Figure 3.26: BeiDou $B1$ signal power spectrum density as a function of the normalized offset from the carrier frequency.

3.4.7 The new BeiDou B2 signal

Likewise B1C signal, China announced a new B2 signal for interoperability with Galileo E5 and GPS L5. The B2 signal is transmitted by MEO satellites and the IGSO satellites of BDS-3 to provide open services, and shall not be transmitted by the GEO satellites. B2 signal employs asymmetric constant envelope BOC (ACE-BOC) modulation scheme (Yao et al., 2016), more specifically ACE-BOC(15,10) which generates two side lobes, the lower lobe at centre frequency of 1176.45 MHz referred to as B2a and the upper lobe at centre frequency of 1207.14 MHz referred to as B2b (Lu et al., 2019), both lobes centred at 1191.795 MHz. Similar to Galileo's E5 AltBOC(15,10), each lobe consists of two components in the phase quadrature with respect to each other. The lower-side band B2a consists of a data component B2a-D and a pilot component B2a-P in phase quadrature. The data component B2a-D carries a binary navigation message at a rate of 200 sps. The upper-side band consists of the B2b-I component, carries the binary navigation message at a rate of 1000 sps. To date it is not clear if the B2b sub-signal consists of one or two phase quadrature components. The BDS ICD document defines only the B2b-I component. The two composite sub-signals B2a and B2b can either be received as two independent QPSK signals located on two different bands respectively, or together as a wideband signal.

ACE-BOC has much higher design flexibility than AltBOC in terms of the number of signal components, power ratio between the components, and hardware complexity. It can combine four or fewer signals with different power levels at two different frequencies into a spectrum-split signal in which each side band consists of no more than two codes (Yao et al., 2016). Signal B2 is composed of the navigation message, ranging code and

carrier

3.4.7.1 Navigation data

The B2a signal transmits the B-CNAV2 navigation message. A maximum of 63 message types can be defined for the B-CNAV2 navigation message. Currently, eight valid message types have been defined: message types 10, 11, 30, 31, 32, 33, 34, and 40. It is very important to know that the user should recognise its message type every time a B-CNAV2 navigation message is received by checking the message type field provided in the navigation message. The B-CNAV2 message data are modulated on the data component B2a-I after CRC encoding for error correction. The frame length of the B-CNAV2 navigation message is 288 bits before error correction encoding and 600 symbols after error correction encoding at a symbol rate of 200 sps. The navigation message contains information about the satellite number (PRN), message type, second of week (SOW), message data participate in the CRC calculation, and some useful parameters such as ephemeris, clock correction, ionospheric delay correction, BDT-UTC time offset, and almanac. The transmission of each frame is 3 s. More detailed information regarding the B-CNAV2 navigation message can be found in (BeiDou, 2017b) (BeiDou, 2017c).

The B2b signal transmits the B-CNAV3 navigation message. Currently, three valid message types have been defined: message types 10, 30, and 40. The user should recognise the message type every time a B-CNAV3 navigation message is received by checking the message type field provided in the navigation message. The B-CNAV3 message data are modulated on the data component B2b-I after CRC encoding for error correction. The frame length of the B-CNAV3 navigation message is 486 bits long before error correction encoding and 1000 symbols after error correction encoding at a symbol rate of 1000 sps. The B-CNAV3 navigation message includes basic navigation information and global basic integrity information. More detailed information regarding the B-CNAV2 navigation message can be found in (BeiDou, 2019c).

3.4.7.2 Spreading code

The B2a sprading codes are tiered codes. As discussed earlier in 3.3.3.2, tiered codes are products of the modulo-2 sum of a primary code with a secondary code. The duration of one chip of the secondary code has the same length as one period of the primary code. The starting time of the first secondary code chip is aligned with the start of the first chip of the secondary code.

Both data and pilot components of B2a primary codes are expanded gold code, 10230 chips long with a chipping rate of 10.23 Mcps, repeating each one ms. Each primary code is generated by expanding the Gold code that is generated by shifting and modulo-2 addition based on two 13-stage linear feedback shift registers. The polynomials for B2a data component primary code registers are defined as (BeiDou, 2017b):

$$G1(X) : 1 + X + X^5 + X^{11} + X^{13}$$
$$G2(X) : 1 + X^3 + X^5 + X^9 + X^{11} + X^{12} + X^{13}$$

while the polynomials for B2a pilot component primary code registers are defined as (BeiDou, 2017b):

$$G1(X) : 1 + X^3 + X^6 + X^7 + X^{13}$$
$$G2(X) : 1 + X + X^5 + X^7 + X^8 + X^{12} + X^{13}$$

The start initial bit value of G1 for both B2a data and pilot are all "1". The initial bit values of register G2 for both B2a data and pilot components are defined in the BDS ICD document (BeiDou, 2017b). At the start of each primary code period, both registers G1 and G2 are reset to their corresponding initial bit values. Further, register G2 is reset at the end of the 8190^{th} chip in each period of a primary code. A primary code with a length of 10230 chips is finally obtained by repeating the above procedure.

The secondary codes of the B2a data components are the same for all satellites, with fixed 5-bit values of 00010 in binary with a period of 5 ms. The secondary codes for the B2a pilot components are different, they are truncated Weil codes of 100 chips long with a period of 100 ms. The code parameters (phase difference and truncation point) of the secondary codes of the B2a pilot components are defined in the BDS ICD document (BeiDou, 2017b).

The B2b-I ranging codes are only primary codes generated in the same way as B2a primary codes at a chipping rate of 10.23 Mcps, 10230 chips long and 1 ms duration. The initial bit value of G1 for B2b-I is all "1". The initial bit values of register G2 for B2b-I are defined in the BDS ICD document (BeiDou, 2017b). The generation polynomials for all B2b-I codes are as follows:

$$G1(X) : 1 + X + X^9 + X^{10} + X^{13}$$
$$G2(X) : 1 + X^3 + X^4 + X^6 + X^9 + X^{12} + X^{13}$$

3.4.7.3 Carrier

BDS employs the new B2 signal for interoperability reasons; the base band signal B2a provides the same envelope as Galileo E5-a and GPS L5 signals. All three signals (from different constellations) share a carrier frequency of 1176.45 MHz. The nominal bandwidth of B2a is 20.46 MHz (BeiDou, 2017b). It is important to point out that the same system can be used to process B2a, L5, and E5a signals; they can be considered as QPSK(10) modulated signals.

B2b provides the same envelope as the Galileo E5-b signal. The two signals (from different constellations) share a carrier frequency of 1207.14 MHz. The nominal bandwidth of B2b is 20.46 MHz (BeiDou, 2019c). The same system can be used to process B2b and E5b signals; they can be seen as a QPSK(10) modulated signal.

3.4.8 The new BeiDou B3 signal

The legacy B1I at a centre frequency of 1561.098 MHz and B3I at a centre frequency 1268.52 MHz signals will still be transmitted by all the BDS-3 satellites (Lu et al., 2019). The BDS B3 is consists of two components. The open signal B1I with BPSK(10) is modulated into an in-phase component. An authorised signal with BPSK(10) is modulated into the quadrature phase component. Both signals form a QPSK(10) modulated signal. A new modernised signal B3A is transmitted beside the earlier B3 signal. B3A is a BOC(15,2.5) modulated signal with two components in phase quadrature, data and pilot. Dual-QPSK (Zhang, 2013) modulation scheme is proposed two combine the B3 and B3A with equal power for the subsequent launches (Kaplan and Hegarty, 2017).

BDS B3I is the only open service in the B3 band, this signal is transmitted from all BDS satellite generations (BDS-2 and BDS-3). The B3I signal is composed of a navigation message, ranging code, and carrier.

3.4.8.1 Navigation data

The B3I signals transmitted from MEO/IGSO satellites broadcast the D1 navigation message format. The B3I signals are transmitted from GEO satellites broadcasting the D2 navigation message format. The D1 navigation message rate is 50 bps, while the D2 navigation message rate is 500 bps. The D1 navigation message includes basic navigation information which defines the week number (WN) URAI satellite health flag (SatHl), clock correction parameters, ephemeris parameters, and ionosphere parameters. D2 includes basic navigation and wide-area differential information (BDS integrity, differential, and ionospheric information) (BeiDou, 2018b).

3.4.8.2 Ranging code

Similar to the BDS B1I code, the ranging code C_{B3I} is a truncated gold code. C_{B3I} has a chipping rate of 10.23 Mcps with a code length of 10230 chips repeated every millisecond. C_{B3I} is the modulo-2 sum of G1 and G2 linear patterns, obtained from two 13-bit shift registers with the following polynomials:

$$G1(X) : 1 + X + X^3 + X^4 + X^{13}$$
$$G2(X) : 1 + X + X^5 + X^6 + X^7 + X^9 + X^{10} + X^{12} + X^{13}$$

The full length sequence generated by each generator is 8191 chips. The initial phase value of G1 sequence is set to all '1' at the start of each 1 ms or when the G1 sequence register value is at '1111111111100', this process generates a sequence of 8190 chips instead of 8191 chips. The initial phase value of the G2 sequence is set to different initial phase values at the start of each 1 ms, each phase value corresponds to a satellite, and the different phase assignments for the satellites for the G2 sequence are defined in (BeiDou, 2018b). The C_{B3I} with a period of 10230 chips is obtained by Modulo-2 addition of G1 and G2 sequences.

3.4.8.3 Carrier

The carrier frequency of the B3 signal is 1268.52 MHz in L-band with a signal bandwidth of 20.46 MHz. Legacy signal B3 consists of two components, I and Q which are in phase quadrature with respect to each other. The B3 signal employs the CDMA multiplexing mode, the ranging code and NAV message are the sum of modulo-2 addition modulated into the carrier using the QPSK modulation scheme.

The open service signal $B3I$ is BPSK modulated into the I component of the B3 carrier while the authorised service signal $B3Q$ is BPSK modulated into the Q component of the B3 carrier. Both the I and Q components of the carrier added together forming a QPSK modulation. Signal B3 can be defined as follows:

$$S^i_{B3} = A_{B3I}.C^i_{B3I}(t).D^i_{B3I}(t).\sin(2\pi f_3 t) + A_{B3Q}.C^i_{B3Q}(t).D^i_{B3Q}(t).\cos(2\pi f_3 t) \quad (3.37)$$

where
Superscript i represents the satellite number;
A_{B3I} and A_{B3Q} are the amplitudes of the B3I and B3Q signals respectively;
C_{B3I} and C_{B3Q} are the ranging codes of B3I and B3Q signals respectively;
D_{B3I} and D_{B3Q} are data modulated on the ranging code of B3I and B3Q signals respectively;

f_3 is the carrier frequency of B3.
Figure 3.27 shows the B3 signal PSD, the two components share the signal power equally.

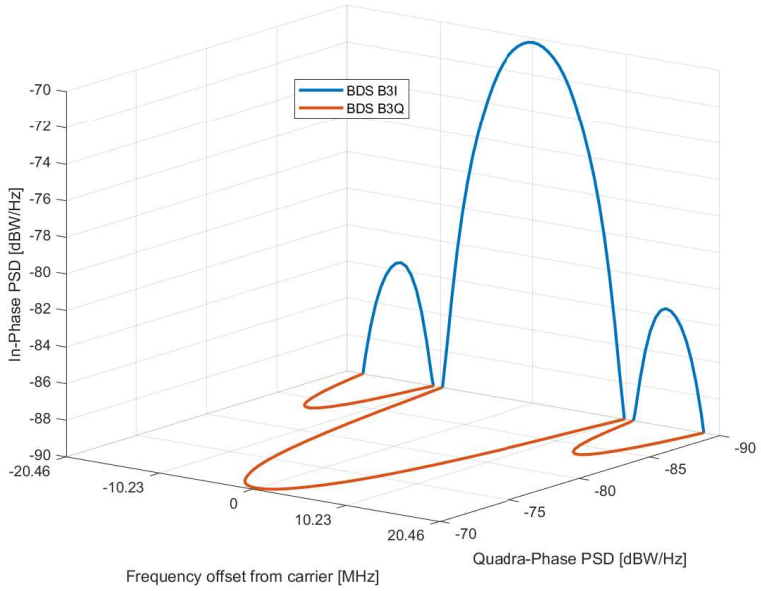


Figure 3.27: BeiDou $B3$ signal power spectrum density.

3.5 Multi-constellation and multi-frequencies system

Receivers that have access to a highest number of constellations and frequencies offer the best positioning availability, accuracy and resilience even in challenging environments. The more signals the receiver can access, the more accurate and reliable the computed position will be. In environments such as urban, the view of the sky is often partially blocked by buildings. This means that the number of visible satellites significantly drops. Multi-constellation and multi-frequencies (MCMF) system GNSS receivers which track all the signals from all possible satellite systems, see as many satellites as possible, which helps them to be robust in such challenging conditions. In addition to robust performance in difficult environments there are other advantages of a MCMF receiver.

- Mitigation of ionospheric errors: Using multi-frequency receivers is the most effective way to remove ionospheric error from the position calculation. Ionospheric error varies with frequency so it impacts the various GNSS signals differently. By comparing the delays of two GNSS signals, L1 and L2, for example, the receiver can correct for the impact of ionospheric errors.

- Radiofrequency interference robustness: Interference happens when other signals on the same frequency overpower GNSS signals. Interference can be Intentional caused by radio amateurs or non-intentional by the other satellite constellations and it usually effects one of the GNSS frequencies at a time. The frequency diversity provided by using multiple frequencies allows the receiver to switch to another frequency if interference on one frequency is detected.

- Better multipath rejection: The L1 signal, which is used in single frequency receivers is susceptible to multipath. Multipath is the distortion of direct line-of-sight signals

as they are contaminated by identical signals reflected from objects such as buildings, cars or trees. New GNSS signals such as GPS L5, Galileo L1BC and particularly Galileo E5-AltBoc are inherently more robust to multipath, so a receiver using these signals will suffer less from multipath errors.

- Better accuracy: Having access to many satellites creates redundancy, which enables a statistical analysis of the satellites and their signals. Having such statistical information allows the receiver to detect and remove accidental faults of ranging signals for improved positioning accuracy.

- Additional spoofing detection: Receivers that make use of multiple frequencies can use these frequencies for additional spoofing checks. By comparing range (distance to satellite) information from various signals, anomalies can be detected and flagged.

On the other hand migrating to MCMF will add hardware and software challenges to the design and development of the GNSS receivers. At this level there is no compatibility and interoperability between the different constellations, there will be requirements for using multiple antennas and front-ends due to the frequency diversity, this will lead to an increase in the size, weight, power and cost (SWaP-C) of the GNSS receiver. In parallel this will require more complicity in software design such as more computation burden in acquisition, tracking and PNT solution.

3.6 Summary

The GNSS consists of four main satellite systems: GPS, GLONASS, Galileo, and BDS. Each system consists mainly of three segments: space, control, and user segments. These segments are almost similar in all four systems, which together form GNSS.

GPS is the first GNSS system launched in the 1970s by the United States Department of Defense. It deploys a constellation of 27 satellites. The GLONASS is operated by the Russian government. The GLONASS constellation consists of 24 satellites. Galileo is a civil GNSS system operated by the European Global Navigation Satellite Systems Agency (GSA). Galileo used 27 satellites. The full constellation reached the FOC by the end of 2020. BeiDou is a Chinese navigation satellite system. The system consists of 35 satellites. Regional services became operational in 2012. BeiDou extended to provide global coverage and achieved FOC by the 2021.

The space segment consists of GNSS satellites orbiting the earth above 20,000 km approximately. Each GNSS has a dedicated constellation of satellites arranged in orbits to provide the desired global coverage.

The control segment comprises a ground-based network of master control stations, data uploading stations, and monitoring stations. The master control station adjusts the satellites, monitors the transmitted signals, and uploads satellites using navigation data.

The user segment consists of equipment that processes the received signals from the GNSS satellites to derive the PNT solutions. The equipment ranges from smartphones to sophisticated specialised receivers used for survey and mapping applications.

The legacy GPS C/A open signal is transmitted on L1 with a carrier component of 1575.42 MHz. This signal deploys a gold-code spreading sequence and carries LNAV data. The GPS L2C open signal with a carrier component of 1227.6 MHz, this signal is a chip-by-chip time multiplex combination of two spreading codes known as Civil CM and Civil Long CL and carries CNAV data. The GPS L5 open signal with a carrier component

of 1176.45 MHz, deploys QPSK modulation. The in-phase component I5 is modulated with the I5 code and carries CNAV data, whereas the in-quadrature component Q5 is modulated with the Q5 code and carries no navigation data. L5 spreading codes have a faster chipping rate and longer sequence compared to L1 C/A codes. The faster chipping rate provides a sharper ACF peak, which results in better ranging precision and better multipath resolution.

The early version of GLONASS satellites deployed FDMA modulation, signals transmitted by a specific satellite with the same ranging code, but different carrier frequencies in the L1 and L2 sub-bands to differentiate the satellites. The GLONASS-M SV, launched in 2014, added an additional open CDMA signal on the L3 sub-band.

The Galileo E1 open signal is transmitted on L1 with a carrier component of 1575.42 MHz. The E1 OS deploys optimised primary (memory) codes, which consist of two components modulated with the CBOC(6,1,1/11) scheme. A data component (E1-B) is modulated by the I/NAV navigation message. The second component, E1-C, is data less (not carrying navigation data). The Galileo E5 open signal is transmitted with a carrier frequency centred at 1191.795 MHz. E5 consist of two sub-signals E5a and E5b, at 1176.45 and 1207.14 MHz respectively, multiplexed together through an AltBOC scheme. The E5b-I signal components are modulated using I/NAV navigation data. The Galileo E6 commercial signal is transmitted at a frequency centred at 1191.795 MHz. The signal consists of two main components. A BPSK(5) commercial signal (CS) multiplexed with a BOCc(10; 5) PRS signal. The commercial signal with a BPSK(5) occupies the in-phase component, and the PRS signal is in the quadrature-phase.

The BDS is the third independently developed GNSS after GPS and GLONASS. BDS provides open service to users globally, while providing enhanced services for the Asian-Pacific. The space segment is a hybrid constellation consisting of satellites in three types of orbits: GEO, IGSO, and MEO. The BDS B1 open signal is transmitted with a carrier frequency centred at 1561.098 MHz, the in-phase component B1I provides an open service, and B1Q provides an authorised service. China announced the B1C signal to make BeiDou interoperable with Galileo and GPS. This signal is transmitted from BDS-3 satellites with a frequency centred at 1575.42 MHz. The BDS B2 open signal is transmitted with a carrier frequency centred at 1561.098 MHz. This signal is similar to the Galileo E5 signal, which is composed of two sub-band signals referred to as B2a and B2b. Although GNSS technology is widely used, its further development is far from being over. Multiple GNSS constellations improve availability of signals, gives operators more access, and increases accuracy.

The benefits of satellite navigation are everywhere. For example, GNSS improves traffic flow and vehicle efficiency, guides users, and helps track assets and shipments. It can facilitate civil protection operations in harsh environments, speed up rescue operations and provide critical tools to coastguard and border control authorities. GNSS is also a key technology for time-stamping power sectors, telecommunications, financial transactions, for conducting scientific research and geodesy.

Chapter 4

GNSS user segment

4.1 Overview

The user segments of all satellite navigation systems are similar, consisting of a radio receiver with an operating frequency in the L-band range, capable of down-converting the Rf to IF signal, apply DSP, and provide a PVT solution to the user. As shown in Figure 4.1, a GNSS receiver consists mainly of four parts:

- Antenna;
- RF front-end;
- Analogue-to-digital converter (ADC);
- Signal processor unit.

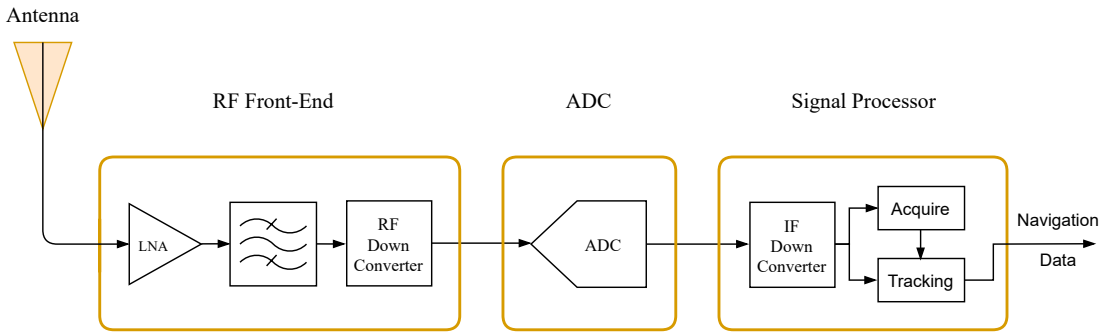


Figure 4.1: A typical GNSS receiver.

4.2 Antenna

The GNSS user antenna converts the received electromagnetic waves from satellites to an electric signal in the receiver. The most modern satellite navigation antennas are capable of detecting all in-view satellites from different constellations, that is, detecting all transmitted radio signals in the L-band from satellites above the horizon with 5° of elevation. There are some particular requirements that need to be taken into account when designing an antenna; these requirements might affect the antenna performance

according to the applications. A short overview of some of these requirements is provided in the following sections (Rao et al., 2013).

4.2.1 Centre frequency

The antenna centre frequency is the main operating frequency. Figure 2.3, illustrates the RF allocation for each navigation satellite system. It can be noted that some of the frequency bands for different navigation satellite systems are close or overlapping, which has advantages and disadvantages in terms of antenna design and performance. The advantage is that all the close and overlapped signals from different consolations can be received using one wideband antenna. The disadvantage of overlapping signals from different systems is the interference between the systems. For example, in the case of Galileo, there are three allocated centre frequencies (E1, E5, and E6). The E1 frequency at 1.57542 GHz is allocated for the open and public restricted service, E5 at the centre frequency of 1.191 GHz for open service, and finally E6 at a centre frequency of 1.278 GHz for commercial purposes. To utilise all the services provided by Galileo, there is a need for a Galileo multi-band antenna that can cover the E1, E5, and E6 frequencies and their band limit.

4.2.2 Bandwidth

The bandwidth is the frequency range around the centre frequency which can be covered by an antenna. This is an important requirement in antenna design to rebuild the received signals correctly. The bandwidth of a specific signal related to the modulation scheme and the chipping rate, for example, an antenna designed to receive GPS L1 C/A signal requires $\pm 1.023 MHz$ bandwidth, while the L5 signals require 10 times higher bandwidth, as the chipping rate of the L5 code is 10 times higher. In the case of Galileo E1 OS, the signal requires a 24.552 MHz bandwidth to rebuild the signal as BOC(6,1) and only 4.096 MHz to rebuild the BOC(1,1) component. An antenna that receives all GNSS services must be operational from 1166 to 1606 MHz, as shown in Figure 2.3, covering the entire lower L-band (1166–1300 MHz) and upper L-band (1559–1606 MHz). It is important to note that most GNSS antennas should maintain right-hand circular polarisation (RHCP) within a given bandwidth.

4.2.3 Radiation pattern

The radiation pattern of an antenna is a representation of radiation properties, such as the electric field or power, as a function of spatial coordinates. These patterns can be represented in two-dimensional and three-dimensional plots and are commonly presented as functions of the observation angles around the antenna.

4.2.4 Antenna gain

The gain G of an antenna indicates the efficiency of an antenna to receive power from or transmit in a specific direction in comparison to an idealised, lossless isotropic antenna (Balanis, 2016). It is defined as the product of the efficiency e of the energy conversion

inside the antenna and the directivity D

$$G = e \cdot D \tag{4.1}$$

$$D = \frac{4\pi \cdot U}{P_{rad}} \tag{4.2}$$

where U is the antenna's radiation intensity in a given direction, and $P_{rad}/4\pi$ is the corresponding value for the isotropic radiator. In general, the gain is expressed on a logarithmic scale as follows:

$$G_{dBi} = 10 \log\left(\frac{G}{10}\right) \tag{4.3}$$

where G_{dBi} denotes decibels relative to an isotropic antenna.
According to equation 4.1, a high-gain antenna will radiate most of its power in a particular direction, while a low-gain antenna will radiate over a wide angle. GNSS user receiving antennas are usually low-gain antennas having reception over a large range of angles, and high-gain antennas are used for transmitting signals on the GNSS satellites as they radiate most of the power in a particular direction.

4.2.5 Polarisation

The wave emitted from a satellite antenna is an electromagnetic wave. An electromagnetic wave is composed of two vector components: an electric field vector and a magmatic field vector. In general, the polarisation of an antenna is defined as the direction of its electric field vector. If the electric field vector is aligned with the horizon, the antenna is denoted as horizontally polarised, whereas when the electric field vector is perpendicularly aligned to the horizon, the antenna is denoted as vertically polarised.

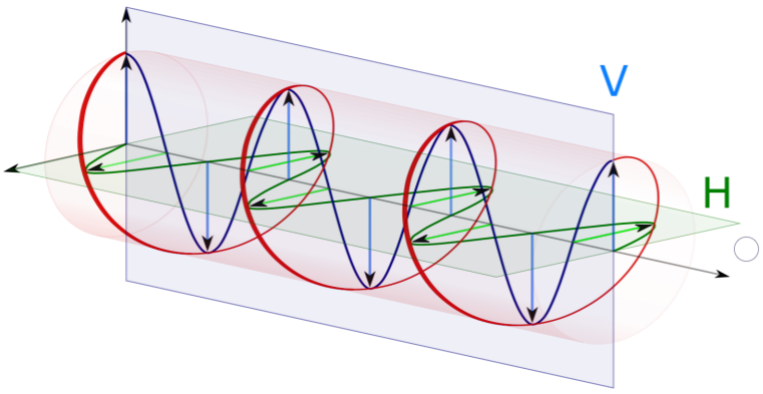


Figure 4.2: RHCP polarisation waveform.
(Robb, 2017)

The conventional GNSS transmitter/receiver antennas deploy RHCP polarisation, which means that the GNSS signal is composed of two orthogonal waves of equal amplitude with a 90° phase shift between the components as shown in Figure 4.2. The resultant electric field circulates in a clockwise direction. In the case of the signal reflection (due to the collision, for example), the polarisation may change; in other words, the signal will circulate in an anticlockwise direction, thus making the signal left-hand circularly polarised (LHCP) wave. Ideally, such signal components should be stamped out by a GNSS-receiving antenna to reject multipath signals.

4.2.6 Phase-centre stability

The antenna phase centre is a point within the antenna radiation pattern, where all the power emerges from (for a transmitter antenna) or converges to (for a receiver antenna). In fact, this point does not represent the centre of the physical shape of the antenna, but represents the electrical phase centre of the antenna and also depends on the signal frequency. The measured signals and the calculated PVT solution are all related to this position; thus, the antenna phase centre plays a key role in achieving millimetre resolution in positioning. The phase centre offset must be specified for every direction and frequency for accurate PVT estimation (Yegin, 2018).

The GNSS signal is below the thermal noise level when it arrives at the user receiver; for example, the GPS minimum received power is –158.5 dBW for L1 (Dunn and DISL, 2012). The cable and receiver generate thermal noise, which can affect the signal. To avoid signal weakness due to thermal noise, a low-noise amplifier (LNA) is provided directly after the antenna amplifies the signal above the thermal noise. GNSS antennas are of two types: passive and active. The passive antennas are not supplied with power because there is no LNA embedded in; therefore, they have to be attached (or very close) to the receiver to avoid cable losses. In a passive GNSS antenna, the LNA is built into the RF unit of the receiver. An example of this type of antenna is the passive antenna used in mobile devices. In an active antenna, an LNA is embedded in the antenna. The advantage is that the cable loss can be compensated by applying an appropriate gain at the LNA output. The active antenna is supplied with power from the receiver through the antenna cable.

4.3 RF front-end

The GNSS signal at the antenna output is at a very high frequency, between 1.1 to 1.6 GHz. Processing this signal is very complicated and can be very expensive; to avoid the complexity and cost of processing, in some technologies, the RF signal is down-converted to an IF in the front-end. In the GNSS receiver, the RF front end is the most critical component. The size, cost, and power consumption of the receiver is determined by this component. The bandwidth of the filter in the front end determines the pseudorange accuracy. The noise figure is another critical point affecting the performance of the RF front-end, where the noise figure is the thermal noise added to the main received signal while passing through the analog chain such as cables and the receiver components. The main function of the front end is to down-convert the RF signal to IF. Figure 4.3, illustrates the block diagram of an RF front-end, which consists of LNA, RF and IF filters, a mixer, and a local oscillator (LO) used for down-conversion.

4.3.1 Low noise amplifier

The aim of LNA is to amplify the signal extremely (usually above 20 dB) the signal before it has been degraded because of the noise figure effect. The noise figure is a fundamental parameter of the receiver, and for this reason, the LNA is directly located next to the antenna output. In some applications where the antenna is far away from the receiver and they have to be linked through a cable, the LNA must be embedded in the antenna to avoid the losses in the cable before arriving at the receiver, that is, an active antenna should be used. The first component following the antenna can be either a filter or an

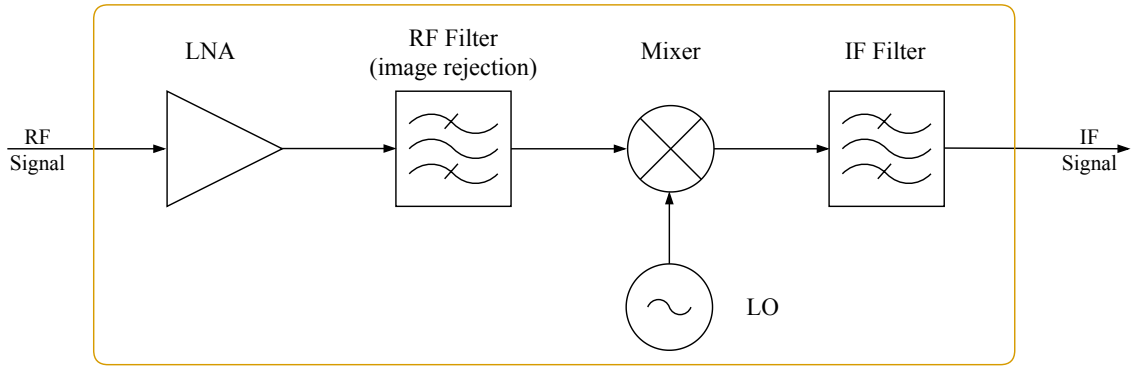


Figure 4.3: Block diagram of a RF Front-End.

amplifier. If the amplifier is embedded in the antenna, the first component after the antenna is the amplifier. Both arrangements have advantages and disadvantages, and the noise figure F of a receiver can be expressed as (Morton et al., 2003)

$$F = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 G_2} + \cdots + \frac{F_N - 1}{G_1 G_2 \cdots G_{N1}} \tag{4.4}$$

where F_i and $G_i (i = 1, 2, ...N)$ are the noise figure and gain of each individual component in the RF chain, respectively. If the amplifier is the first component, the noise figure of the receiver is low at this stage and is approximately equal to only the noise figure of the first amplifier, which can be less than 2 dB (Tsui, 2005). The overall noise figure of the receiver caused by the second component, such as the filter, is compensated by the gain of the amplifier. The disadvantage of this approach is that strong signals in the bandwidth of the amplifier may drive it into saturation and generate spurious frequencies. If the first component is a filter, it can stop out-of-band signals from entering the input of the amplifier, and the insertion loss of such a filter is usually high. The receiver noise figure with the filter as the first component is approximately 2–3 dB higher than that of the previous arrangement. Therefore, the GNSS receivers use the first approach to compensate for the noise figure from the amplification power.

In fact, commercial LNAs will not only amplify the signal but also add noise to the amplified signal. Manufacturers are trying to build an efficient component that amplifies the signal and adds minimal noise. The main parameters used to define an LNA are as follows:

- Gain, expressed in dB, is assumed to be the maximum at a frequency and slightly lower for other frequencies.
- Noise figure, expressed in dB, the amount of noise that will be added to the signal after amplification.
- Operating frequency range.

4.3.2 RF and IF filters

Next to the LNA, a bandpass filter is placed to select the desired RF band that needs to be covered. This filter passes only the frequencies of interest and rejects (attenuates massively) the image frequency. In general, most commercial GNSS receivers deploy surface acoustic wave (SAW) type filters for this stage of filtering. The aim of placing an IF filter

after the mixer is to reject the undesired frequencies generated by the intermodulation products of the mixer and to reduce the bandwidth to the band of interest (this depends on the ranging code chipping rate and the modulation scheme), reducing the noise.

4.3.3 Mixer/Local oscillator

The mixer mixes the incoming RF signal with the signal generated from the local oscillator to down-convert the RF to IF signal after filtering out the out-of-band frequencies from the intermodulation product. Mixing an RF signal with a centre frequency of f_{RF}, with a local oscillator tone of frequency f_{LO}, produces intermodulation products. One decision to be made when designing a superheterodyne receiver is whether the frequency of the local oscillator is below the frequency of the input carrier (low-side tuning) or above the frequency of the input carrier (high-side tuning) (Ziemer and Tranter, 2014). As illustrated in Figure 4.4, in case of $f_{LO} < f_{RF}$, the first f_{IF} component of the intermodulation product translates the information from the RF band to the IF (difference between two frequencies), which is the desired down-converted signal.

$$f_{IF} = f_{RF} - f_{LO} \tag{4.5}$$

The second component of the intermodulation product f_h is the sum of two frequencies

$$f_h = f_{RF} + f_{LO} \tag{4.6}$$

The sum frequency at the output of the mixer is rejected by the second IF filter placed after the mixer. A low-pass filter should be more than sufficient, and it is used to select only the desired difference frequency and reject the sum frequency. In reality, a wide input RF band might contain an image frequency f_{im}.

$$f_{im} = f_{RF} - 2f_{IF} \tag{4.7}$$

The LO will also produce an intermodulation product with the image frequency at the output of the mixer, generating the sum and difference components, distorting the desired IF component, owing to overlapping at the same frequency or around frequencies when the image frequency is not exactly multiple of the IF frequency. To reduce the distortion produced in the desired IF signal, the image frequency must be attenuated before the mixing process by filtering the incoming RF signal. In Equation 4.7, it can be noted that if f_{IF} chosen very low, the image frequency is near the centre frequency of the RF signal, requiring an extremely critical filter to attenuate the signal sufficiently. A sufficient filter at these frequencies can be costly, implying large insertion losses. On the other hand, choosing an IF frequency is related to the bandwidth of the ADC component used to digitise the IF signal. The bandwidth of the ADC B_{AD} should verify the following conditions (La Valle et al., 2011):

$$B_{AD} = f_{IF} + \frac{B_{RF}}{2} \tag{4.8}$$

where B_{RF} is the bandwidth of the signal of interest.
In equation 4.8, the higher the IF frequency, the wider the ADC bandwidth. Therefore, there is a trade-off between the attenuation of the image frequency and the cost of ADC.

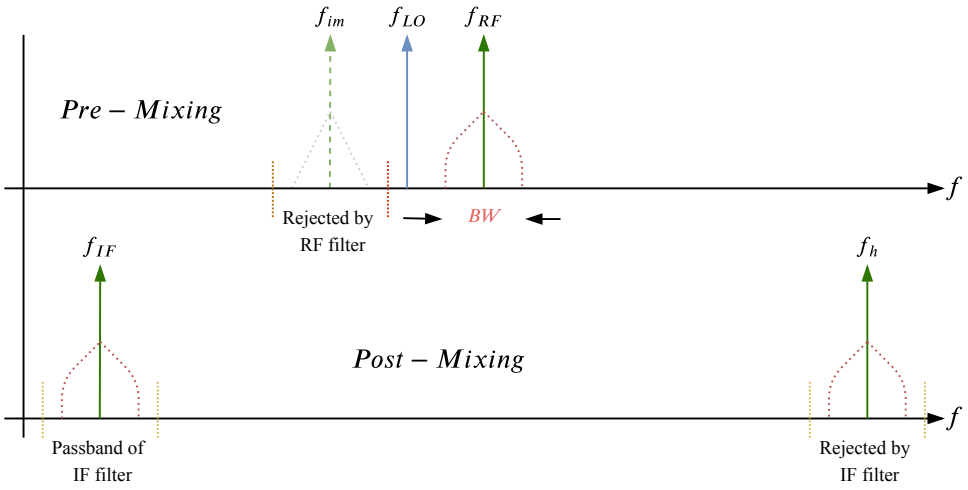


Figure 4.4: Intermodulation product of mixing two signals (low-side tuning).

4.4 Down-conversion methods of RF signal

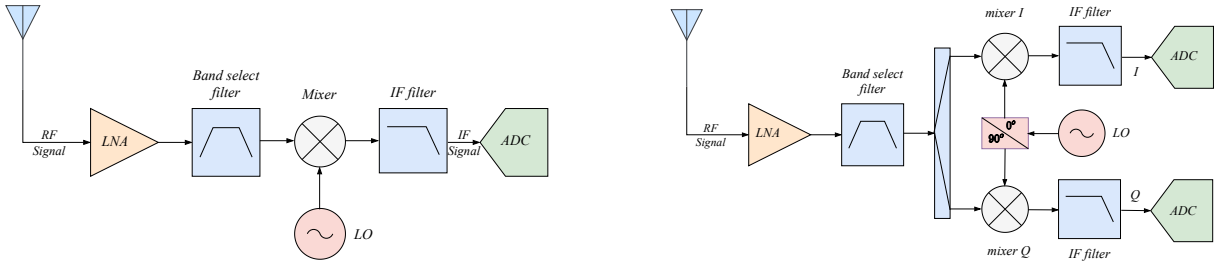
There are two main approaches to process the RF frequency in the front-end: homodyne and superheterodyne down-conversions. In the homodyne approach, the Rf frequency is directly processed without down-conversion to a lower frequency; therefore, it is also known as a direct conversion receiver (DCR) or zero IF receiver. In the superheterodyne approach, the RF frequency is converted to an IF before it has been converted from analog to digital. In general, the frequency of a signal $x(t)$ can be shifted by multiplying the signal with another sinusoidal signal $s(t)$.
If $x(t) = A \cdot \cos(\omega_{Rf} \cdot t)$ and $s(t) = \cos(\omega_{LO} \cdot t)$ then:

$$x(t) \cdot s(t) = \frac{A}{2}\cos(\omega_{Rf} - \omega_{LO})t + \frac{A}{2}\cos(\omega_{Rf} + \omega_{LO})t \tag{4.9}$$

The multiplication product is the sum and the difference between the two frequencies which produces high-and low-frequency components. Rejecting the high component using a low-pass filter, the remaining component is only a low component. The multiplication in the down converters is performed using a mixer, the sinusoid signal is generated from a tunable LO, and a low-pass filter is used to remove the high component of the mixing process. These three components are key components of a basic down-converter.

4.4.1 Homodyne down-conversion

Homodyne down-conversion is a special case of single-stage frequency down-conversion. In this scheme, the LO produces a synchronised signal (at the same frequency) as the received RF signal to down-convert the signal to the direct current (DC) signal by coinciding with both centre frequencies. This method is a simple conversion scheme because it converts the signal down to the DC (baseband). A simple homodyne down-converter is shown in Figure 4.5a, where the LO signal frequency is equal to the input RF frequency. the desired signal selection requires only a low-pass filter with sharp cut-off characteristics. While the GNSS signals deploy frequency and phase modulation techniques, the down-conversion should provide quadrature outputs, as shown in Figure 4.5b, to avoid loss of information (Razavi and Behzad, 1998). The advantages of this technique are:



(a) Simple homodyne downconversion.

(b) IQ homodyne downconversion.

Figure 4.5: Block diagram of homodyne downconversion.

- Reduced system complexity due using only single mixer and single filter at low frequency.
- No need to image rejection filter before mixing process.
- High selectivity (extracting the desired signal satisfactorily in presence of interference).

The disadvantages of this technique are:

- DC offset in signal due to self-mixing. If the LO leaks into the RF path, it will self-mix, produce a DC offset, and cause signal degradation (Razavi and Behzad, 1998).
- There is a need for high-precision LO to synchronise with the incoming signal. complicated feedback loops may be required for synchronisation.

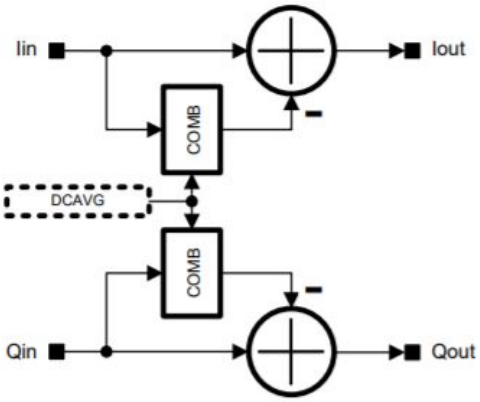


Figure 4.6: DC offset correction module as implemented in LMS7002M transceiver.

The DC offset in signal due to self-mixing can be cancelled by a compensation loop running in real time to track and correct the DC at the output. A simple digital implementation of such a loop is given in Figure 4.6. The averaging (COMB) filter calculates the DC of the corrector input and subtracts it to cancel out the offset. The loop is running all the time so any change of the receiver DC due to the signal level change, the gain change or temperature will be tracked and cancelled automatically. This technieque is implemented in LMS7002M transceiver from "Lime microsystem".

The homodyne down-conversion is an efficient approach for software-based RF, where the down-converted signal to the base band is sampled by ADC and mixing, and filtering is performed digitally.

4.4.2 Heterodyne down-conversion

The heterodyne down-conversion (Douglas, 1990) is the most commonly used downconversion method in commercial receivers because it provides better selectivity and improved SNR by reducing the signal bandwidth. In this scheme, the RF signal is down-converted to a lower IF signal that is not direct to a base-band signal. Heterodyne down-conversion may occur at multiple stages. Figure 4.7, Illustrates a dual stage heterodyne block di-

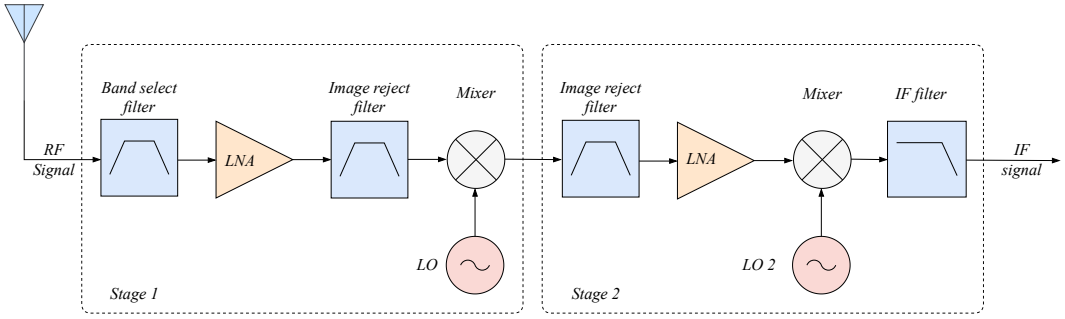


Figure 4.7: Block diagram of a generic dual stage heterodyne.

agram. In the first stage the bandwidth of interest is selected by a band select filter, amplified and mixed with a sinusoid signal from LO. The LO frequency $f_{LO} = f_{RF} + f_{IF}$ (high-side tuning). Figure 4.8, shows the spectra illustrating of the mixing process, the Lo signal down-converts the desired Rf signal to an IF signal, meanwhile it down-converts the image signal at f_{im} which is away from the RF signal by $2f_{IF}$. The product of the heterodyne will be an IF signal composed of two components, the desired and the image signal. The image signal overlapped with the desired signal, this phenomenon distorts the desired signal, to prevent this issue the desired signal is band passed through an image rejection filter to suppress the image signal, before the mixing process. Figure 4.9, shows the filtering process of the desired signal to suppress the image signals before mixing with LO sinusoid signal. In fact, it is very important to minimise the image rejection ratio (IRR) at the output of the IF port. IRR is defined as the desired signal to the image signal power:

$$IRR = \frac{Power\ of\ the\ desiered\ signal}{Power\ of\ the\ image\ signal}$$

There is a trade-off between trying to minimise the image and trying to filter out the desired signal, that is, between sensitivity and selectivity. In the case of selecting a very low IF as shown in Figure 4.10a, the interference of the image signal will not be attenuated efficiently by the image rejection filter because it will be partially within the bandwidth of the filter, which overlaps with the desired signal after down-conversion and distorts the desired signal, thereby reducing the receiver sensitivity. In the case of selecting a high IF, as shown in Figure 4.10b, the interference of the image signal will be rejected efficiently by the image rejection filter because it will be outside the bandwidth of the filter, improving the sensitivity of the receiver. This relaxes the requirements of the RF filter. However, a higher IF requires an IF filter with a higher Q-factor to suppress the adjacent channel

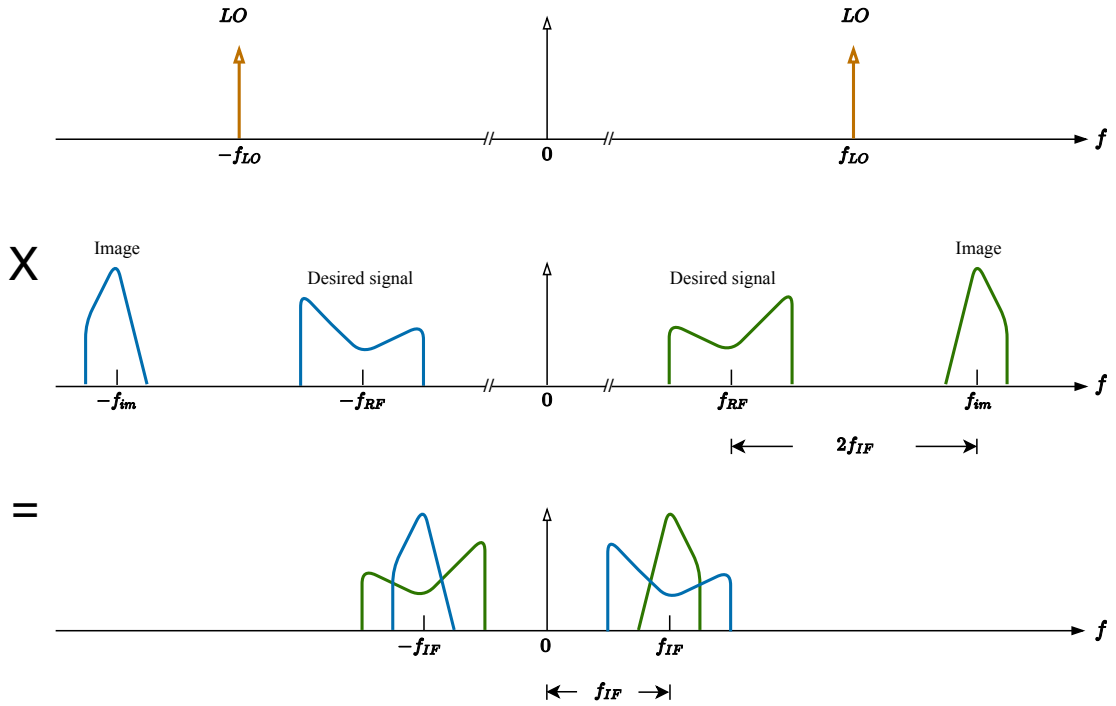


Figure 4.8: Spectra illustrating heterodyne down-conversion on the bottom shows overlapping image signals with the desired signal after mixing.

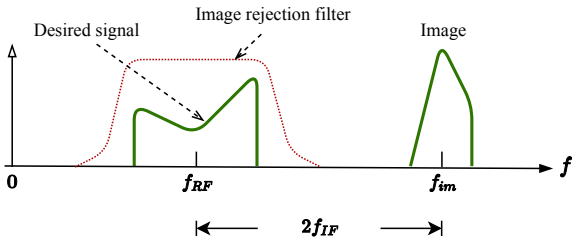


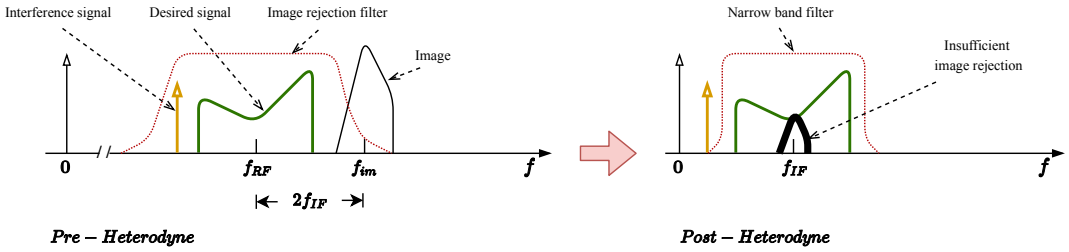
Figure 4.9: Spectra illustrating of rejecting the image signal by band-pass filtering the desired signal before mixing process.

interferers. To resolve this issue, the concept of heterodyning can be extended to two-step or multiple downconversions, each followed by filtering and amplification. As illustrated in Figure 4.7, this technique performs partial channel selection at progressively lower centre frequencies, thereby relaxing the Q required of each filter. The lower-frequency operations in the second stage of mixing and filtering can be performed more efficiently in the digital domain using quadrature (I and Q) down-conversion to avoid information loss due to the phase modulation technique in GNSS signals and I and Q mismatching problems which are common in the analog domain. As shown in Figure 4.11, the IF from the first stage is digitised, mixed with a digital I and Q sinusoid, and low-pass filtered to obtain I and Q base-band signals. This approach is sometimes denoted as "digital-IF architecture" (Razavi and Behzad, 1998).

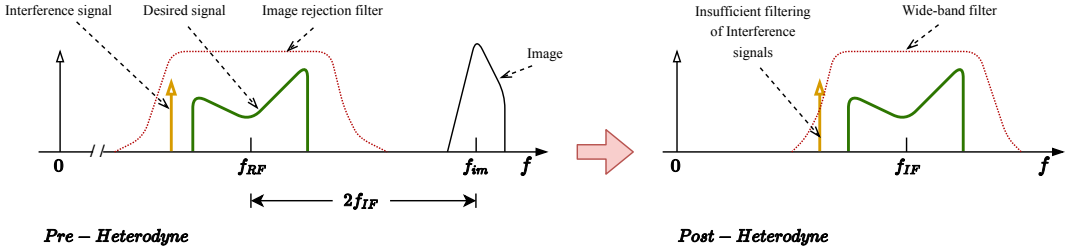
Following are the disadvantages of heterodyne architecture:

- This method requires additional LOs and RF Mixers to convert signal from RF to IF before conversion to baseband. This increases cost of overall receiver.
- Moreover, filters are also needed to remove undesired frequency components to pre-

vent image frequencies. This also increases cost as well as complexity of the receiver.



(a) Low intermediate frequency selection.



(b) High intermediate frequency selection.

Figure 4.10: The trade off between the choices of intermediate frequency.

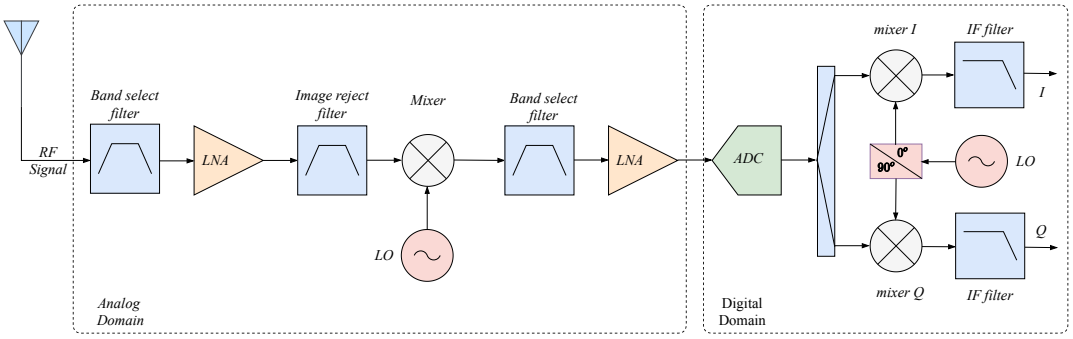


Figure 4.11: Digital-IF architecture.

4.4.3 Subsampling down-conversion

An RF signal with bandwidth W and centre frequency f_{RF} can be sampled with f_s at much lower rate of f_{RF}. As illustrated in Figure 4.12, sampling with a frequency $f_{RF} > f_s > 2W$ creates copies of the spectrum with no aliasing in the bandpass. This operation called "subsampling" causing a large reduction in RF centre frequency. This operation can be used as a down-conversion technique to convert RF to IF. The subsampling technique can simplify the receiver design by dismissing most of the components in the analog chain within the front-end, such as the mixer and LO, and using a digital down converter (DDC) instead of down-converting the IF to a baseband signal. Despite these benefits, subsampling operation sufferers from jitter and folded thermal noise (Meng et al., 2016) (Oya et al., 2012). Figure 4.13, shows the thermal noise folded back in the band of interest in the subsampling process.

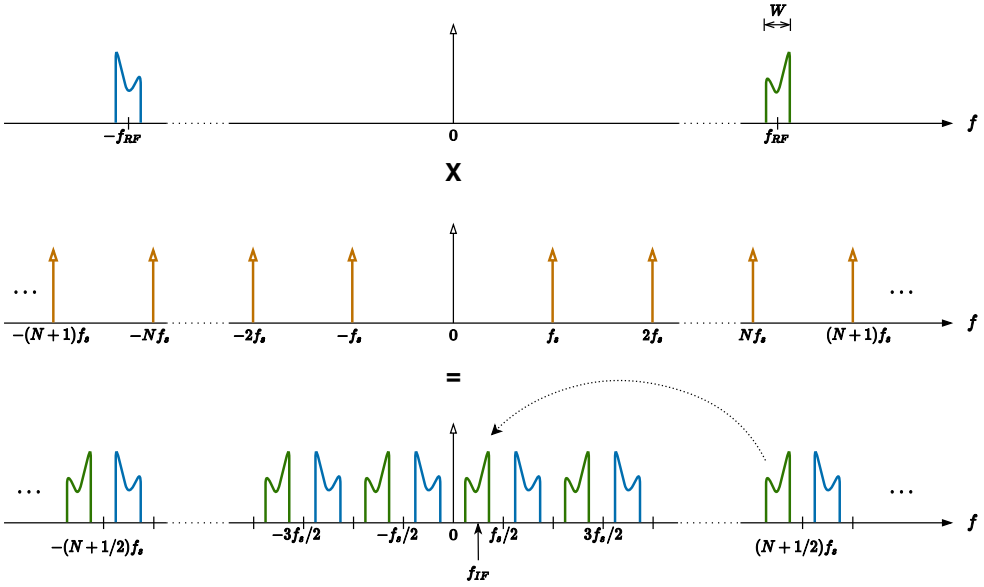


Figure 4.12: Illustration of subsampling concept: (top) frequency domain representation of RF bandpass signal, (middle) frequency domain representation of sampling signal and (bottom) signal replicas following subsampling process.

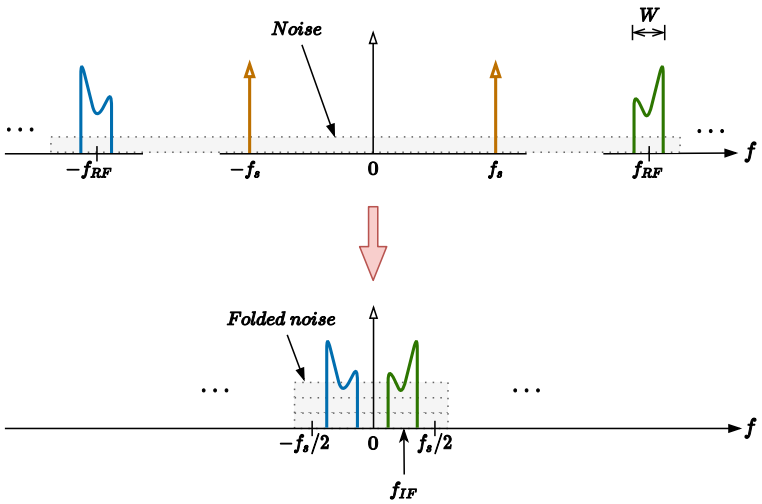


Figure 4.13: Thermal noise folded in the band of interest in subsampling conversion.

4.5 Analog to digital conversion

The ADC is the last part of the front-end. The ADC converts the continuous time to discrete time by sampling the signal in equally spaced time slots and converts the continuous amplitude to discrete amplitude by quantizing the continuous amplitude to the corresponding binary value in each time slot. As shown in Figure 4.14, the key component of the ADC is a local oscillator; to set the sampling frequency, the sample and hold circuit samples the voltage of the continuous analog signal and holds this value for a sampling period of time, a threshold comparator to compare the analog amplitude with a number of predefined thresholds, and an encoder to map the result of the comparison

to a certain number in binary representation. Choosing a proper sampling rate for a

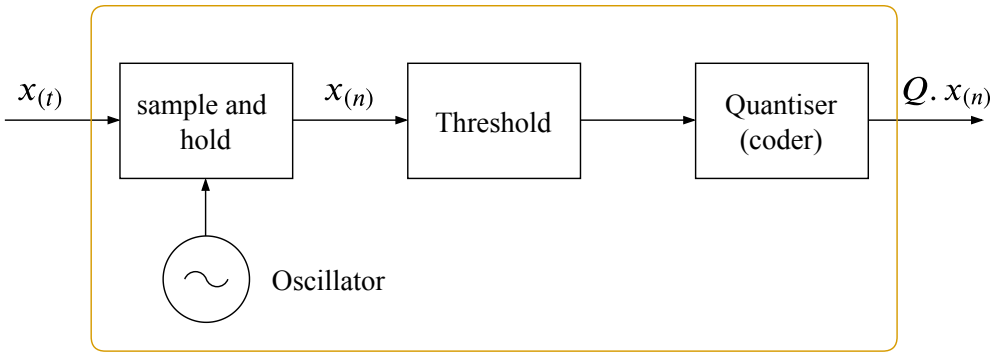


Figure 4.14: Schematic diagram of an analog-to-digital converter.

GNSS receiver is very important. The Nyquist frequency of the sampling theorem (Eldar, 2015) is maintained. To understand this more properly, assuming a signal $x(t)$ with a bandwidth of WHz sampled at a sampling rate f_s, where $f_s > 2W$. The signal $x(t)$ can be reconstructed from the sampled signal $x_\delta(t)$ by passing it through an ideal low-pass filter with bandwidth B, where $W < B < (f_s - W)$. The frequency $2W$ is referred to as the Nyquist frequency (Ziemer and Tranter, 2014).

Figure 4.15 (a), illustrates the spectrum of the signal $x(t)$ while Figure 4.15 (b), illustrates the spectrum of the sampled signal $x_\delta(t)$ reconstructed with an ideal low-pass filter. In this figure it can be noted that in case of $f_s > 2W$ i.e. when the sampling theory is satisfied, the signal can recovere without distortions from the sampled version by using a proper low-pass filter. If the sampling theorem is violated, either due to

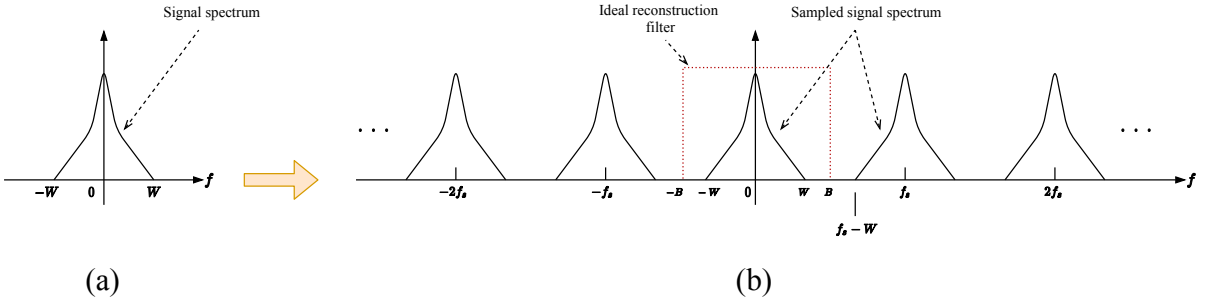


Figure 4.15: Signal spectra. (a) Assumed spectrum for $x(t)$. (b) Spectrum of the sampled signal.

not band-limiting the signal or because of $f_s < 2W$, the reconstructed signal will be distorted due to overlapping of the sampled signals on the output of the filter, which is denoted as aliasing, as illustrated in Figure 4.16 (a), which can be avoided either by increasing the sampling frequency or by filtering the signal before sampling. Distortion might occur during the filtering process dbecause of the non-ideal frequency response of the real filters, as illustrated in Figure 4.16 (b), which can be attenuated by deploying reconstruction filters with sharper frequency response characteristics or by increasing the sampling frequency. It has been observed through experiments that user position accuracy decreases if the sampling frequency is an integer multiple of the nominal code rate (Tran et al., 2018). When the Nyquist rate is significantly higher than the normal rate (oversampling), a larger band relative to the GNSS signal band is sampled (Teunissen and

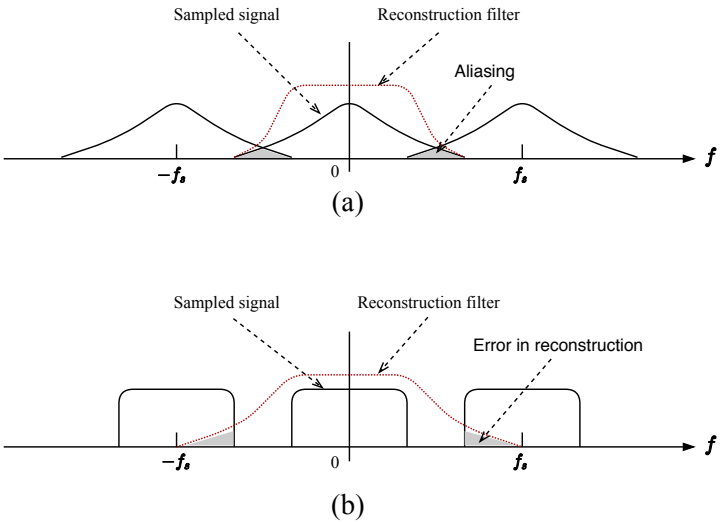


Figure 4.16: Spectra illustrating two types of errors the in reconstruction of the sampled signals. (a) Aliasing error in the reconstruction of sampled signals. (b) Error due to the non-ideal reconstruction filter.

Montenbruck, 2017). This leads to a situation in which the noise bandwidth and potential interference outside the useful band are sampled. The CDMA nature of the GNSS signal

Table 4.1: Signal degradation due to finite-bit quantization in the A/D converter.

$IF - Bandwidth$	$1 - bit ADC$	$3 - bit ADC$
$Narrow\ (1/T_C\ Hz)$	$3.5dB$	$0.7dB$
$Wide\ (5/T_C\ Hz)$	$2.25dB$	$0.3dB$

requires a very small dynamic range from the sampled signal (Borre et al., 2007). Simple GNSS receivers deploy one-bit quantisation, while better-quality receivers deploy three-bit quantisation. The degradation of the signal due to quantisation is dependent on two factors, in addition to the number of quantisation levels. The first is the IF bandwidth. In other words, the bandwidth of the final stage of the front end. The second is the ratio of the maximum A/D threshold to the RMS noise level. The signal degradation due to quantisation in the ADC for 1-bit and 3-bit is given in Table 4.1 (Braasch and Van Dierendonck, 1999).

4.5.1 GNSS fornt-end examples

Figure 4.17, shows the block diagram of a four-channel RF front-end IC for the reception of GNSS signals from NTLab. This FE capable of receiving signals from GPS, GLONASS, Galileo, BeiDou at all frequency bands in various combinations: L1, L2, L3, L5, E1, E5a, E5b, E6, B1, B2, B3. Galileo E5 band as well as BeiDou B1, B2, B3 (phase 3) band can be obtained as entire signal with two channels fed by the same LO and then restored in digital domain to true complex data. Each setting, including output signal

frequency bandwidth, AGC options, mirror channel suppression option, etc., can be set for every channel individually. NT1065 includes two fully independent frequency synthesizers.

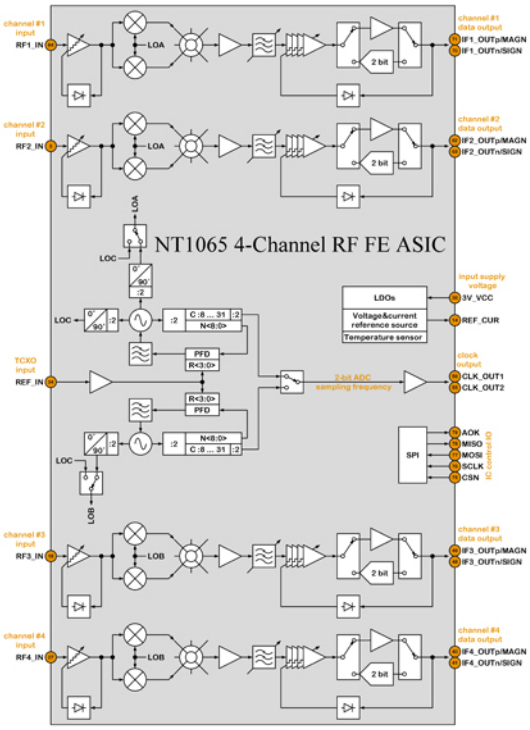


Figure 4.17: NT1065 Block diagram (from NTLab).

Channel 1 and channel 2 are supplied with LO signal generated in PLL "A" while PLL "B" is assigned for channels 3 and 4. For specific applications there is an option to feed all four channels with single LO source from PLL "A". All the functionality allows application of NT1065 in high precision GNSS based positioning, goniometric, driverless car systems and related branches.

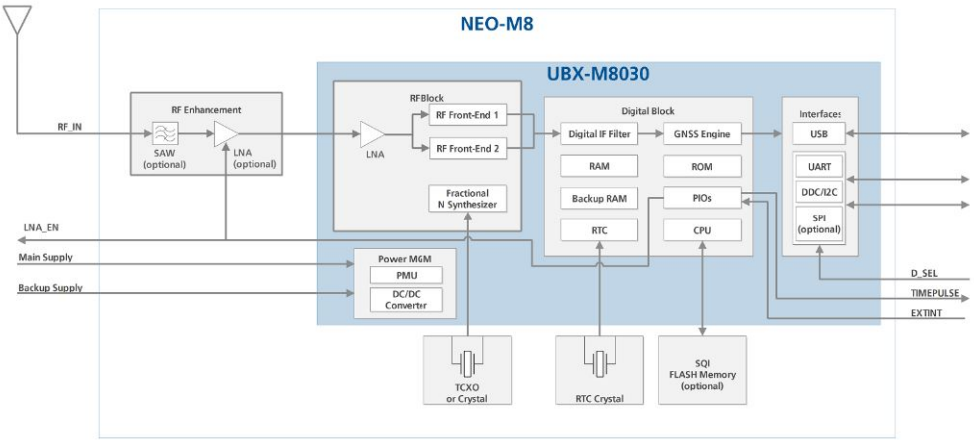


Figure 4.18: NEO-M8 Block diagram (from ublox).

NEOM8 is another example from "ublox" as shown in Figure 4.18. NEO-M8 is a concurrent GNSS receiver which can receive and track multiple GNSS systems: GPS, Galileo, GLONASS and BeiDou. Owing to the dual-frequency RF FE architecture, either

GLONASS or BeiDou can be processed concurrently with GPS and Galileo signals providing reception of three GNSS systems. The receiver can be configured to receive any single GNSS constellation or within the set of permissible combinations. By default the M8 receiver are configured for concurrent GPS and GLONASS. If power consumption is a key factor, the receiver should be configured for a single GNSS operation using GPS, Galileo, GLONASS or BeiDou.

4.6 Summary

The RF front-end is the most critical component of the GNSS receiver. The size and power consumption of the receiver is determined by this component. The accuracy of the receiver is determined by the bandwidth of the filter and the noise figure in the front end. The main function of the front-end is to down-convert the RF signal to IF. LNA is the first component in the front-end, and is located next to the antenna output to amplify the signal (usually above 20 dB) before it has been degraded owing to the noise figure effect. The second component is the RF bandpass filter, to select the desired RF band of interest and reject the image frequency. The incoming RF signal is downconverted to IF using the superheterodyne process. In this process, the RF is mixed with a locally generated signal in the local oscillator to downconvert it to a desired IF using dedicated mixers for this purpose. To reject the undesired frequencies generated by the intermodulation products by the mixer and to reduce the bandwidth to the band of interest, an IF filter is placed after the mixer.

The ADC is the last part of the front end. It converts the continuous amplitude to discrete amplitude. It is important to maintain the Nyquist frequency of the sampling theorem during the ADC process. When the sampling theory is satisfied, the signal can recover without distortions from the sampled version by using a proper low-pass filter. Different topologies are used to downconvert GNSS signals. In the homodyne (direct) down-conversion, the received RF signal is down-converted to a DC signal by generating a local copy at the same frequency as the received RF signal. This method reduces the complexity by using only a single mixer and a single filter at the IF side, and does not require an image rejection filter before the mixing process. However, the drawback of this method is the DC offset in the signal due to self-mixing, and the system requires a high-precision LO.

Heterodyne down-conversion is another method that is most commonly used in GNSS receivers because it provides better selectivity and improved SNR by reducing the signal bandwidth. In this method, the RF signal is down-converted to a lower IF signal that is not direct to a base-band signal, and down-conversion might be in multiple stages. In this method, to suppress the image signal, the desired signal is passed through an image rejection filter before the mixing process. There is a trade-off between trying to minimise the image and trying to filter out the desired signal, that is, between sensitivity and selectivity. To improve the performance, a two-step down-conversion is proposed, which is also known as dual-IF heterodyne. In the first stage, a high IF is performed to remove the image signal. In the second stage, a low IF is performed to remove the in-band interference. Subsampling down-conversion is another method which takes advantage of the aliasing phenomena. This method can simplify the receiver design by dismissing most of the components in the analog chain within the front-end, such as the mixer and LO. The drawback of this method is the jitter and folded thermal noise.

Chapter 5

GNSS signal acquisition

5.1 Acquisition

Before a GNSS receiver can make any measurements or determine position, it must acquire the satellite signals. The aim of acquisition is to find visible satellites in users scope and coarse values of carrier frequency and code phase of the satellite signals. It is important to point out that before acquisition process the incoming signal is downconverted to the baseband, however, this signal is not pure DC, but it is modulated with the Doppler shift frequency, therefore, it is important to estimate the the Doppler shift effect to be able to generate a local carrier signal to remove the incoming Doppler shift frequency from the signal. The code phase, is the chip offset of the received PRN code sequence. It is necessary to know the code phase in order to generate a local PRN code that is perfectly aligned with the incoming code. Only when this is the case, the incoming code can be removed from the signal. Once the signal is detected, the carrier frequency and code phase passed to a tracking unit to track the signal by compensating the Doppler shift frequency and strip-off the PRN code continuously to obtain the navigation data.

5.2 Effect of Doppler

As the Satellite Vehicle (SV) moves with respect to the receiver, the carrier frequency will change due to the Doppler effect. This leads to an uncertainty as to where exactly in frequency the carrier is during the signal acquisition processing by the receiver. The main factors affecting the Doppler frequency are, satellite motion in the sky with respect to the receiver, user (receiver) motion with respect to the satellite and used reference clock resolution in the receiver.

5.2.1 Doppler effect due to the satellite movement

A GPS satellite circles with a speed $V_s \approx 3874$ m/sec around the earth in Medium Earth Orbit (MEO) with a radius of $26560KM$, induce appreciable Doppler on the carrier wave. The Doppler frequency f_D can be calculated by using a vector approach and equations as shown in Figure 5.1 (Doberstein, 2011). In this figure the radial velocity component of the SV is broken into two parts, one perpendicular to the line of sight (LOS) and the other parallel to the LOS vector. The component parallel to the LOS, V_d is the velocity

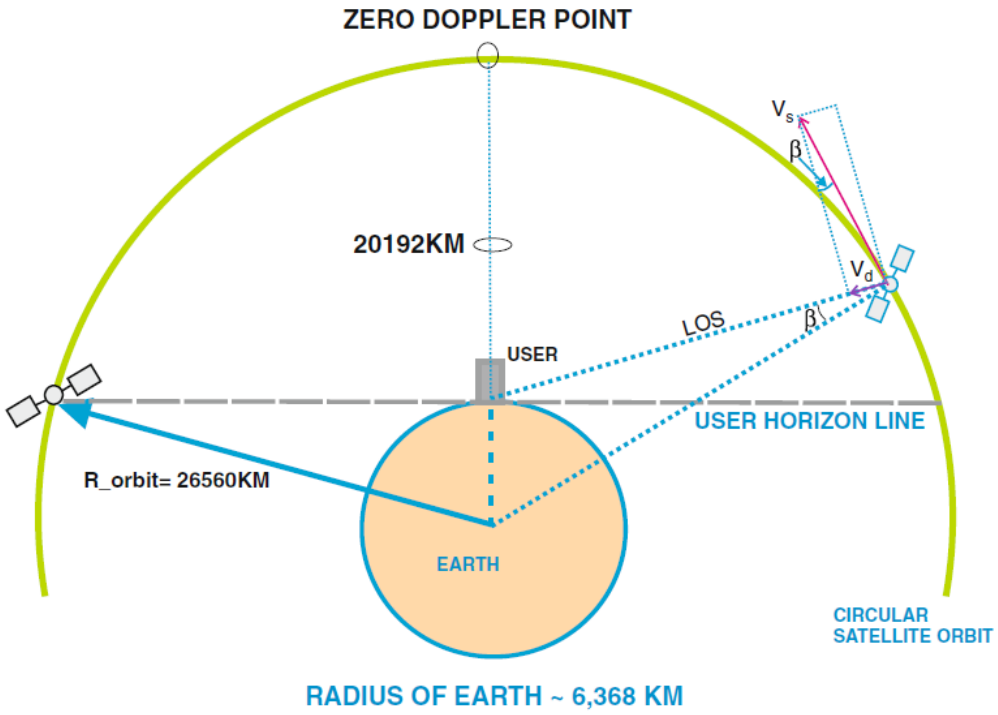


Figure 5.1: Doppler calculations for a GPS SV, L1 carrier (Doberstein, 2011).

component that causes the Doppler effect.

$$V_d = V_s . \sin(\beta) \tag{5.1}$$

$$f_D = \frac{f_{L1} . V_d}{C} \tag{5.2}$$

where:
$C = 299792458 M/S$ (speed of light)
$f_{L1} = 1575.42 MHz$ (L1 carrier frequency)
As the SV passes through its highest point in the sky the angle β will be zero, which results in zero Doppler shift. It will also undergo a sign reversal, that is from a positive to a negative Doppler. Maximum Doppler is encountered when the SV is low in the sky, i.e., low elevation angles. The SV radial velocity is not constant in magnitude or direction. This is due to the very slight elliptical orbit and small forces that perturb the orbit. The range of Doppler induced on the L1 carrier is approximately $\pm$ $5kHz$.

$$f_D(max/min) \approx +/- 5kHz \tag{5.3}$$

This frequency offset appears not only on the received carrier at $1575.42 MHz$ but also appears on all the modulated signals with the carrier, C/A code and navigation data. The exact amount of shift at a particular component modulated with the carrier will depend on the multiplying ratio between the carrier frequency and that component frequency. For example, the C/A code clock is at a frequency of $1.023 MHz$. The ratio to the carrier is exactly 1540. Therefore, the incoming C/A code clock (as seen by the receiver) could have a Doppler- shift of approximately $\pm 3.2 Hz$.

5.2.2 Doppler effect due to the receiver movement

The receiver (user) velocity changes the received SV carrier frequency by an extra number of wavelengths which drives through each second. If the receiver moving directly towards the SV, the received frequency will increase by the user speed in wavelengths (λ)/Second and vice versa.

$$f_{Dr} = \frac{V_{Dr}}{\lambda} \tag{5.4}$$

where:
f_{Dr} = doppler shift due to receiver movement
V_{Dr} = receiver velocity component that causes the Doppler effect.
λ = carrier wavelength ($0.19m$ for L1 carrier)
For example if a car drives directly towards a SV with a speed of $100Km/h$, the received frequency will increase by $100Km/h = 27.8m/s$.

$$f_{Dr} = \frac{27.8m/s}{0.19m}$$

$$f_{Dr} = 146Hz$$

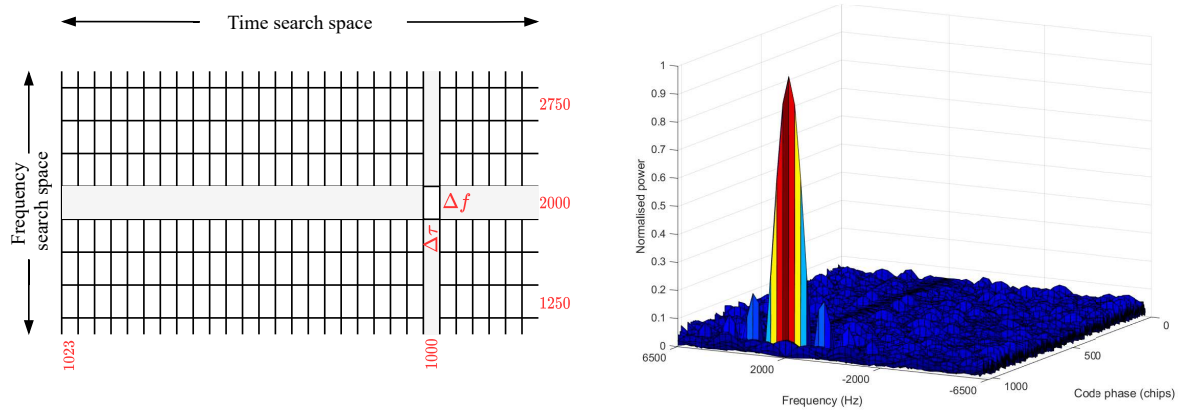
5.2.3 Offset due to the receiver clock error

For every one part per million (ppm) clock offset, there will be a $\pm 1Hz$ error. For example, if signal 1575.42 MHz generated using an oscillator with $\pm 1ppm$, the signal frequency will be offset by $\pm 1.57542KHz$.

5.3 Frequency and code-delay search

Each GNSS satellite transmits the navigation data at a fixed frequency in the allocated L band, however, the signals are not received at the same frequency on the user end because of the Doppler shift caused by the satellite motion and the receiver motion and any frequency offset in the receiver clock as discussed in the previous section. Even if the receiver has the correct frequency, it must still find the correct code delay for the correlators to obtain a correlation peak. Therefore, to obtain a coarse alignment with the received incoming signal a set of local replicas taking all possible values for the Doppler shift and code phase is used to generate correlator peak. This gives the receiver a two-dimensional search space for each satellite, denoted as the frequency and code delay search space or acquisition grid.

To make this more clear a quantitative example has been taken. Assuming a receiver searching for GPS C/A signal with coherent integration time $T_C = 1ms$ at the static mode using an on-board oscillator with $\pm 1ppm$. The receiver will have to search 10 kHz of unknown frequencies caused by the Doppler effect of satellite motion. There will be no Doppler effect caused by the receiver in the static mode. There will be an additional 3 kHz of unknown frequency offset for 1 ppm oscillator. Therefore, the total unknown frequency range will be 13 kHz. If the receiver search these frequencies in bins of about 250 Hz each, this means 52 bins to search. On the other hand, the receiver without any priori knowledge of code delay will also have to search all possible code-delay. The total



(a) Two dimensional acquisition search grid. (b) Three dimensional Acquisition matrix.

Figure 5.2: Frequency and Code-Delay Search Space for GPS C/A SV1. The search space is 1023 chip by 52 frequency bins each of 250 Hz. The maximum peak is at 1000 chips and 2000 Hz.

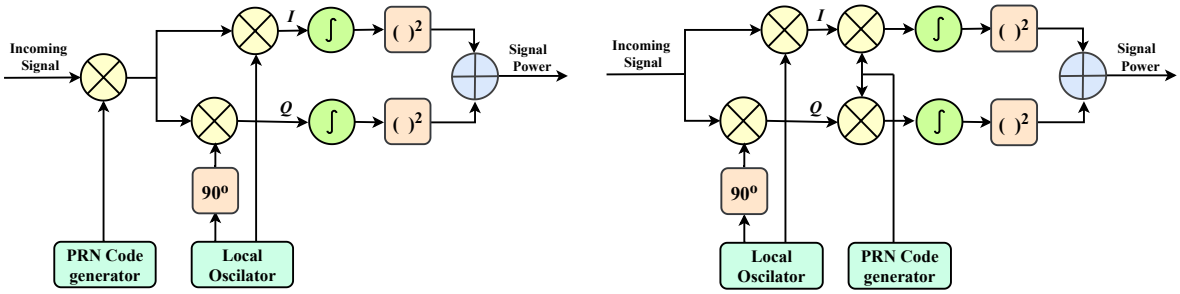
code-delay search space is 1023 chips for C/A code, considering 1 sample per chip. Figure 5.2, shows the frequency and Code-Delay Search Space for GPS C/A SV1. 5.2a shows the two dimensional search grid, it contains 1023x52 cells, each cell corresponds to a possible value that the incoming Doppler frequency and the code delay can take. 5.2b shows the three dimensional acquisition matrix plot, it shows the correlation peak as a function of frequency and code-delay, it can be seen that the incoming signal has a peak on a Doppler frequency at 2000 Hz and a code delay of 1000 chips.

5.4 Acquisition methodology

The implementation of correlators for the acquisition unit differs from the tracking channel implementation as a wide Doppler range is considered, a wide code-phase range (typically the whole PRN code period) is considered and the accuracy requirements in terms of code-phase resolution or Doppler resolution are rather low. Thus different methods for code and Doppler correlation are considered. The following sub-sections provide different methods adopted fo acquisition.

5.4.1 Serial acquisition

This algorithm is based on multiplication of locally generated PRN code sequences and carrier with incoming signal. As illustrated in Figure 5.3a, the PRN generator generates a PRN sequence with a certain code phase, corresponding to a specific satellite. The generated PRN sequence is a sample based version with at least two samples per chip. For example if a sequence of 1 millisecond which is 1023 chips of GPS L1 C/A code sampled eight times, there would be 8184 different code phases. The local carrier generator generates a complex signal (I and Q with 90° phase shift) with an estimated frequency. The estimated frequency is sum of down-converted carrier and the frequency steps of Doppler range. In the worst case, the Doppler frequency can deviate up to ± 10 kHz. The frequency step size (Doppler bins) f_b is linked to the coherent integration time (T_C) (Leclère, 2014).



(a) Removing the carrier first.

(b) Removing the code first.

Figure 5.3: Principle of the serial search acquisition.

$$f_b = \frac{1}{2T_C} \tag{5.5}$$

This implies that the higher T_C is, the smaler f_b. If the coherent integration time is one millisecond, the frequency step size (Doppler bins) $f_b = 500Hz$.
The generated local carrier is sampled with the same PRN sequence sampling rate. For example, if a PRN code with chipping rate $f_c = 1.023MHz$ and a locally generated carrier with $f_{local} = 2.046MHz$ sampled with a frequency $f_s = 8.184MHz$, there will be eight samples per chip and four per the generated carrier.

After multiplying the incoming signal by localy generated PRN and carrier with estemated code phase and frequency step, the I and Q signals are integrated over T_C and finally squared and added. Ideally, the signal power should be located in the I part of the signal, as the C/A code is only modulated onto that part. Because the phase of received signal is unknown (carrier phase ambiguities), the generated I signal could not be correspond to generated I signal at the satellite. It is necessary to investigate both the I and Q channel to obtain output power due to correlation between incoming and locally generated signals (Borre et al., 2007).

5.4.2 Integration schemes

When the receiver is not with an open view to the satellite due to obstacles, the signal becomes too weak and this will affect the acquisition performance. In the case of weak signal the acquisition process requires long coherent or non-coherent integration. Increasing the coherent integration will increase the receiver sensitivity, but is limited to a specific time period because of the unpredicted navigation data bit transition and offset due to the receiver clock error. On the other side increasing the coherent integration will increase the frequency search space i.e. increases Doppler bin numbers in the search space. The non-coherent is less sensitive, because of squaring loss. There is a trade off between non-coherent gain and loss. More details can be found in (Lin and Tsui, 2000)(Garcia et al., 2010).

5.4.2.1 Coherent integration

Coherent Integration is the first step in any acquisition method to find the available satellites in the users scope. Coherent integration is referred to any method which correlates an input signal with the replica codes over time period equal or longer than the replica code period. The coherent integration scheme is shown in the Figure 5.3. In this scheme

the signal is accumulated over T_C ms with $T_C > T_{prn}$, where T_{prn} is the the spreading code period. The coherent integration provides the best performance in terms of noise reduction since the noise power at the output of a correlator is inversely proportional to T_C (Foucras, 2015). However, when longer coherent integration above data bit transition is considered, data bit transition during the correlation process degrades the performance due to the sign flip. The second drawback is that the frequency bin size is inversely proportional with T_C as provided in equation 5.5, i.e. increasing the coherent time increases the number of cells in the acquisition grid and then the search time. The third disadvantage is degradation due to the misalignment between the replica and the incoming code because of receiver clock error.

5.4.2.2 Non-coherent integration

The noncoherent integration is performed by combining the amplitude of several coherent matrices sequentially. Normally it is implemented after short periods of coherent integration and it is not influenced by the data bit transition. As shown in the Figure 5.4, in this scheme a number of the coherent integration outputs over T_C ms with $T_C > T_{prn}$, where T_{prn} is the the spreading code period, is computed and then added together (Borio and Akos, 2009).

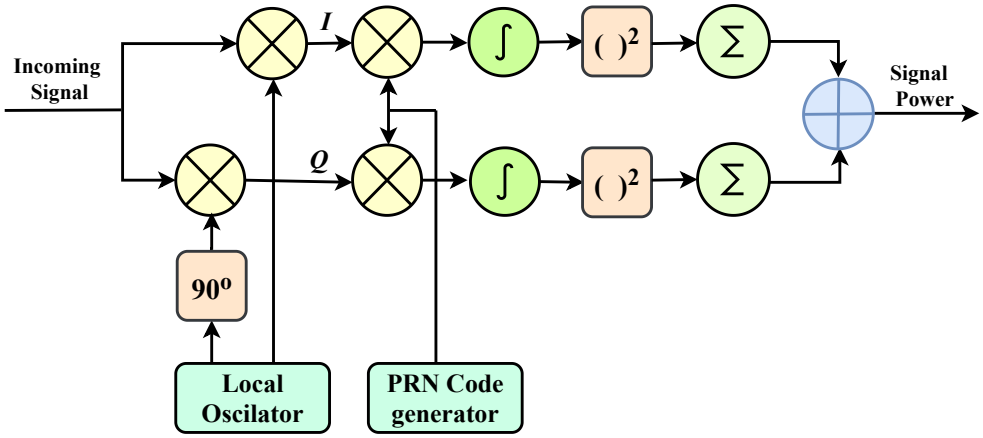


Figure 5.4: Non-coherent integration scheme.

5.4.3 Parallel search acquisition

Parallel processing is one of the solutions to reduce the acquisition time due to long calculation in serial algorithm. This could be accomplished by performing Fourier transform in one of the search spaces, code phase or Doppler frequency space.

5.4.3.1 Parallel frequency space search

As illustrated in Figure 5.5, the PRN generator generates a PRN sequence with a certain code phase, corresponding to a specific satellite. By approaching the right code phase, the locally generated PRN code is aligned with incoming signal code. The result is a continuous wave. The resulting signal is transformed into the frequency domain using discrete Fourier transform (DFT). The output of DFT will show an isolated peak. A peak will be located at the frequency index corresponding to the frequency of the continuous-wave.

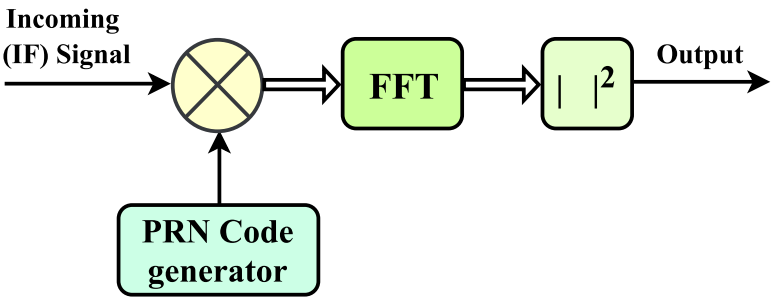


Figure 5.5: Parallel frequency space search diagram.

The accuracy of the determined frequency depends on the length of the DFT, in other words the number of samples of the incoming signal. If 1 ms of incoming signal sampled with a frequency $f_s = 8,184MHz$ analyzed using DFT, the number of samples $N = 8,184$. The DFT output is composed of 8184 frequency bins. The frequency resolution Δf_b of the output is:

$$\begin{aligned} \Delta f_b &= \frac{f_s}{N} = \frac{1}{T_C} \\ &= 1KHz \end{aligned} \tag{5.6}$$

The accuracy of the estimated carrier frequency is 1 kHz compared to the accuracy of 500 Hz in serial search acquisition. The serial search acquisition method steps through possible code phases and carrier frequencies, the parallel frequency space search acquisition only steps through the code phases. the maximum number of repeated computation for one satellite will be:

$$= N_{chip} \times \frac{f_s}{f_c} \tag{5.7}$$

5.4.3.2 Parallel code phase search acquisition

The frequency steps in parallel frequency search acquisition is performed with one DFT application, however the code space is still very wide. The code phase can be found by applying circular correlation using DFT and the inverse DFT. If the ACF in frequency domain is found, the time-domain representation can be found through inverse Fourier transform (Borre et al., 2007). As illustrated in Figure 5.6, the incoming signal is multiplied by a locally generated complex carrier, forming a complex signal $(I + jQ)$ input to Fast Fourier Transform (FFT) function and transformed to frequency domain. The PRN generator generates a specific code with no code phase, only once. The generated PRN code is transformed into the frequency domain and the result is complex conjugated. The result of both FFT functions multiplied and transformed into the time domain by applying inverse Fourier transform. The absolute value of the inverse Fourier transform represents the correlation between incoming PRN code and the generated PRN code. If the carrier striped off by multiplying the incoming signal with correct frequency through the steps a peak will aper in the correlation, the index of this peak represents the PRN code phase of the incoming signal. For each of the frequency steps only one Fourier transform and one inverse Fourier transform is performed. The calculation efficiency of this method depends on the implementation of FFT functions.

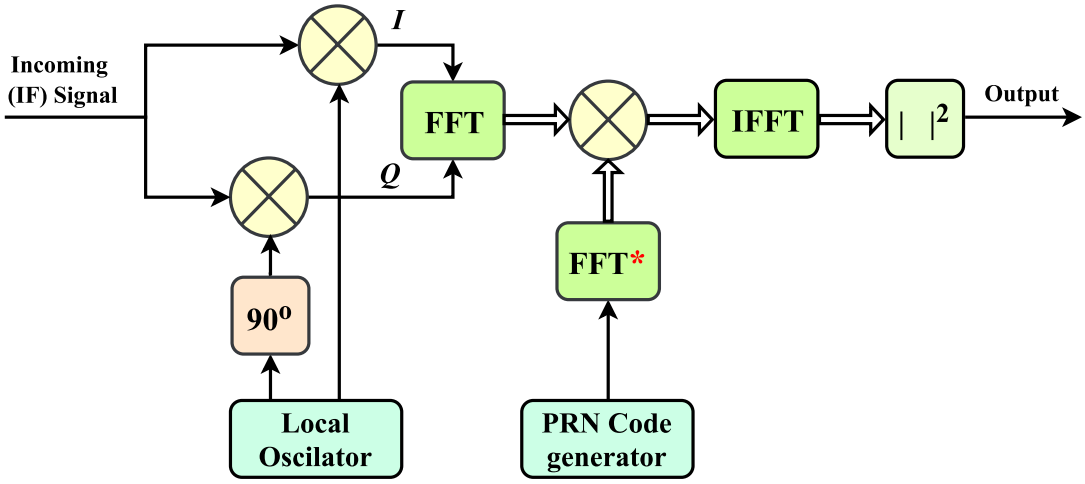


Figure 5.6: Parallel code phase search algorithm diagram.

5.4.4 Double-block zero padding

The double-block zero padding (DBZP) (Lin and Tsui, 2003b) is an improved method for the acquisition of GNSS signals. This method performs long coherent integration with reduced operations and higher sensitivity than other FFT-based techniques. DBZP implements many partial correlations over part of the incoming signal and generated replica codes, unlike the conventional correlation which uses the complete incoming signal duration. This is obtained by dividing the incoming data signal and generating replica codes into smaller parts. The input parameters of the DBZP (defined by the user) are (Foucras, 2015):

- The coherent integration time T_C. In general, the coherent integration time is chosen to be at least equal to the spreading code period or a multiple of that period.
- Doppler uncertainty range $[f_{Dmin}, f_{Dmax}]$, where f_{Dmin} and f_{Dmax} are the minimum and maximum expected values of the incoming Doppler frequency, respectively. The central frequency of the Doppler shift range is denoted as f_{Dmid}.

The number of DBZP blocks which represent Doppler frequency bins and their resolutions are fixed by the algorithm and cannot be chosen by the user, unlike the conventional search acquisition method.

- The number of blocks (Doppler frequency bins) M is determined by:

$$\begin{aligned} M &= \left(\frac{f_{Dmax} - f_{Dmin}}{\frac{1}{T_C}} \right) \\ &= 2 \times f_{Dmax} \times T_C \end{aligned} \tag{5.8}$$

- The duration of one block t_b is:

$$\begin{aligned} t_b &= \frac{T_C}{M} \\ &= \frac{1}{2 \times f_{Dmax}} \end{aligned} \tag{5.9}$$

- The number of samples per block N_b is equal to:

$$\begin{aligned} N_b &= \frac{N}{M} \\ &= t_b \times f_s \end{aligned} \tag{5.10}$$

where
N is the total number of samples across T_C,
f_s is the sampling frequency.

- The Doppler frequency resolution Δf is:

$$\begin{aligned} \Delta f &= \frac{2 \times f_{Dmax}}{M} \\ &= \frac{1}{T_C} \end{aligned} \tag{5.11}$$

The Doppler frequency resolution is twice as wide as that of the serial search acquisition method for which $\Delta f = \frac{1}{2T_C}$. The DBZP acquisition method can be described in five steps.

- Forming double blocks.
- Forming zero padded blocks.
- Performing circular correlation using DFT.
- performing DFT.
- Permutation of code or data blocks.

5.4.4.1 Forming double blocks

The incoming signal is converted to the baseband by multiplying it by a complex locally generated carrier $e^{(-i2f_{IF}nT_s)}$. Where f_{IF} is the IF without any compensation of the estimated Doppler, which means that the local complex carrier does not try to compensate the incoming Doppler frequency by cycling through the frequency bins (unlike the conventional search method). The obtained T_C-long baseband samples are divided into M blocks of equal length. Each pair of two consecutive blocks of N_b samples concatenated to form double blocks of size $2N_b$ samples (thus the name"Double Block"). The last block is combined with additional samples as illustrated in Figure 5.7. Note: It is recommended to sample the incoming signal at least with $(T_C + t_b)$ to use the extra length to form the last double block.

5.4.4.2 Forming zero padded blocks

T_C milliseconds of the local code are generated and divided into M blocks of N_b samples each. Then, each block is zero-padded, which means that N_b samples of zero value are appended to each block as illustrated in Figure 5.8, the zero value blocks are represented by a grey box. When the locally generated PRN codes aligned with incoming signal PRN codes, the normalized partial auto correlation is as same as normalized full autocorrelation. The drawback of the partial correlation is that the correlation is done on only a part of

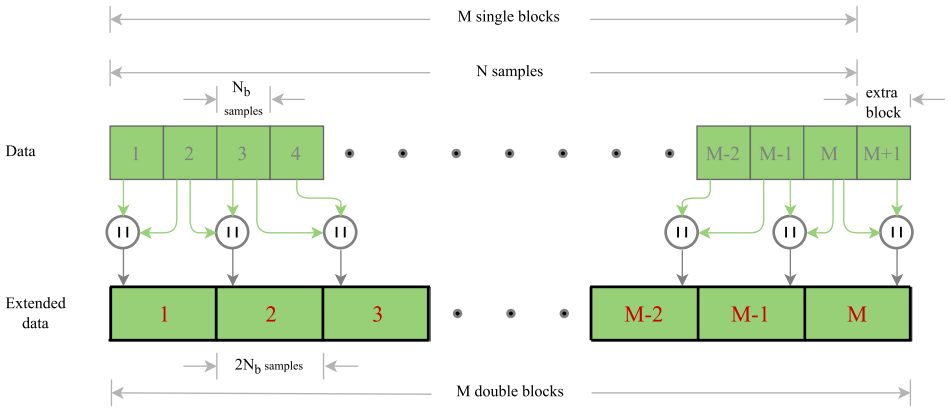


Figure 5.7: Forming double blocks.

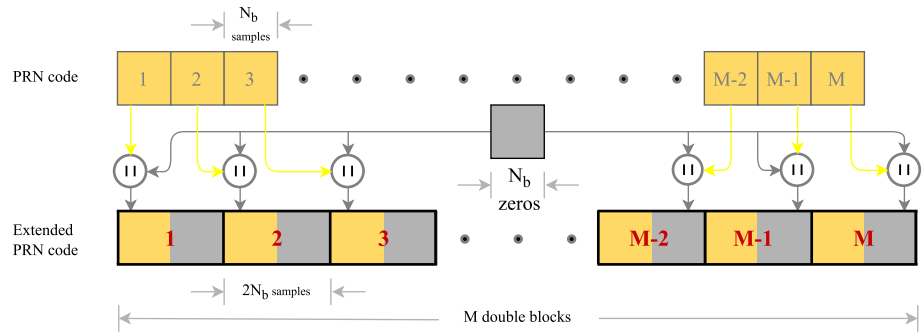


Figure 5.8: Forming zero padded blocks.

the whole spreading code and thus the properties of the spreading code is not maintained (the auto correlation peak isolation is attenuated) (Foucras, 2015). Doubling the incoming signal adds continuity to the partial code. When the partial correlation is computed using $2t_b$ of signal and zero-padding the local partial code, the autocorrelation function peak is highly isolated and not attenuated.

5.4.4.3 Performing partial circular correlation on M blocks

The first $2N_b$ samples block of the incoming signal is circularly correlated with the first zero-padded code block. This results in a partial circular correlation, and only the first half is preserved as shown in Figure 5.9. The preserved N_b output samples represent a partial correlation on t_b ms (much shorter than a spreading code period) over N_b possible code delays.

5.4.4.4 performing DFT on each column

The partial correlator outputs can be stored in a matrix of size $M \times N_b$ as illustrated in Figure 5.10, where:

- There are as many columns as possible code delays. Each column contains all partial correlator outputs for a given code delay error.
- There are as many rows as partial correlations, with each row containing the partial correlator outputs for a given slice of time.

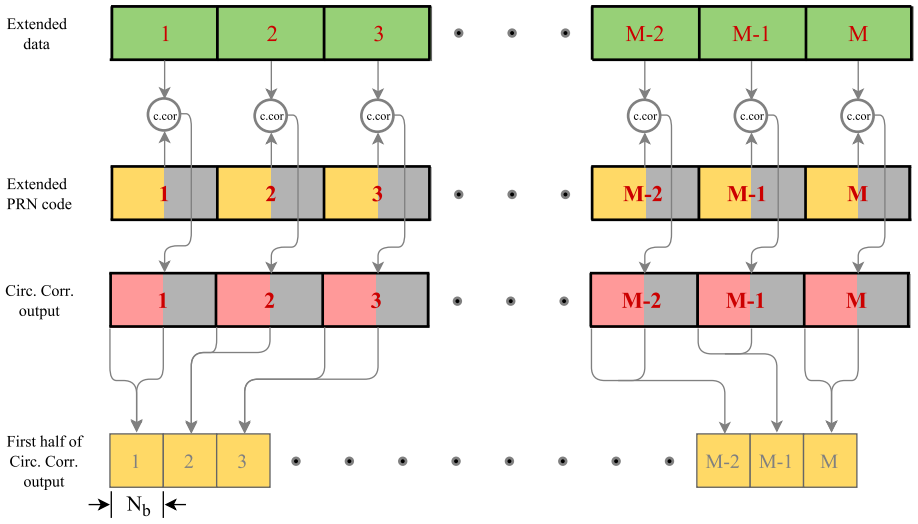


Figure 5.9: Performing circular correlation.

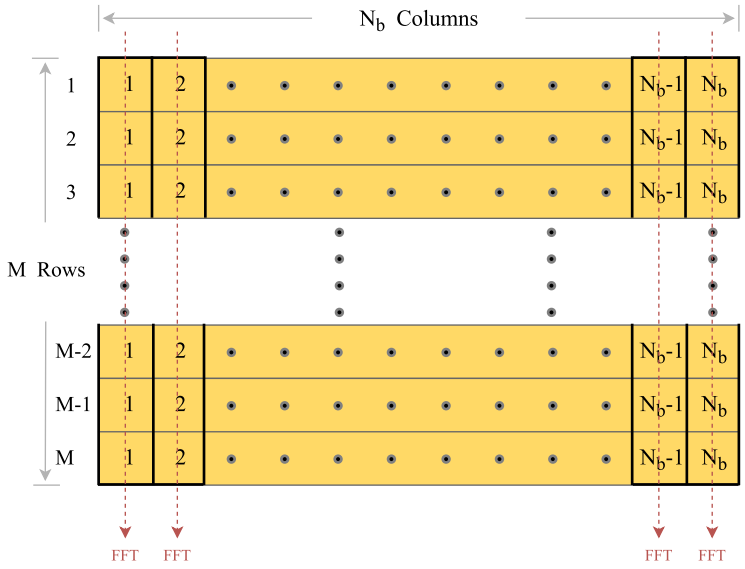


Figure 5.10: Performing FFT.

An N_b-point FFT is applied to the set of the partial correlation outputs corresponding to a given code delay. This permits the determination of the Doppler frequency of the incoming signal.

5.4.4.5 Permutation of data blocks or code blocks

In the process previously described, only code delays in the first code delay time slice $[0, t_b]$ are tested. To investigate all code delays, the local code blocks or the data blocks are circularly permutated, for example if the code blocks are permutated, the data blocks kept fixed and vice versa, the N_b-th block becomes the first block, the first block becomes the second block, etc, and the above steps repeated until the code phase and doppler frequency found. If one sequence of the ranging code is less than the incoming data duration, there will be a need to circulate the blocks to cover the code delays only in one sequence duration. If the coherent integration time T_C is equal to the spreading

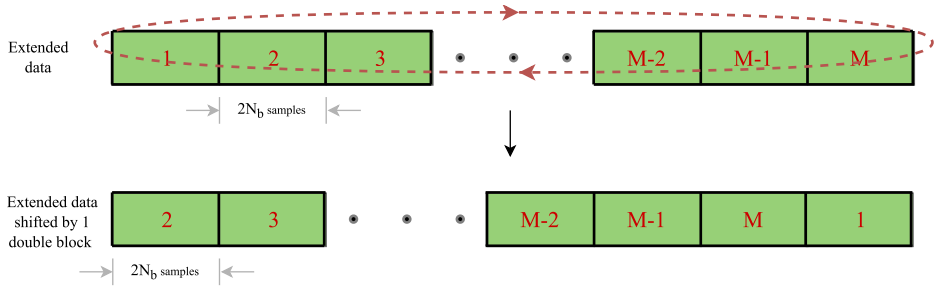


Figure 5.11: Permutation of data blocks.

code period,the number of maximum circular permutations (C_p) will be the same as the number of blocks M, that is, $C_p = M$. If T_C is longer than one spreading code period, for example a GPS L1 C/A signal with $T_C = 10$ ms and one PRN period T_{C1} of 1 ms, the number of maximum circular permutations reduces to $C_p = \frac{M}{T_C/T_{C1}}$ because of the spreading code periodicity.

5.4.4.6 Strengths of the DBZP algorithm

- Calculation efficiency
  "For one acquisition, there are 2.2 times less elementary operations (additions and multiplications) for the DBZP algorithm in regard with the Reference Acquisition. There are 1.3e8 operations for the DBZP algorithm and 2.7e8 for the Reference Acquisition, these numbers are obtained with optimized FFT algorithm" (Foucras et al., 2012).

- Signal-to-Noise Ratio
  There is no additional noise when acquiring with the DBZP acquisition method compared to the classical acquisition method because the (SNR) at the correlator output and at the DBZP output are the same. Both acquisition methods are equivalent in terms of SNR at the output (Foucras, 2015).

- Width of the peak
  In the frequency domain, the width of the main peak (for the right code delay and for the right incoming Doppler frequency) is the same for the DBZP acquisition method $\frac{1}{T_C}$ (Foucras, 2015).

5.4.4.7 Weaknesses of the DBZP algorithm

- Code Doppler impact on partial correlations
  The spreading code Doppler of incoming signal can have a significant impact on the acquisition performance, typically with high Doppler shift frequencies because the Doppler changes the spreading code period. The last partial autocorrelation term will be attenuated.

- 50% of the power consumption and time during the partial correlation process is lost because of performing correlation on doubled blocks (two partial periods) and maintaining the output from only the first block.

- Real-time implementation of DBZP algorithm in hardware is not feasible due to processing complexity such as sharing the same block between two double blocks which requires long buffering size of the incoming signal.

5.5 Comparison between acquisition methodologies

This section provides a qualitative assessment to compare the performance of the different acquisition methodologies in FPGA. Assuming a base-band Galileo E1b BOC(1,1) modulated signal with a coherent integration time $T_C = 4millisecond$, sampling frequency $f_s = 5,12MHz$, minimum Doppler frequency $f_{Dmin} = -10KHz$ and maximum Doppler frequency $f_{Dmax} = 10KHz$ is used for acquisition with a 100 MHz system clock. The rational for selecting this specific sampling frequency is clarified in sub-section 5.5.4.
The total number of the samples N within T_C will be:

$$N = N_{chip} \times \frac{f_s}{f_c} \tag{5.12}$$
$$N = 20480$$

Where, N_{chip} is number of chips in one PRN sequence and f_c is chipping rate frequency.

5.5.1 Serial acquisition

The serial search algorithm (Figure 5.3) demodulates the carrier signal by cycling through all possible frequencies affected by Doppler shift in steps of not larger than as given by Eq. 5.5

$$f_b = \frac{1}{2T_C} = 125Hz$$

This approach demodulates the PRN code through N different phases, Theoretically, the maximum number of clock cycles needed to perform serial acquisition on one code-phase (C_{cp}) of the incoming signal will be:

$$C_{cp} = N\left(\frac{[f_{Dmax} - f_{Dmin}]}{f_b} + 1\right) \tag{5.13}$$

For N code-phases which is the number of clock cycles for computation of one satellite search (C_{sv}) will be:

$$C_{sv} = N \times N\left(\frac{[f_{Dmax} - f_{Dmin}]}{f_b} + 1\right) \tag{5.14}$$
$$C_{sv} = 67.528.294.400$$

The required time for computation of one satellite search assuming 100 MHz system clock (t_{sv}) will be approximately:

$$t_{sv} = \frac{67.528.294.400}{100 \times 10^6} = 675\ sec$$

This leads to an inefficient acquisition due to a long computation. This is especially true when the frequency search space is large, or with modern signals having long codes. This

algorithm is usually performed in an application specific integrated circuit (ASIC) because the data could be processed in parallel. This long computation burden is possible to be reduced by taking advantage of the parallel processing implementation in FPGA. For example when it is possible to process 250 code phases in parallel, the required time for computation of one satellite search assuming 100 MHz system clock will be approximately:

$$t_{sv}\ utilising\ 250\ parallel\ phases = \frac{675}{250} = 2,7\ sec$$

5.5.2 Parallel frequency space search acquisition

The parallel frequency space search algorithm (Figure 5.5) demodulates the PRN code through N different phases.
This method is only steps through the code phases, while the frequency is found using DFT in one step.
An additional N clock cycles latency (size of the transform) is required when utilising a FFT block in FPGA. Therefore, the maximum number of clock cycles for one code-phase search will be $2N$. For N samples, there will be N code phases. If processing N samples for N code phases are excuted consigutively, there will be only N samples latency. The maximum number of clock cycles for all code-phase search will be $N \times N + N$.
For N code-phases which is the number of clock cycles for computation of one satellite search (C_{sv}) will be:

$$C_{sv} = (N)^2 + (N) = 419.450.880$$

The required time for computation of one satellite search assuming 100 MHz system clock (t_{sv}) will be approximately:

$$t_{sv} = \frac{419.450.880}{100 \times 10^6} = 4,2\ sec$$

5.5.3 Parallel code phase search acquisition

The parallel code search algorithm (Figure 5.6) demodulates the carrier signal by cycling through all possible frequencies affected by Doppler shift in steps of not larger than as given by Eq. 5.5

$$f_b = \frac{1}{2T_C} = 125Hz$$

This approach is only steps through the possible Doppler bins, while the code-phase is found using circuller correlation method in one step.
An additional 2N clock cycles latency (size of the transform) is required when utilising a FFT and IFFT block in FPGA. Therefore, the maximum number of clock cycles for one frequency bin search will be $3N$. The number of clock cycles for computation of one satellite search (C_{sv}) The maximum number of clock cycles for all frequency bin search which is computation of one satellite search (C_{sv}) will be:

$$C_{sv} = \left(\frac{[10000 + 10000]}{125} + 1\right) \times 3N \tag{5.15}$$
$$C_{sv} = 9.891.840$$

The required time for computation of one satellite search (t_{sv})assuming 100 MHz system clock will be approximately:

$$t_{sv} = \frac{9.891.840}{100 \times 10^6} = 99\ ms$$

5.5.4 Double-block zero-padding acquisition

The number of DBZP blocks M which represent Doppler frequency bins and their resolutions are fixed by the algorithm and obtained from Eq. 5.8

$$M = 2 \times 10 \times 10^3 \times 4 \times 10^{-3} = 80$$

The number of samples per block N_b obtained from Eq. 5.10

$$N_b = \frac{20480}{80} = 256$$

It is important to point out that the sampling frequency should be chosen carefully to produce samples per block with an integer radix 2 number, to comply with FFT block transform size in the FPGA and avoid obtaining fractional numbers per block. In addition, the real time Implementation of this algorithm in FPGA is very complicated due to performing correlation on double blocks formed from the current and previous samples which needs large size buffers to pipeline the incoming data.

The modern satellite system signals have longer spreading codes, which makes the implementation of this algorithm very complicated in FPGA. Therefore, for the seek of this comparison, we assume that the signal is recorded in a sampler memory and ready to re-play for processing.

In the first step, partial correlation will be performed on each 2M blocks for 80 times and only the first half is saved in the memory. This will need 2 x 256 x 80 = 40.960 clock cycle. This process will include an additional 512 clock cycles latency (size of the transform) for FFT and another 512 for IFFT and 512 x 80 clock cycles delay for writing the results in the memory before applying DFT on each column. This makes the total clock cycles 40.960 + 1024 = 41.984 Clock cycles. Secondly, FFT will be performed on the 256 column, each 80 point long. Transforming 80 point in FPGA will require 128 point length block i.e. 256 x 128 = 32.768 clock cycles. This process will include an additional 128 clock cycles latency (size of the transform) for FFT. This concludes that the first 256 code-phases within the first block will require (40.960 + 32.768) and (512 + 512 + 40.960) latency clock cycles. To search all code phases, the sampled signal should be permutated 79 more times. Assuming that the process of all 80 blocks performed consecutively, the latency will be occurred only at the first round, i.e. processing the other 79 will be without any delay. Therefore, the maximum number of clock cycles for computation of one satellite search (C_{sv}) will be:

$$C_{sv} = 80 \times 73.728 + 42.112 = 5.940.352$$

The required time for computation of one satellite search assuming 100 MHz system clock (t_{sv}) will be approximately:

$$t_{sv} = \frac{5.940.352}{100 \times 10^6} = 59.4\ ms$$

5.6 Summary

Satellite signal acquisition includes a search in a two-dimensional space, which is defined by the code phase and Doppler frequency shift. The search is typically performed through the PRN code correlation operation. The aim of acquisition is to detect the visible satellites in users scope. GNSS receiver effectiveness is usually determined by the mean acquisition time. Acquisition is the first step of the signal processing for a GNSS receiver, and it is a time-consuming process. Due to the satellite and receiver movement and receiver clock error, the Doppler shift changes greatly, resulting in a larger frequency search range and longer search time.

There are different acquisition methodologies including serial search acquisition, parallel frequency space search and parallel code-phase search and DBZP acquisition. The serial search approach performs the two-dimensional searches, but its efficiency is limited due to a long computation, especially when the frequency search space is large, or with modern signals having long codes. This algorithm is usually performed in an ASIC because the data could be processed in parallel. The parallel frequency space search is a search algorithm based on the FFT. This technique uses the FFT to compute all possible frequency bins (in frequency space) in one step, thereby eliminating the time-consuming code phase shift process. The parallel code phase search uses circular correlation to speed up the correlation. This method derives a code shift for a single satellite with FFT/IFFT operations. Finally, the DBZP algorithm improves the acquisition method by reducing the number of operations in the block correlation used in determining Doppler frequency and time of the received GNSS signal.

Chapter 6

Proposed system

A reconfigurable acquisition engine capable of performing DSP algorithms in an efficient manner can be obtained using a general-purposee system on chip (SoC) (ul Haq et al., 2009) (Huang et al., 2009). SoC is an integrated circuit (IC) which integrates the software programmability of a processor such as ARM with the hardware programmability of an FPGA, enabling key analytics and hardware acceleration while integrating the central processing unit (CPU), DSP, and mixed signal functionality on a single device. The SoC components are connected through a communication bus using specific protocols. The advantage of the FPGA is that the programmable fabric can be configured to implement any combination of digital processing. In addition, algorithms can be implemented in a parallel method, which means that a substantial amount of data processing can be performed very quickly and efficiently. The FPGA can be used to process front-end IF sample streams, such as down-converting to baseband, acquiring, and tracking GNSS satellites. The on-chip processor is used to manage all internal units, control the DSP data path, and process the acquisition and tracking results to calculate the PNT solutions.
In general, FPGAs are static random access memory (SRAM) based devices, the SRAM is volatile memory and lose the data when they are powered off, therefore FPGAs lose their programming each time when they are powered off. Due to this, the configuration data (bitstream) should be saved in an external memory. The configuration data source could be at its simplest a flash memory component or stored on a network or any other device such as a PC. The SoC firmware version can be upgraded to include the latest techniques. This can be achieved by over-the-air (OTA) or online firmware upgrade through UART, USB or communication port such as Ethernet port. The flash memory of the device is divided into two sections, boot-loader and boot-loadable. Bootloader is the section of memory that stores the code that is responsible for the device's booting operation, checks if the host has an updated firmware image for the boot-loadable section, receives the updated firmware image from the host over an Ethernet interface and writes the updated image to the boot-loadable section of memory. The boot-loadable section of memory contains FPGA configuration data and the actual application code that defines the system's functionality. More details about remote firmware upgrading can be found in (Jurkovic and Sruk, 2014) (Shade, 2011).

6.1 Building in the FPGA

Unlike the conventional correlation scheme which uses the complete incoming signal duration, the DBZP algorithm implements many partial correlations over part of the incoming

data signal and generated replica codes. Implementing partial correlation instead of full-length correlation results in using less logic resources in the FPGA. This makes the DBZP an attractive algorithm to be implemented in the hardware.

To implement the DBZP algorithm in a more efficient manner, faster DBZP (FDBZP) method is proposed. In this section, a detailed design for developing FDBZP in the FPGA is provided, which is called the Acquisition Block. Acquisition Block is the incarnation of the FDBZP method in the FPGA, which is able to acquire modernised satellite signals in a very efficient method with no need for forming the double blocks and zero padding the code, without using any third-party IPs such as FFT, DSP, or multipliers. The Acquisition Block is a stand-alone core, designed using only the Notepad++ code editor and ModelSim for debugging and simulation. MATLAB is used to synthesise the Galileo signal and data analysis. The design can be targeted to any FPGA platform.

Figure 6.1 shows a generic block diagram of a SoC based GNSS receiver. Acquisition Block will be responsible for the GNSS signal acquisition in a hardware based receiver. Note that the diagram does not illustrate the other parts of the receiver such as tracking, down-converting components. The Rf Front-End down-converts the L band signal to the

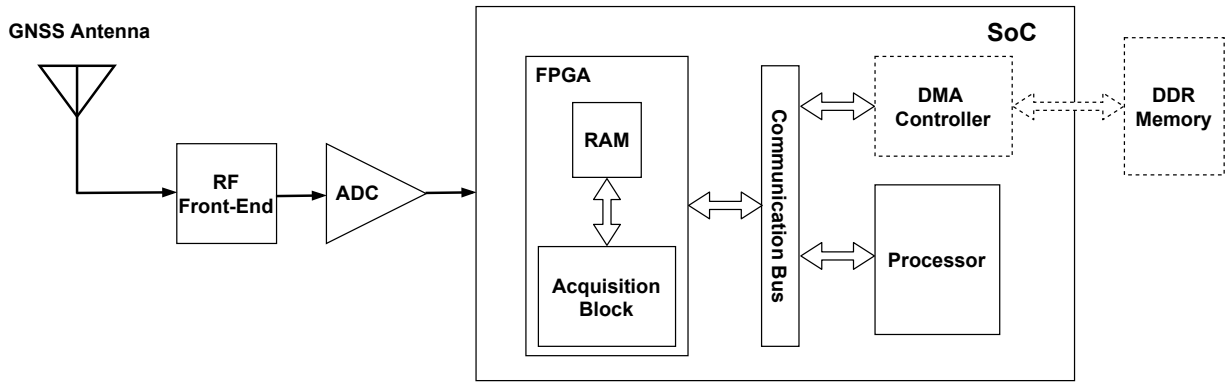


Figure 6.1: General block diagram of a GNSS SoC based receiver.

IF signal. The ADC digitises the IF signal for signal processing purpose in the hardware. A down-converter in the FPGA down-converts the digital IF to the base-band signal. The Acquisition Block acquires the available satellites in the user scope as a stand-alone core and informs the processor with the acquisition result through the connection bus dedicated for communication between the processor and FPGA or the external on board components through a bridge. The on chip processor manages all internal units, control signal processing data paths and process the acquisition and tracking results to calculate PNT solutions. The DDR memory is an optional component which depends on the saving size requirements.

For the proof-of-concept of building FDBZP acquisition system in the FPGA, able to acquire the modernised GNSS signals, the Galileo E1 OS signal is taken as a sample for this purpose. As discussed in sub-section 3.3.3.3, E1 OS deploys optimised primary (memory) codes.The data channel deploys primary codes of 4092 chips long with a 4 ms period at 1.023 Mcps. The pilot channel deploys tiered codes of 25 x 4092 chips, a 100 ms period with 1.023 Mcps, and the tiered codes are generated using 4092 chip long primary codes at 1.023 Mcps with 25 chips secondary code at a chipping rate of 250 cps. The E1 OS modulation scheme is CBOC. The E1 OS CBOC modulation scheme is a result of multiplexing a wideband signal BOC(6,1) and a narrowband signal BOC(1,1) with two different power levels; only 1/11 of the power is allocated, on average, to the BOC(6,1)

component.

The E1 OS signal can be acquired using only the narrowband component, BOC(1,1). The signal can be reformed by sampling at 5 MHz. The advantage of this approach is that a lower sampling rate is required to reconstruct the signal, which reduces the resource requirement in the hardware. In this section, the E1 OS data channel (E1b) signal component for acquisition is considered and it is assumed that there is no bit transition occurring in the sampled signal, with the following parameters:

- Coherent integration time $T_c = 4ms$ (signal period),
- Doppler uncertainty range = $[-10000, 10000]$,
- Signal sampling rate $f_s = 5MHz$,
- Quantisation = 3 bit (signed),
- Number of blocks (Doppler frequency bins) $M = 80$, (Eq. 5.8)
- Number of samples per block $N_b = 250$, (Eq. 5.10)
- Total number of samples $N = 80 * 250 = 20000$.

The E1 OS signal is down-converted to a complex base band at the front end. When the incoming signal is an IF signal with a specific frequency, the signal can be down-converted to a complex baseband in the FPGA by multiplying with a complex replica carrier. Note that the signal is converted to the baseband, but the Doppler frequency is not compensated because of the uncertainty, that is, the baseband is still modulated with the Doppler frequency. The Doppler is estimated to be $[-10000,\ 10000]$.

6.1.1 Acquisition Block

In this subsection and the subsequent subsections, the circular shift of the M blocks data set by one block is denoted as permutation. Figure 6.2 shows a functional block

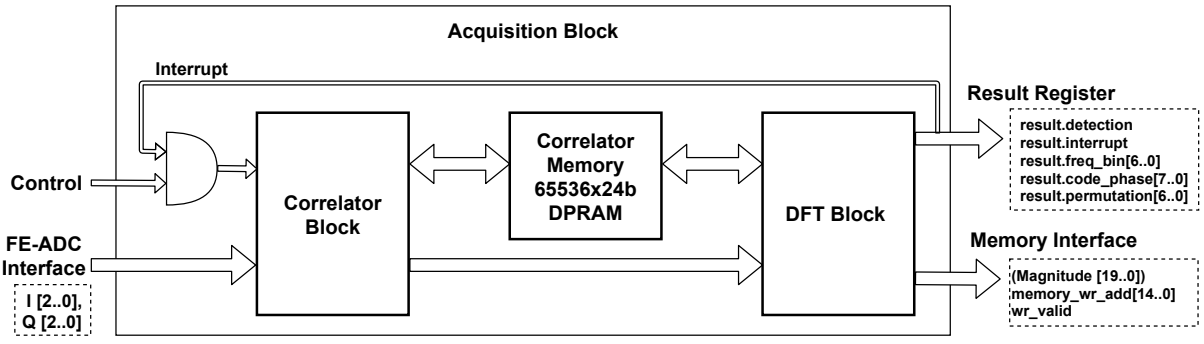


Figure 6.2: Acquisition Block functional block diagram.

diagram of the Acquisition Block. The block is composed of three components: the Correlator Block, Correlator Memory, and DFT Block. The system takes control signals from the user and three bit I and Q samples from the ADC RF front-end interface. The block outputs the acquisition results, such as the code-phase, Doppler frequency bin, and acquisition matrix. The correlation results are written to registers for read purposes by the processor, which includes a permutation cycle number which contains the correlation

peak, the code-phase, Doppler bin, and the result of the acquisition in logic mode, where logic 1 indicates detection and 0 indicates no detection. The acquisition matrix is a 20 bit absolute (magnitude) value data stream saved in a RAM, which might be an internal dual-port memory block in the FPGA or an external on-board memory such as double data rate (DDR) (depending on the design and on-chip memory block size) for subsequent processing stages. The size of the acquisition matrix depends on the number of blocks and samples per block. For 80 blocks and 250 samples per block, each matrix will contain 20000 samples. The block provides an acquisition memory interface with the required signals for writing purposes. The following subsections provide detailed descriptions of each subsystem.

6.1.1.1 Correlator Block

Figure 6.3 shows a functional block diagram of the Correlator Block. The block is composed of three elements: a Correlator Sampler Dual-Ported RAM (DPRAM), Correlator Coder, and Correlator.

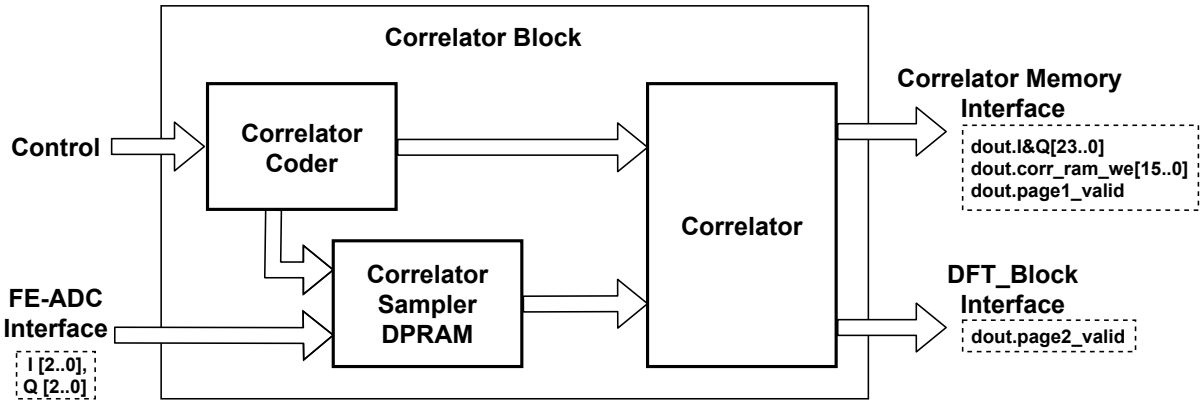


Figure 6.3: Correlator Block functional block diagram.

6.1.1.1.1 Correlator Sampler There are two approaches for processing incoming sampled signals from the front end. In the first approach, streaming data are processed at the sampling rate in real time. In the second approach, the signal is recorded in memory and replayed at a higher rate. Although the second option requires additional storage, it allows a significant gain in processing time if the recorded data are clocked at a higher rate. For example, if the signal is sampled and saved at 5 MHz, then it is clocked out from the memory at 200 MHz, the processing time is 40 times faster than processing the signal in real time. The disadvantage of this method is an error in the code phase owing to the computation time delay, which can be solved by compensating for the error in the code phase if the computation period is known. In this thesis, the record and replay method is adopted.

The Correlator Sampler is a 32768x8-bit dual port RAM (DPRAM) to hold complex (I and Q) baseband samples, records, and replays the data at different rates. Port A is dedicated to writing to the DPRAM, while port B is dedicated to reading purposes. The input of the DPRAM is 3 bit (signed)I and Q sample stream, concatenated to form six bits, from the ADC front-end interface. The output is connected to the correlator interface. The read and write address and control signals are provided by a control unit inside the Correlator Sampler.

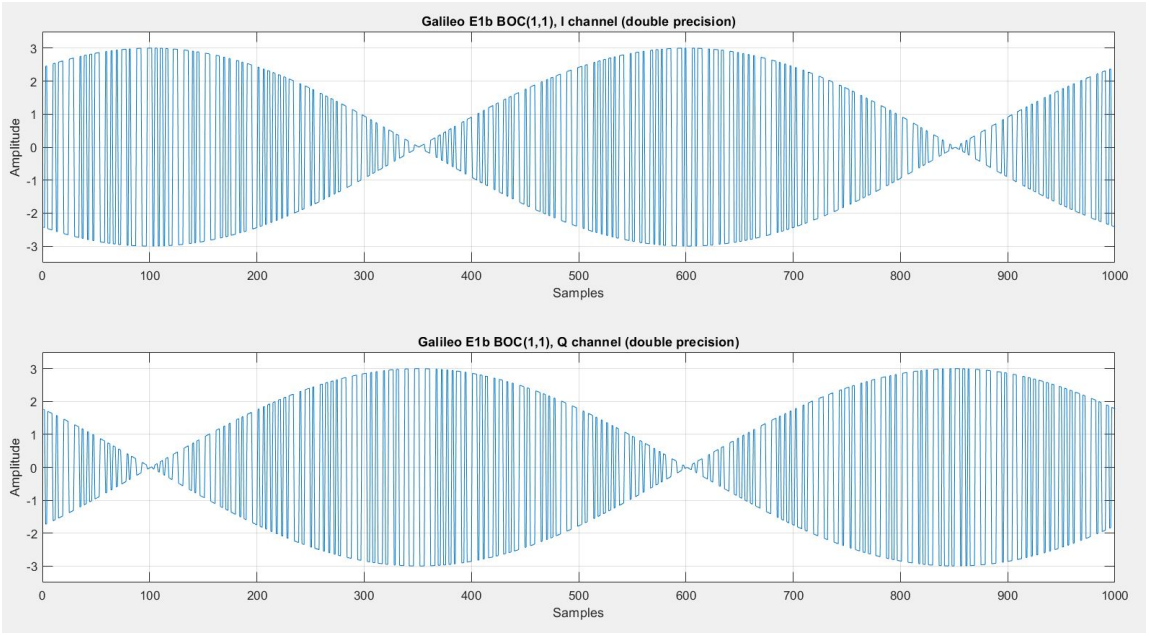


Figure 6.4: The first 1000 I and Q samples of Galileo E1b code for SV1 with 100 sample code phase, modulated with BOC(1,1) sub-carrier and a complex carrier at 5 KHz in double-precision format synthesised in MATLAB.

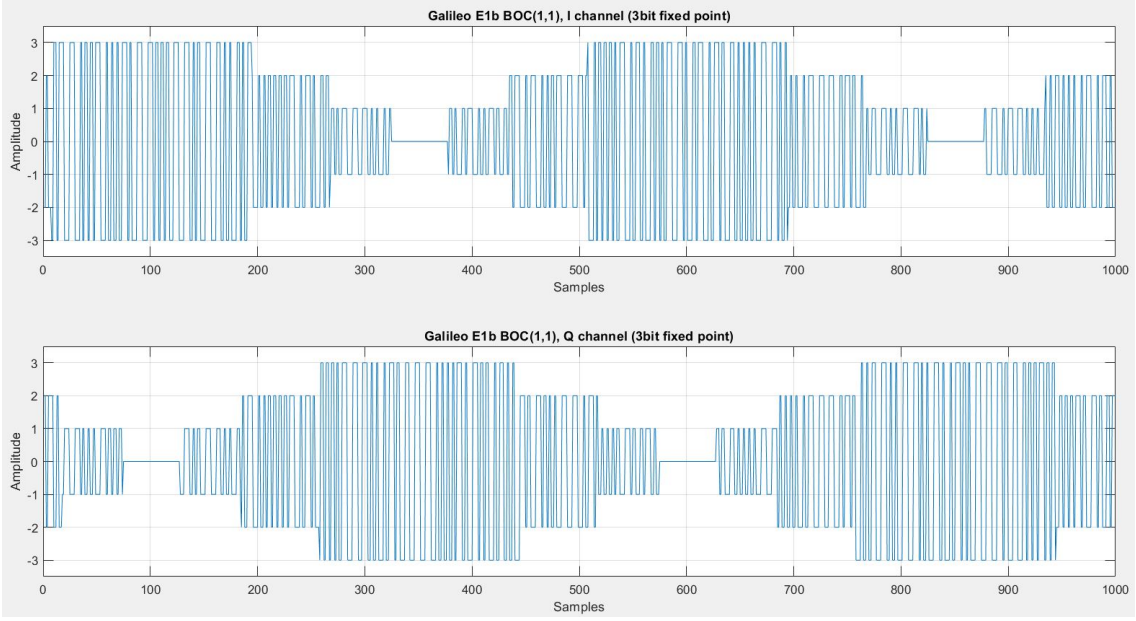


Figure 6.5: The first 1000 I and Q samples of Galileo E1b code for SV1 with 100 sample code phase, modulated with BOC(1,1) sub-carrier and a complex carrier at 5 KHz after converting to 3 bit (signed) fixed-point precision format synthesised in MATLAB.

The DPRAM is 8 bit wide, the 3 bit Q samples occupy 2 down to 0 bits, while the 3 bit I samples occupy 6 down to 4 bits of a total of 8 bits. In the record mode, the user enables the sampler control unit to provide the sampler memory with a write address at 5 MHz through a counter and a valid signal for a specific time (the coherent integration time). In the replay mode, the user will enable the sampler to drive the data at a much higher speed than the sampling rate through a counter.

The DPRAM is initialised with a pre-known pattern for system verification and debugging. The pattern is a synthesised Galileo E1b code for satellite vehicle one with a known code phase (100 samples), multiplied by the BOC(1,1) sub-carrier (at twice the frequency of the code) and modulated with a complex carrier at 5 kHz, which represents the Doppler frequency. The length of the pattern is 4 ms, sampled at 5 MHz, results in 20000 samples. The pattern is synthesised in MATLAB in a double-precision format and converted into a 3 bit (signed) fixed-point format. Figure 6.4 and 6.5 show the first 1000 samples of the pattern in double-precision and 3 bit fixed-point, respectively. In the test mode, the user can drive the pattern to the correlator with the system clock to test the system without the need to record the real data. The advantage of this approach is that testing a newly developed system with a pre-known pattern such as code identification (ID), code phase, and Doppler simplifies the verification steps by avoiding cycling through different allocated codes and using potentially corrupted data during the recording process.

6.1.1.1.2 Correlator Coder Figure 6.6 shows the functional block diagram of the Correlator Coder. The block is composed of three elements: address counter and control signal generator, PRN code RAM, and Shifter. This block accepts the control signals from the user and the DFT block which generates 250 code phases in parallel for correlation purposes in the Correlator element and provides the Correlator Sampler and the Correlator interface with the required control signals for enabling purposes.

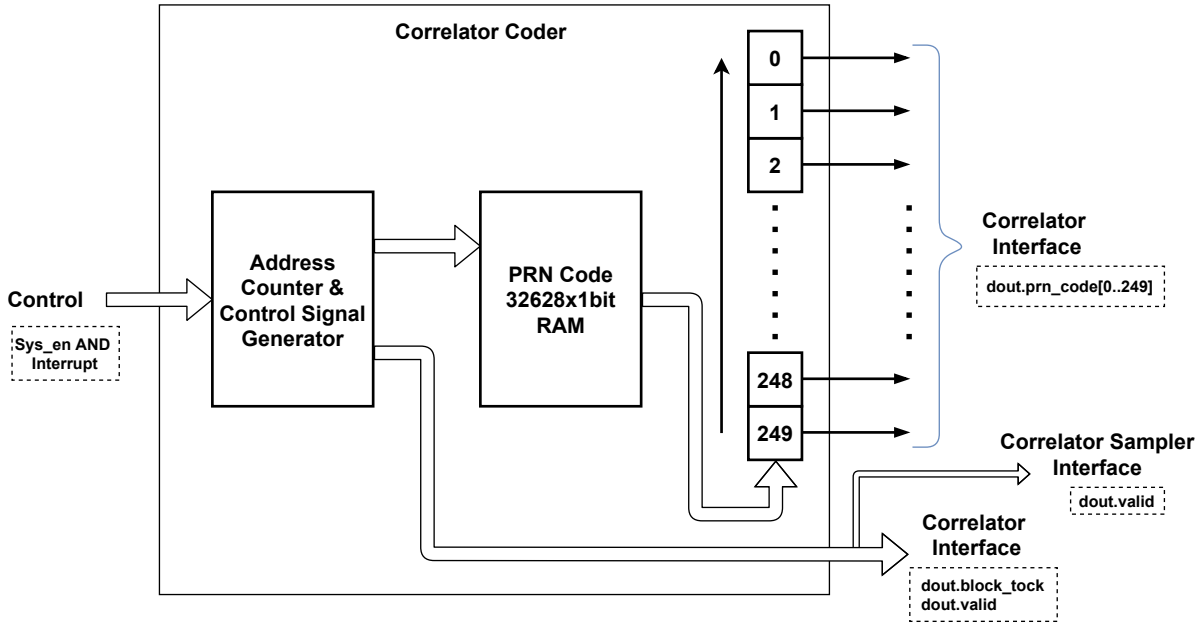


Figure 6.6: Correlator Coder functional block diagram.

Address Counter and Control Signal Generator (ACACSG) This element drives the PRN Code RAM by repeatedly providing the memory read address from 0 to 1999 through a counter when the system is enabled. The address counter and control signal generator ACACSG generates control signals (dout.valid) to enable both the Correlator Sampler and Correlator. The ACACSG provides the Correlator with a control signal (dout.block-tock) to reset the Correlator and save the final result after performing the correlation on each block of the FDBZP which contains 250 samples.

PRN Code RAM There are two approaches to generating replica codes: using shift registers or saving codes in the memory. Galileo E1b and E1c codes are optimised codes which cannot be generated using shift registers; they should be saved in memory (Borre et al., 2007). To obtain the BOC(1,1) sequence, the code must be multiplied by a subcarrier with twice the code chipping rate or save the BOC(1,1) sequence directly in the memory, which reduces the modulation process inside the FPGA at the cost of a larger memory size. In this thesis, saving the BOC(1,1) sequence in RAM blocks is adopted. The different sequences belonging to different satellites are saved in internal RAM blocks and launched accordingly; for example, if the Correlator Block performs the correlation with a specific code ID, the RAM block which contains the different sequences will launch only that specific code. To reduce memory block utilisation, the sequences can be saved in an external memory block such as a DDR. Only one sequence is loaded to the internal RAM blocks whenever that code is required for processing. This approach significantly reduces the use of internal RAM blocks. In real-time processing, when the sequence is launched from the RAM block, it should be re-sampled at the same rate as the incoming signal from the ADC front-end. In another approach, the sequence can be saved with a sampled version, which enables a faster processing speed above the sampling rate when the record and replay technique is adopted. In this thesis, to prove the concept, the sampled sequence method is implemented, and the sequence of BOC(1,1) for Galileo SV1 of 4 ms is sampled at 5 MHz (producing 20000 samples) and saved in the PRN Code RAM memory.

Shifter The shifter generates 250 code phases. This phase array is used in the Correlator element to determine the code phase of any incoming signal by computing the partial correlation on each block containing 250 samples. The number of phases needed in the Correlator depends on the number of tiles which perform correlation in parallel. The shifter is a 250 bit linear-feedback shift register (LFSR). The LFSR is loaded from the PRN Code RAM memory at register number 250 after determining the code ID by demand from the user. The LSFR and PRN code RAM are clocked with the main system clock. Each bit of the LFSR is tapped out to form a PRN code with a certain code phase, as illustrated in Fig.13. When the Correlator Coder is enabled by a control signal from the user, the ACACSG first enables both the PRN Code RAM and the shifter. As soon as the shifter is filled with the first 250 bits, the ACACSG enables the Correlator Sampler to start correlating 250 code phases in parallel with the replayed recorded signal in the Correlator tiles.

6.1.1.1.3 Correlator Figure 6.7 shows a functional block diagram of the Correlator. The block is composed of three elements: the Correlator Distributor, Correlator Tile, and Correlator Sequencer. This block accepts three bit I and Q samples from the Correlator Sampler DPRAM and control signals from the Correlator Coder interface. The block outputs an 80 x 250 I and Q data array each 12 bit wide, the I and Q channels are concatenated to form 24 bits, and are sent to the Correlator Memory interface, which also provides this interface with the write address and writes a valid signal. The block also provides the DFT Block interface with a control signal (dout.page2-valid) to enable this block.

Correlator Distributor The Correlator Distributor performs a two-stage pipeline to copy the input data stream to 250 Correlator Tile circuits. The first stage produces 16

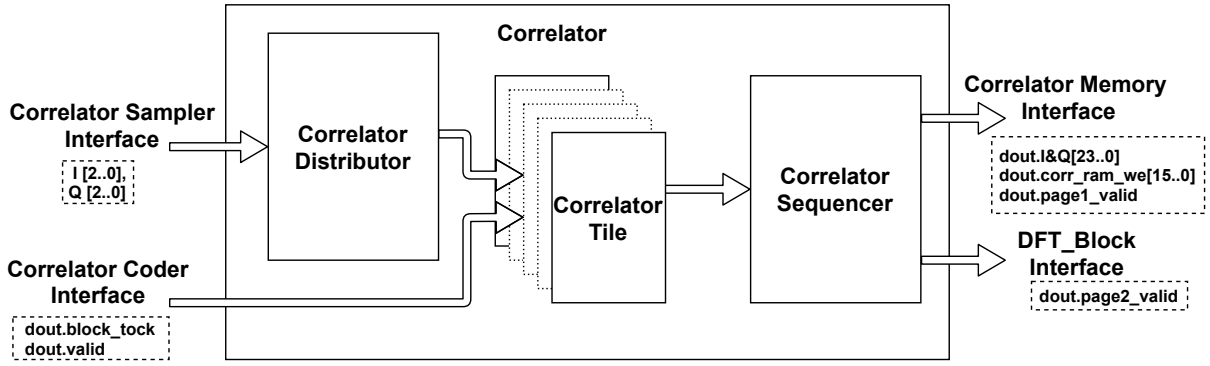


Figure 6.7: Correlator functional block diagram.

copies from the input, and the second produces 16 copies from each first-stage output.

Correlator Tile The overall functionality of the Correlator Tile circuit is shown in Figure 6.8. In total, there are 250 tiles in the Correlator. The number of tiles depends on the number of code phases that must be correlated in parallel. Each tile accepts one code phase of the 250 phases created by the Correlator Coder, one copy of the 250 copies of 3 bit I and Q created by the Correlator Distributor and control signals to clear the accumulation results after each 250 sample period. In the diagram, the first adder correlates the current sample with the current PRN code phase and samples the result into a Delta register. The second adder sums the deltas over a period of 250 samples, maintaining the correlation results in the accumulator register Sum. The correlation starts at the beginning of the

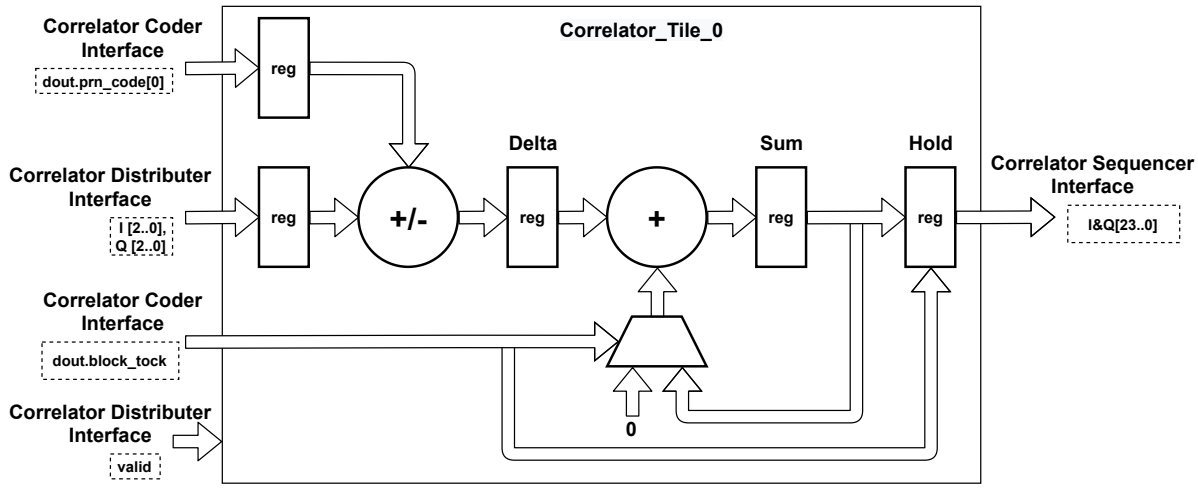


Figure 6.8: Correlator Tile functional block diagram.

first sample period, and the results are cleared at the end of 250 sample periods, where a control signal (dout.block-tock) from Correlator Coder is asserted, using the multiplexer shown. The data output from the accumulator register Sum at the end of each 250 sample period are the final results, representing the correlation result of one block, that is, 250 samples. These are written in the Hold register for later reading by the sequencer. Each tile performs a correlation on the complex I and Q channels. The 12 bit wide values in both I and Q Hold registers are concatenated to form 24 bit data, to be read by the sequencer after each clearing process by the control signal (dout.block-tick).

Correlator Sequencer The overall functionality of the Correlator Sequencer circuit is shown in Figure 6.9. The input to this circuit is a 256 data stream array in parallel, only 250 of them (from 0 to 249) are used which receives data stream from 250 tiles in the Correlator. The input to this circuit is a 256 data stream array in parallel; only 250 of them (from 0 to 249) are used which receives data stream from 250 tiles in the Correlator. The Sequencer reads the 250 concatenated I and Q correlation results in Hold registers from each of the 250 tile circuits directly after the results are written into the Hold register. This data stream is pipelined into the Correlator Memory interface. The Correlator Sequencer also provides the Correlator Memory and the DFT Block interface with read address and control signals.

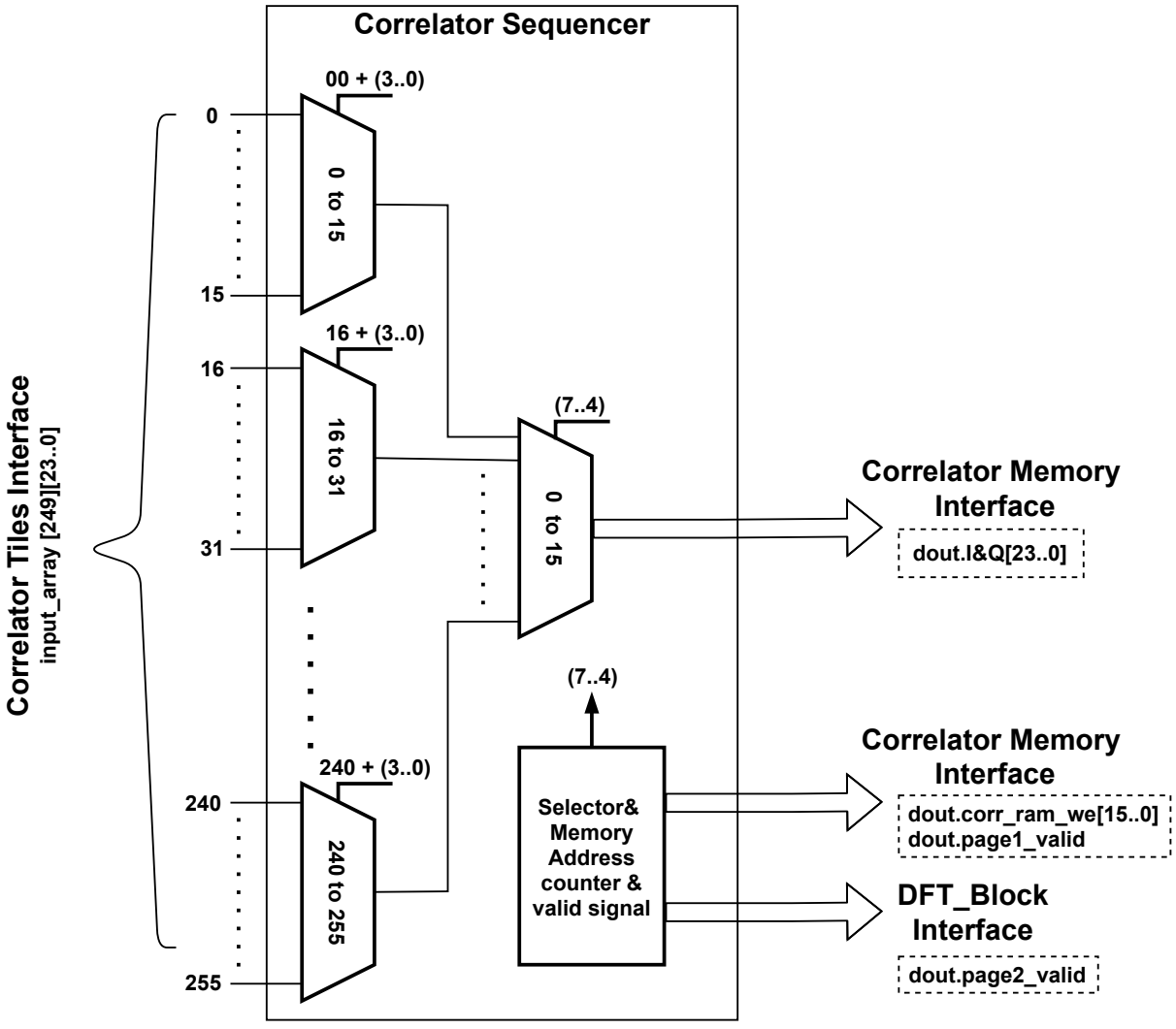


Figure 6.9: Correlator Sequencer functional block diagram.

6.1.1.2 Correlator Memory

The Correlator Memory is a 65536x24-bit DPRAM to hold the complex (I and Q) correlation result stream, writeable through port A, and readable from port B. The input of the DPRAM is through port A, which is a 12 bit I and Q, concatenated to form 24 bits, connected to the Correlator Block interface. The output of port B is connected to the DFT Block. The DPRAM is 24 bits wide, the 12 Q samples occupy 0 to 11, while the

12 I samples occupy 12 to 23 of the total 24 bits. The write address and control signals to port A are provided from a control circuit inside the Correlator Block to write the correlation results of 80 blocks to this memory, while the read address and control signals to port B are provided from a control circuit inside the DFT Block to read the correlation results from this memory. The Correlator Memory is divided into two pages, each of page holds 20000 samples. In total, the Correlator Memory holds the correlation results of two sequence samples (2 x 80 x 250), that is, 40000 samples of 24 bit wide (I and Q). When the Correlator Sequencer writes the results to the first page, the DFT Block reads the data from the second page and vice versa. This approach enables the correlation and computation of the DFT of each block without halting the correlation process.

Each page contains a 250 column by 80 row data sample array which is the result of performing correlation on 250 samples for 80 blocks of the FDBZP. In the writing process of page 1, the first 250 addresses (from 0 to 249) hold the correlation result for the first block, the second 250 addresses (from 250 to 499) hold the correlation result for the second block, and so on. In the writing process of page 2, the first 250 addresses (from 20000 to 20249) hold the correlation result for the first block, the second 250 address (from 20250 to 20499) holds the correlation result for the second block, and so on. Note that each page holds correlation results for each permutation. The maximum likelihood to find the correlation will be in the last permutation, that is, 79 permutations (first block is excluded); therefore, each page might be written 20 times.

According to the DBZP algorithm, the correlation peak can be found by applying DFT to each column which contains 80 samples in our case. Therefore, when the DFT Block reads the samples, it increments the read address by 250 each time. For example, reading out the first column on page 1 requires an increment in memory from 0, 250, 500, 750, 1000..., 19750. While reading out the second column requires an increment in memory from 1, 251, 501, 751, 1001..., 19751, and reading out the last column requires an increment in memory from 249, 499, 749, 799, 1249..., 19999. While reading out the first column on page 2 will require an increment in memory from 20000, 20250, 20500, 20750, 21000..., 39750. Reading out the second column requires an increment in memory from 20001, 20251, 20501, 20751, 21001..., 39751, and reading out the last column requires an increment in memory from 20249, 20499, 20749, 20799, 21249..., 39999.

The Correlator Memory DPRAM is initialised with a known pattern for system verification and debugging purposes during the early stage of development of the DFT Block. The pattern is synthesised data representing the partial correlation results for 80 blocks of 250 samples each belonging to Galileo E1b code for SV1 with a known code phase, multiplied with BOC(1,1) sub-carrier (at twice the frequency of the code) and modulated with a complex carrier at 8 kHz, which represents the Doppler frequency. The size of the data is 250 x 80 which gives 20000 samples of I and Q each 12 bit wide. The pattern is synthesised in MATLAB in double-precision and converted into a 3 bit fixed-point. In the test mode, the user can drive the pattern to the DFT Block to test the DFT Block without the need to record the real data. This simplifies the verification at early development stages by avoiding the use of potentially corrupted data during the recording process.

6.1.1.3 DFT Block

The Fourier transform (FT) theory states that any continuous periodic signal can be represented as the sum of a series of sinusoidal signals of different amplitudes and frequencies.

The Fourier series can be generalised to complex numbers and expressed as (Kaiser, 2010)

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j2\pi ft} \cdot dt \tag{6.1}$$

The Fourier transform allows a signal to be converted from time domain to its equivalent representation in frequency domain and conversely by using inverse Fourier transform. The inverse Fourier transform expressed as

$$x(t) = \int_{-\infty}^{\infty} X(f) \cdot e^{j2\pi ft} \cdot df \tag{6.2}$$

A periodic continuous signal is referred to as a Fourier series, whereas a non-periodic continuous signal is referred to as the Fourier transform. In real time, to obtain the frequency spectrum of a signal, a sampling process is required. The signal is discrete owing to the sampling process. The sampled version is referred to as the discrete Fourier transform (DFT) or discrete Fourier series. DFT is a sampled signal in time and is periodic. As defined in Fourier theory, to decompose a signal into the sum of different sinusoids of different frequencies and amplitudes, the signal has to be periodic. In this case, only a finite number of samples (N) were available for calculation. This problem is overcome by cascading an infinite number of blocks with the same N samples to maintain the signal periodicity properties (mathematically). The equation for obtaining the N-point DFT is as follows (Kester, 2003):

$$\begin{aligned} X(k) &= \frac{1}{N} \sum_{n=0}^{N-1} x(n) \cdot e^{-j2\pi nk/N} \\ &= \frac{1}{N} \sum_{n=0}^{N-1} x(n) \left[\cos(2\pi nk/N) - j \sin(2\pi nk/N)\right] \end{aligned} \tag{6.3}$$

Where:
X(k) is the DFT frequency output at the kth spectral point.
k ranges from 0 to N-1.
N is the number of samples in DFT data block.
x(n) is the signal sample in time domain and n ranges from 0 to N-1.
The complex DFT has real and imaginary values on both the input and output sides. In real time, a signal consists of a real part only, and the imaginary parts are set to zero to make the calculation possible. The real and imaginary parts must be calculated to obtain both the amplitude and phase of the signal in the frequency domain.

To understand the algorithm better, assume a signal with only real part (imaginary part set to zero) with 8 samples per one period i.e. N = 8 as illustrated in figure 6.10 and 6.11. The DFT output X(k) is the correlation between the input signal samples in time domain and N cosine and sine waves. The cosine and sine waves used in correlation process denoted as basis functions and the width of each frequency bin is equal to sampling frequency $(f_s)/N$ (Kester, 2003).
The equation 6.3 for the real and imaginary part becomes as follows:

$$X_{Re}(k) = \frac{1}{N} \sum_{n=0}^{N-1} x_{Re}(n) \cdot cos(2\pi nk/N) \tag{6.4}$$

$$X_{Im}(k) = \frac{-1}{N} \sum_{n=0}^{N-1} x_{Re}(n) \cdot sin(2\pi nk/N) \tag{6.5}$$

Where: $x_{Re}(n)$ is the real part of the signal.
$X_{Re}(k)$ and $X_{Im}(k)$ are the real and imaginary parts of the DFT frequency output at the kth spectral point respectively.

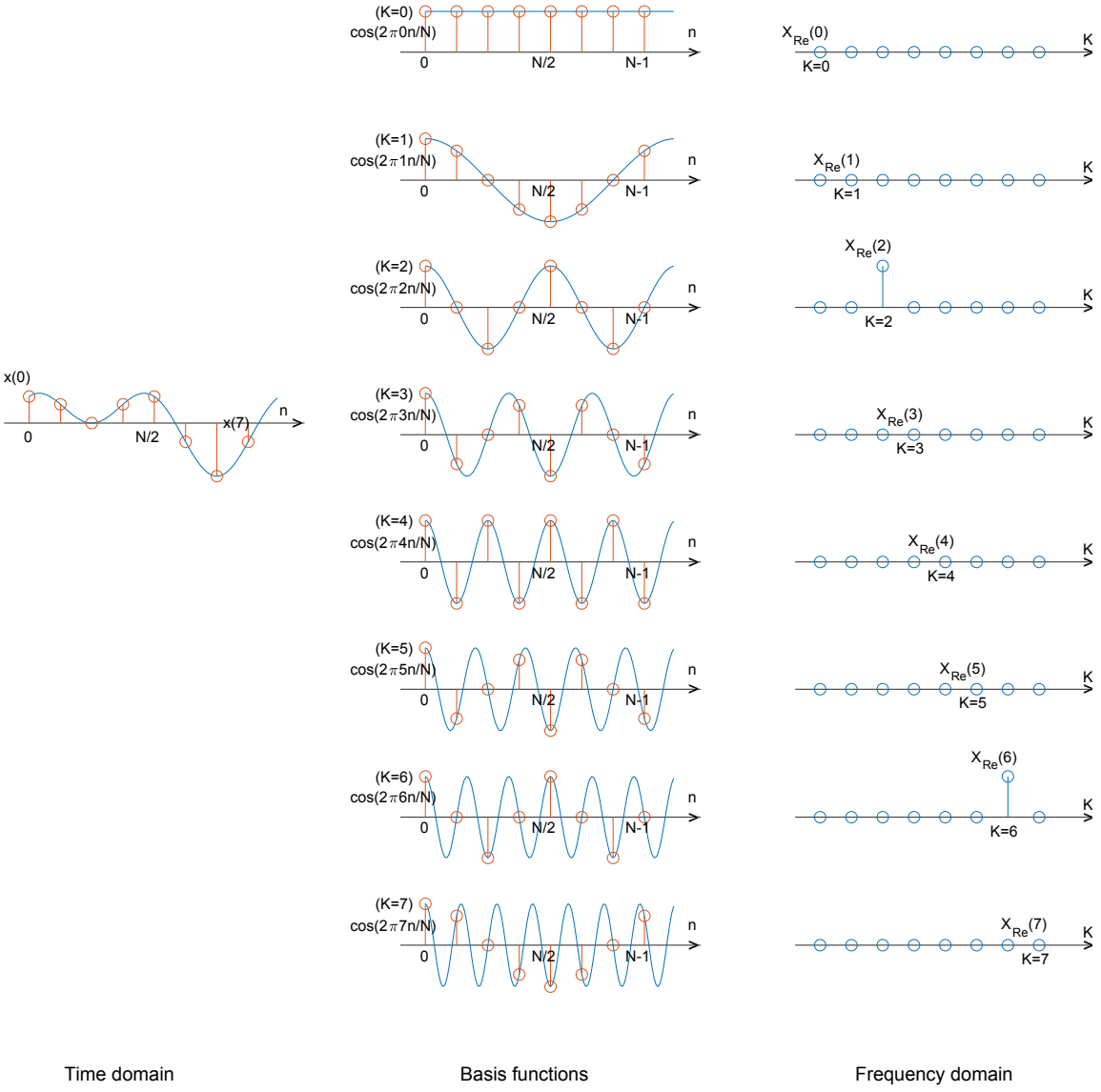


Figure 6.10: Correlation of the real signal samples with real basis functions using DFT for N=8.

The basis functions (sine and cosine waves) of both real and imaginary parts are with different intervals, depending on the number of samples, it can be obtained from equation 6.3. When k = 0, the first basis function on real part cos(0) = 1, while on the imaginary part sin(0) = 0. When k = 1 the basis functions on both real and imaginary parts are making one cycle in the interval N, when k = 2 making two cycles, when k = 3 making three cycles and when k = N the basis functions making N cycles. On the real part of the DFT output, Figure 6.10, $X_{Re}(0)$ is the sum of N input signal samples, because cos(0) = 1. This value is the average value of the input signal, it represents the DC offset value of the input signal. The second point $X_{Re}(1)$ is calculated by multiplying each of the input samples (N samples) with corresponding point on a cosine wave and summing the results. The third point $X_{Re}(2)$ is calculated by multiplying each of the input samples with

corresponding point on a cosine wave and summing the results. This process continues until all N outputs have been calculated. All N outputs must be multiplied by the scaling factor 1/N. The imaginary part of the DFT output, Figure 6.11, $X_{Im}(0)$ is zero, because

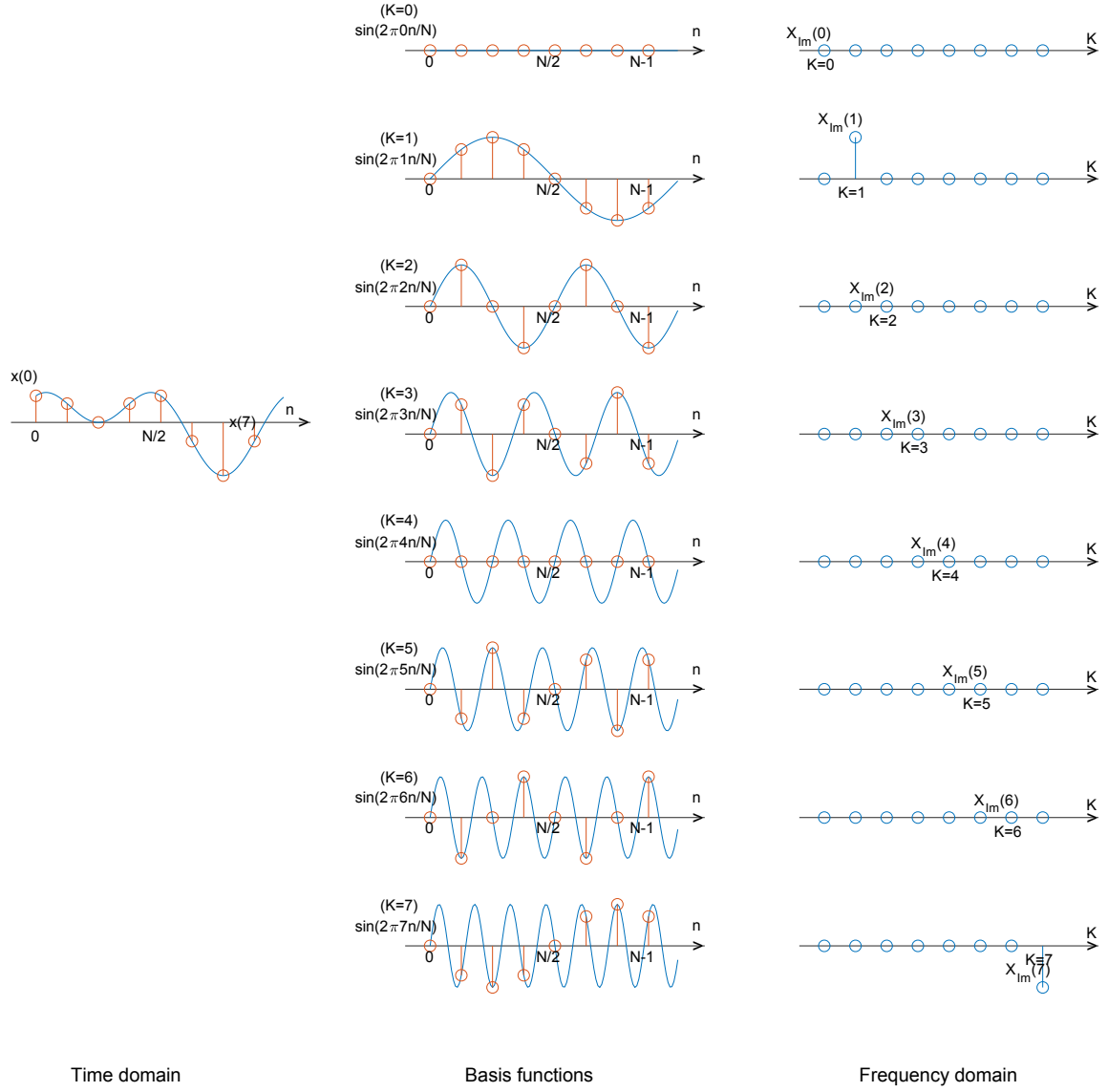


Figure 6.11: Correlation of the imaginary signal samples with imaginary basis functions using DFT for N=8.

$\sin(0) = 0$. The second point $X_{Im}(1)$ is calculated by multiplying each of the input samples (N samples) with corresponding point on a sine wave and summing the results. The third point $X_{Im}(2)$ is calculated by multiplying each of the input samples with corresponding points on a sine wave and summing the results. $X_{Im}(4)$ is zero, because $\sin(8\pi) = 0$. This process continues until all N outputs have been calculated. All N outputs must be multiplied by the scaling factor 1/N. In the Figure 6.10 and 6.11, the following can be noted:

- On the real part, the points from $X_{Re}(N/2 + 1)$ to $X_{Re}(N - 1)$ are a replica of $X_{Re}(1)$ to $X_{Re}(N/2 - 1)$, this means that the real part has even symmetry about N/2.

- On the imaginary part, the points from $X_{Im}(N/2+1)$ to $X_{Im}(N-1)$ are a negative replica of $X_{Im}(1)$ to $X_{Im}(N/2-1)$, this means that the imaginary part has odd symmetry about N/2.
- On the imaginary part, $X_{Im}(0)$ and $X_{Im}(N/2)$ are always zero.
- For a given N/2 points of a real signal DFT, the complex DFT can be generated utilising the symmetry property.

Figure 6.12 shows the functional block diagram of the DFT Block. The block performs a Fourier transform on discrete complex sets. The DFT Block is composed of four elements: DFT Baser, DFT Correlator, DFT Sequencer, and Detector. The DFT Block receives control signals from the Correlator Block interface, which triggers the block as soon as the correlation result for the first block is entirely written on page 1 of the Correlator Memory. At this stage, the Correlator Block will write to page 2, while the DFT Block reads the content on page 1 to compute the DFT on an 80 x 250 data array. The input to this circuit is an 80 x 250 I and Q data array from the Correlator Memory interface, which represents the correlation results performed on 80 blocks each of 250 samples in Correlator and saved in the Correlator Memory. To read the data array, the DFT Block provides the Correlator Memory with read address on port B, from a circuit in DFT Baser. The block outputs the acquisition results, such as codephase, Doppler frequency bins, and acquisition matrix data. The correlation results are written to registers for read purposes by the processor, which includes permutation cycle numbers which contain the correlation peak, the code-phase, Doppler bin, and the result of the auto correlation in logic mode, where logic '1' indicates detection and '0" indicates no detection.

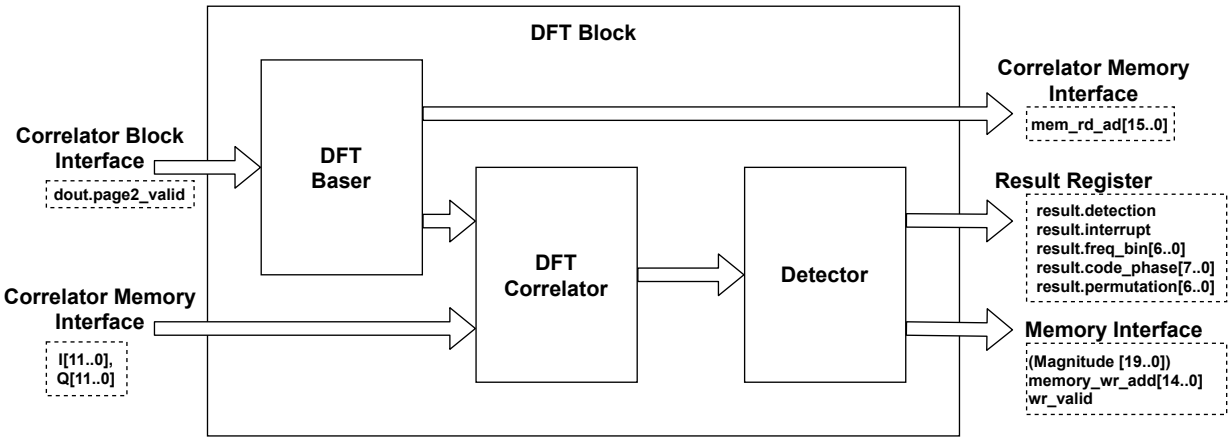


Figure 6.12: DFT Block functional block diagram.

6.1.1.3.1 DFT Baser The basis functions (sine and cosine waves) of both the real and imaginary parts are at different intervals, depending on the number of samples. The overall functionality of the DFT Baser circuit is shown in Figure 6.13. This circuit generates sine and cosine base functions for the DFT computation. The circuit receives control signals from the Correlator Block interface, which triggers the circuit as soon as the correlation result is entirely written into page 1 of the Correlator Memory, to generate sin and cos base functions required to compute 250 DFT, each containing 80 samples. This circuit also provides the Correlator Memory interface with a read address and the DFT Correlator with control signals.

To compute the DFT of 80 samples, there is a need for 80 basis functions, each consisting of 80 samples. For 3 bit quantisation, the basis function forms an 80 x (80 x 3) bit matrix. This will require 128 x 256 bit wide memory to hold the matrix, that is, 8 memory blocks of 128 x 32 bit width. To compute the DFT using a parallel approach, the incoming signal should be correlated with 80 basis functions in parallel using 80 tiles.

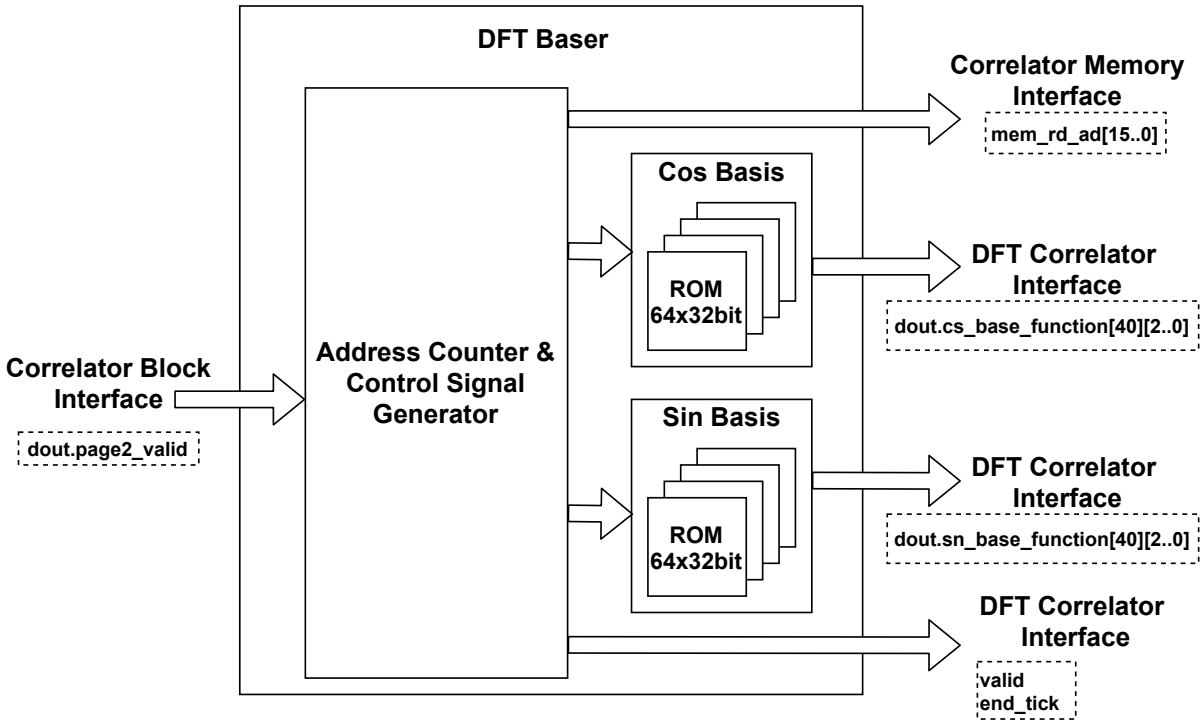


Figure 6.13: DFT Baser functional block diagram.

Owing to the periodicity and symmetry of the sine and cosine basis functions, only the first half of the samples can be used to obtain periodic functions. This is achieved by only saving the first half of the samples in the memory blocks and driving the memory read address with an up-down counter. The counter drives the samples from first to end, and then reverses the process by driving the sample from the end to the first.

While the real part of the DFT has symmetry about N/2 and the imaginary part has odd symmetry about N/2, the complex DFT can be generated by utilising the symmetry property. This will reduce the correlation computation to half, that is, only the first half (N/2) of the basis functions will be used to obtain the complex DFT. This approach reduces the basis function matrix size from 80 x 240 to 40 x 120 for 3 bit quantisation, in other words, 75% fewer memory blocks and 50% fewer correlation tiles are required. Therefore, the cos and sin base functions are saved in two memory blocks, each consisting of four 64x32 bit RAM blocks that are cascaded to form a 64 x 128 bit ROM, that is, each address is 128 bit wide. The first 41 addresses are used to hold the 41 samples of the basis functions. The basis functions are 3 bit signed quantised, bits 0 to 2 are assigned to the first bases function, bits 3 to 5 are assigned to the second basis function, and so on. The last eight bits are not used. In summary, each address holds the samples, while each 3 bits basis functions is cascaded in parallel to form 120 bits. To launch the resided basis functions in sin and cos basis memories, the ACACSG provides the memories with the read address, which will drive the first samples for 41 basis functions in parallel each 3 bit

wide which forms 120 bits in total; the second address will drive out the second samples and so on. Note that the last eight bits of each address are not used (does not hold any data).

6.1.1.3.2 DFT Correlator The DFT output X(k) is the correlation between the input signal samples in the time domain and the N cosine and sine waves. The cosine and sine waves used in the correlation process are denoted as basis functions, and the width of each frequency bin is equal to the sampling frequency $(f_s)/N$. Complex DFT can be generated by utilising the symmetry property. This will reduce the correlation computation to half, that is, only the first half (N/2) of the basis functions will be used to obtain the complex DFT. This approach reduces the number of Correlator Tiles from 80 to 41. Figure 6.14 shows a functional block diagram of the DFT Correlator. The block

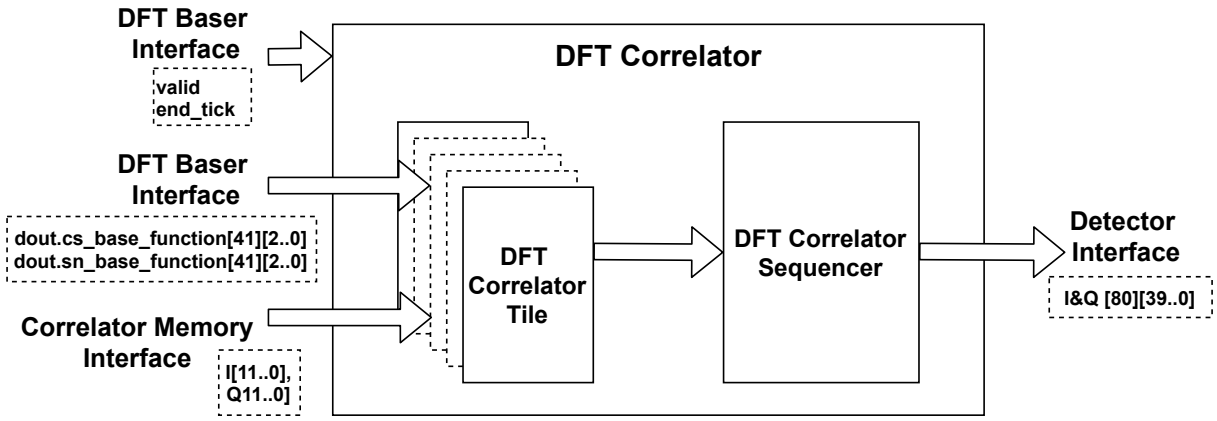


Figure 6.14: DFT Correlator functional block diagram.

performs a correlation on both the I and Q channels. The 20 bit wide values of both I and Q outputs are concatenated to form 40 bit data, to be read by the sequencer after each clearing process by the control signal (end-tick) from the DFT Baser interface. The block is composed of two main elements: 41 DFT Correlator Tile and DFT Correlator Sequencer. The input to the DFT Correlator is a 41 sin and cos basis function data stream array each 3 bit wide and control signals from DFT Baser. The DFT Correlator also receives 80 x 250 I and Q data streams from the Correlator Memory interface, each 12 bit wide. The output is 80 x 250 concatenated I and Q data arrays, each 40 bit wide. The bits from 0 to 19 hold Q data, whereas the bits from 20 to 39 hold I data.

The DFT Correlator Tile Overall circuit functionality is shown in Figure 6.15. Each tile accepts one 3 bit sin and cos of the 41 basis function and a control signal (end-tick) created by the DFT Baser. It accepts also 12 bit I and Q samples from the Correlator Memory interface. In the diagram, to avoid using multipliers and DSP blocks, the first adder correlates the current sample from Correlator Memory with the current basis function samples (last bit which represents the sign), and then holds the result into a delta register. The output from the delta register represents multiplication by one. To obtain multiplication by two, the value in the delta register is added. Adding the last result to the first produces multiplication by three. The three multiplication results are multiplexed and selected using the current 3 bit basis functions values (from -3 to 3). For example, when the basis function value is -1 or 1, the multiplexer selects the first product, that is, multiply by one, when the basis function value is -2 or 2, the multiplexer selects

the second product, that is, multiplied by two, and when the basis function value is -3 or 3, the multiplexer will select the third product, that is, multiply by three. The last adder

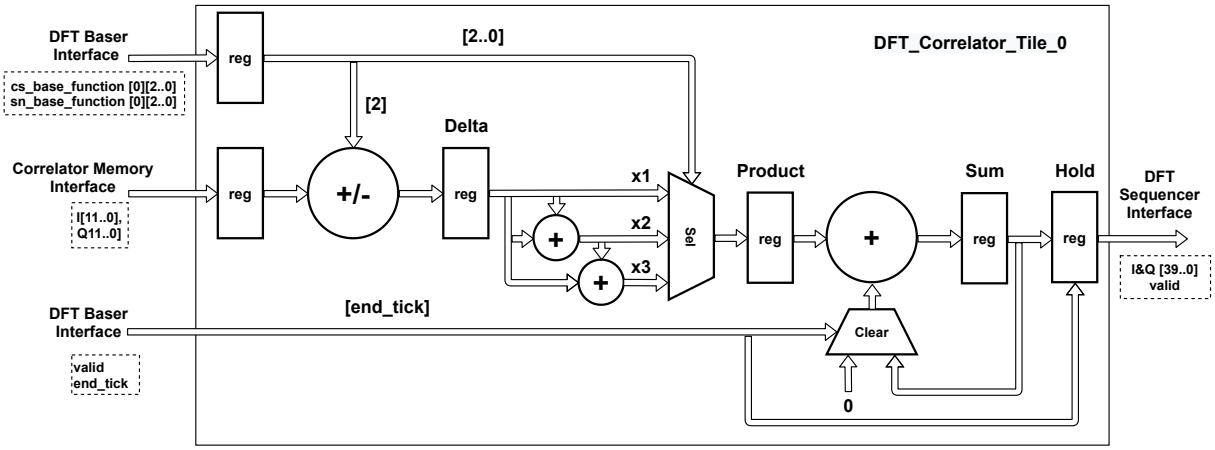


Figure 6.15: DFT Correlator Tile circuit diagram.

sums the products over a period of 80 samples, maintaining the correlation results in the accumulator register Sum.

The correlation starts at the beginning of the first sample period, and the results are cleared at the end of 80 sample periods, where a control signal from DFT Baser is asserted (end-tick), using the Clear multiplexer, as shown in Figure 6.15. The data output from the accumulator register sum at the end of each 80 sample period are the final results, representing the DFT result on one column of 80 samples. These are written in the Hold register for later reading by the sequencer. The DFT Correlator computes the correlation between the input signal samples in the time domain and 41 cosine and sine waves, and the output contains only the first DFT half. To obtain the second half (from 42 to 80), for the real part, the points from 42 to 80 will be a copy of points from 2 to 40 of the real part, while for the imaginary part, the points from 42 to 80 will be a negative copy of points from 2 to 40 of the imaginary part.

The DFT Sequencer This is similar to the Correlator Sequencer, as illustrated in Figure 6.9. It sequences 80 parallel data arrays of the complex DFT from DFT Correlator Tiles to convert them to a serial data stream. The input to this circuit is an 80 data stream array in parallel. The Sequencer reads the 80 concatenated I and Q DFT results from each of the 41 tile circuits directly after the results are written into the Hold register. This data stream is pipelined into the Detector interface.

6.1.1.3.3 Detector The Detector approximates the magnitude of the complex (I and Q) input and compares each magnitude sample against a certain threshold set by the user to detect the peak in the acquisition matrix. Approximations are often used to reduce the computational burden of obtaining the magnitudes of the I and Q vectors. Two of the most popular approximations (named after their originators) are the JPL and Robertson approximations. The JPL approximation is more accurate but has a greater computational burden.

The JPL approximation to $A = \sqrt{I^2 + Q^2}$ is defined by (Kaplan and Hegarty, 2017)

(Levitt and Morris, 1977):

$$A = \begin{cases} X + \dfrac{Y}{8} & if X \geq 3Y \\ \dfrac{7X}{8} + \dfrac{Y}{2} & if X < 3Y \end{cases} \tag{6.6}$$

Where: $X = MAX\left(|I|, |Q|\right)$ and $Y = MIN\left(|I|, |Q|\right)$
The overall functionality of the Detector circuit is shown in Figure 6.16. The input to the Detector is an 80 x 250 I and Q data stream from the DFT Correlator interface, each 20 bit wide. This data stream array is the DFT computation on 80 blocks, with 250 samples in the DBZP algorithm. The output is an 80 x 250 data array magnitude (unsigned) of the I and Q data array, each 20 bit wide. The circuit accepts also 15 bit memorywr-addr and wr-valid signals from the DFT Correlator to be used for writing the magnitude in a memory after a pipeline process. The output from this circuit is a 20 bit magnitude, 15 bit

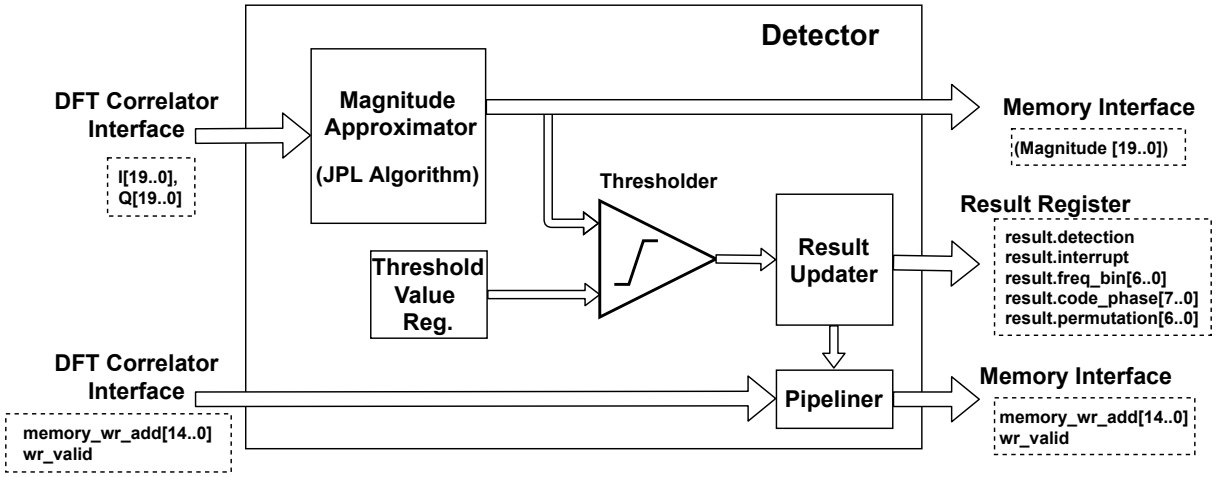


Figure 6.16: Detector circuit diagram.

memory-wr-addr and wr-valid to the memory interface. The circuit also updates the result registers with detection, interrupt, freq-bin, code-phase, and permutation information. In Figure 6.16, the magnitude approximation unit computes the magnitude using the JPL algorithm, as shown in equation 6.6. The threshold value is held in a register, there are two approaches for updating this register, in the first approach the register can be written by the user using a command from the processor or from an external interface, the advantage of this approach is the flexibility of changing the value according to the income signal strength. Second, the register value is hard coded, that is, fixed to a value; unlike the first approach, the value cannot be accessed by the user, except in the firmware update case.

The Thresholder compares each magnitude sample against the threshold value, and if there are any values above the level, the Thresholder will trigger the Result Updater unit to update the result registers with the computation outcome and hold the acquisition matrix in the dedicated memory. The signals in the result registers are

- detection: Logic '1' indicates detection and logic '0' indicates no detection.
- interrupt: This signal goes high, that is, '1' when the peak is detected, to disable the Acquisition Block and interrupt the processor.

- freq-bin: Indicates the frequency bin number.
- code-phase: Indicates code phase delay in samples.
- permutation: Indicates permutation cycle.

In case of any detection, the Detector writes the acquisition matrix into the memory and disables the Acquisition Block using the interrupt signal in the result register. Regardless of the detection time, the Result Update unit will write the whole correlation matrix into the memory first and then disables the Acquisition Block to hold the acquisition matrix with the peak in the memory for later access by the user for analysis purposes.

6.2 Performance and test results

At the early stage of the Acquisition Block's design, the block is addressed as a whole, which simplifies the understanding of the system and its interfaces. System-level requirements are developed. Once these are understood, the system (top-level entity) is then broken down into subsystems, which are further broken down into assemblies (lower-level entities), and then into components until a complete understanding of the system is achieved from top to bottom. This top-down approach is an important element for managing the development of complicated systems. By viewing the system as a whole initially and then progressively breaking the system into smaller elements, the interaction between the components can be understood more thoroughly, which assists in identifying and designing the necessary interfaces between components (internal interfaces) and between this and other systems (external interfaces).

After defining the system break-down structure (lower level entities), components are designed and built using a very high-speed integrated circuit hardware description language (VHDL) in the Notepad++ code editor. Each described component in VHDL is debugged and simulated using the ModelSim tool independently. The design is modified iteratively until the component met the desired performance. The components are then combined into assemblies and then into subsystems from which the system is obtained. At each stage of the integration, debugging and simulation are conducted to verify successful integration. The system is then simulated for the desired properties, and the design is modified until the system met the desired performance.

To verify the concept, a structural test bench containing the top-level entity (Acquisition Block) under test and a stimulus generator is developed and simulated in ModelSim. The purpose of developing the test bench is to apply the generated stimulus from the signal generator to the top-level entity design and observe the response. Then, the response is compared against the expected performance. To obtain testbench verification results, the stimulus provided to the Acquisition Block, concurrent stimulus blocks are used in the test bench to provide the necessary stimuli such as clock, reset, and enable, while the response is written into a file for analysis in MATLAB. During the verification process, the initialised pattern on the Correlator Sampler, as discussed in detail in subsection 6.1.1.1.1 is utilised.

In this thesis, the signal record-replay method is adopted. There is no need for recording, as the Correlator Sampler is initialised with a pre-known pattern. To run the Acquisition Block, the user identifies the PRN code and a control signal to trigger the system. The control signal first enables the Correlator Coder to generate 250 phases in parallel.

The code phases are validated when the shifter is filled with 250 bits to synchronise parallel outputs. This will cause a latency of 250 clock cycles in the code phase generation only at the beginning. The Correlator Coder will continue running until it is disabled by a feedback interrupt signal from the Detector. Each of the 250 samples represents one block of the partial replica codes in the DBZP algorithm, used for partial correlation with the incoming base-band samples.

The valid signal from the Correlator Coder is used to enable the Correlator Sampler and the Correlator to start performing partial correlation on the first block. While the correlation is performed utilising the convolution process i.e. point by point multiply and accumulation of 250 samples in the time domain instead of circular correlation using FFT cores, there is no need to form double blocks and zero padding the code for continuity purposes. This significantly accelerated the computation process. The process of the partial correlation will continue between the incoming recorded samples and the PRN code phases, where the first 250 samples represent the first block, the second 250 samples represent the second block, and the last 250 samples represent the last block, that is, block number 80.

The partial correlation result of a total of 80 blocks forming an 80 x 250 sample array, collected from 250 tiles in parallel, converted to serial format in the Correlator Sequencer, and saved on the first page of the Correlator Memory. After performing correlation on 80 blocks, the Correlator Sampler will permutate the 4 ms (20000 sample) recorded data by one block (250 sample), which is achieved by driving the samples in the Correlator Sampler from address 19750 instead of 0. This permutation process is repeated 79 times. In the case of detecting the peak or permutate 79 times without detecting the peak, the Detector interrupts the Correlator Coder and disables the entire system. This means that the correlation and permutation processes will not stop until the peak is detected or the recorded data are permutated 79 times which indicates no detection.

After performing the correlation on the first block and writing the result on the first page, the Correlator Block interrupts the DFT Block with (page2-valid) to read the correlation result from page one and write the correlation result of the second block into page two. This ping-pong process is conducted until all 80 blocks have been processed. The DFT Block reads the data from the previously written page and performs DFT by correlating the basis functions with the I and Q samples in the DFT Correlator. The DFT output is passed through the Detector to approximate the magnitude of the complex (I and Q) samples using the JPL algorithm and compare each of these samples against a certain threshold to detect any potential peak.

Finally, the computed magnitude of each acquisition matrix (each sequence) is written to an external or internal memory iteratively; only the block which contains the peak will be maintained for analysis in the processor. During the simulation, this block is written to a response file for comparison with the expected performance in MATLAB.

6.2.1 Simulation results and data analysis

At the system level, the Acquisition Block is simulated using the initialised pre-known pattern on the DPRAM, as discussed in detail in sub-section 6.1.1.1.1.During the simulation, the system clock is considered to be at 100 MHz. Figure 6.17 shows the simulation waveform of the Acquisition Block in ModelSim. To investigate the correlation peak, there is need to process 80 x 250=20000 samples for 80 permutations. In this case, the correlation result is within the 0 permutation because the code phase is shifted by 100 samples,

while the block contains 250 samples. The processing required 40600 clock cycles, that is, 406 μs, when the system clock is 100 MHz. As illustrated in Figure 6.2, the system

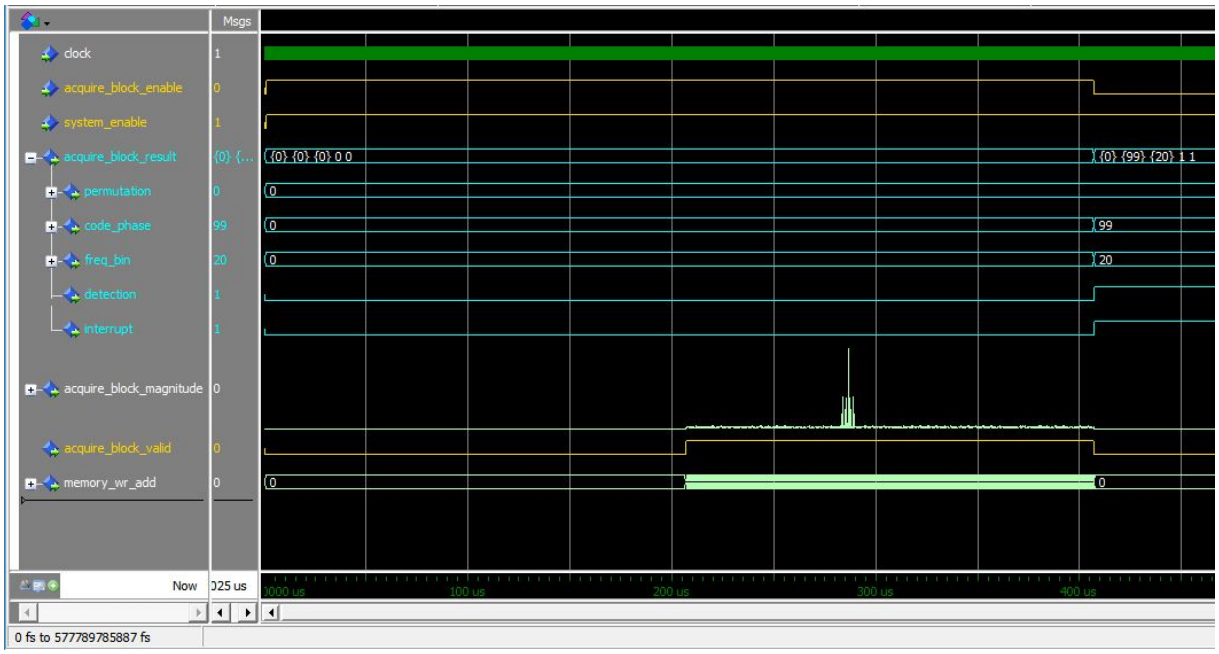


Figure 6.17: Acquisition Block simulation waveform in the ModelSim using the initialised pattern in the Correlator Sampler with a peak at code phase of 100 samples and frequency bin 21, detected in permutation cycle 0.

is enabled using two signals pushed through an AND gate. The first signal is provided by the user, and the second is an inverse feedback signal interrupt. The enabling signal first triggers the Correlator Coder to generate 250 code phases in parallel by repeatedly driving the 4 ms sequence into the LFSR shifter. The Correlator Coder validates these 250 code phases after 250 clock cycles to synchronise them, leading to a 250 clock cycle latency.

The code phase valid signal is used to enable both Correlator Sampler and the Correlator to perform correlation between the recorded complex (I and Q) signal and the 250 code phases in parallel in the Correlator Tiles, the result of each 250 I and Q samples is accumulated in the Tiles then is held in the Hold register to be read by the sequencer, the accumulation process will lead to 250 clock cycle latency. The result of the investigated 20000 samples is written on the first page of the Correlator Memory. The DFT Block will not fetch the data on the first page until the end of the writing process, which will cause a latency of 20000 clock cycles. The DFT Block then performs DFT on the 80 x 250 data grid by correlating the incoming samples with the basis functions, the result of each 80 samples is accumulated in the DFT Correlator Tiles then is held in the Hold register to be read by the DFT Sequencer, the accumulation process will lead to 80 clock cycle latency. Finally, the Detector receives the samples from the DFT block to approximate the magnitude of the I and Q samples using the JPL algorithm and determines whether a peak above a certain threshold is present, then maintains the result in the memory, updates the result registers, and interrupts the user (processor).

During data processing, there is also a latency of 20 clock cycles, owing to the pipelining of the signals for synchronisation purposes. The total latency of the system is 20600 clock cycles. It is important to point out that this latency will be the same regardless

of the permutation cycle, that is, when the peak is detected in the first permutation, the latency will be the same for detection in permutation 79. Table 6.1, summarises the latency in each process within the Acquisition Block.

Table 6.1: Latency in Acquisition Block at 100 MHz system clock.

Process	**Latency (clk)**	**Latency (μs)**
Correlator Coder (shifter)	250	2.5
Correlator Tile (accumulator)	250	2.5
Correlator Memory(writing page1)	20000	200
DFT Correlator Tile (accumulator)	80	0.8
Pipelining (in whole system)	20	0.2
Performing DFT on 1 permutation (80 points x 250 row)x1	1x20.000 =20.000	1x200 =200
Performing DFT on 80 permutation (80 points x 250 row)x80	80x20.000 =1.600.000	80x200 =16.000
Total latency for 1 permutation	**40.600**	2.5+2.5+200+0.8+0.2=**406**
Total latency for 80 permutation	**1.620.600**	**16.206**

In Figure 6.17, the permutation indicates 0, code-phase 99, frequency-bin 20, detection '1' and the interrupt is high '1, the bin size is 250 Hz, which reflects the information in the pattern, initialised by default in the Correlator Memory. The peak can be seen in the magnitude waveform, and the 20000 data points are written to a memory for subsequent processing using memory-wr-add and valid signals. The approximated magnitude of the correlation matrix is written to a response file for comparison with the expected performance in MATLAB. Figure 6.18 shows a three-dimensional plot of the normalised magnitude against the code-phase and frequency bins. The peak has two side mini peaks, which indicate the BOC(1,1) modulation scheme, code phase at 100 samples, and frequency bin 21 which indicates 5 kHz Doppler shift.

Note: The code-phase and frequency bin number in the ModelSim indicates to 99 and 20 respectively, while im MATLAB indicates to 100 and 21. This is due to the fact that ModelSim implements zero based indexing, while MATLAB implements one based indexing. Secondly, the first Doppler bin contains the DC offset value i.e. zero frequency, therefore the bin number 21 in MATLAB indicates (20 x 250) 5 KHz.

To investigate the Acquisition Block performance over a longer time, the Correlator Sampler is initialised with a second pattern of the data. The second pattern is a 3 bit (signed) synthesised Galileo BOC(1,1) with E1b code for SV1. The code phase is shifted by 150 samples in the last block, that is, a total code phase of 19900 and 8 kHz Doppler shift frequency. The length of the pattern is 4 ms, sampled at 5 MHz, results in 20000 samples. during the simulation, the system clock is considered to be at 100 MHz.
Figure 6.19 shows the simulation waveform of the Acquisition Block in ModelSim utilising the second initialised pattern in the Correlator Memory.

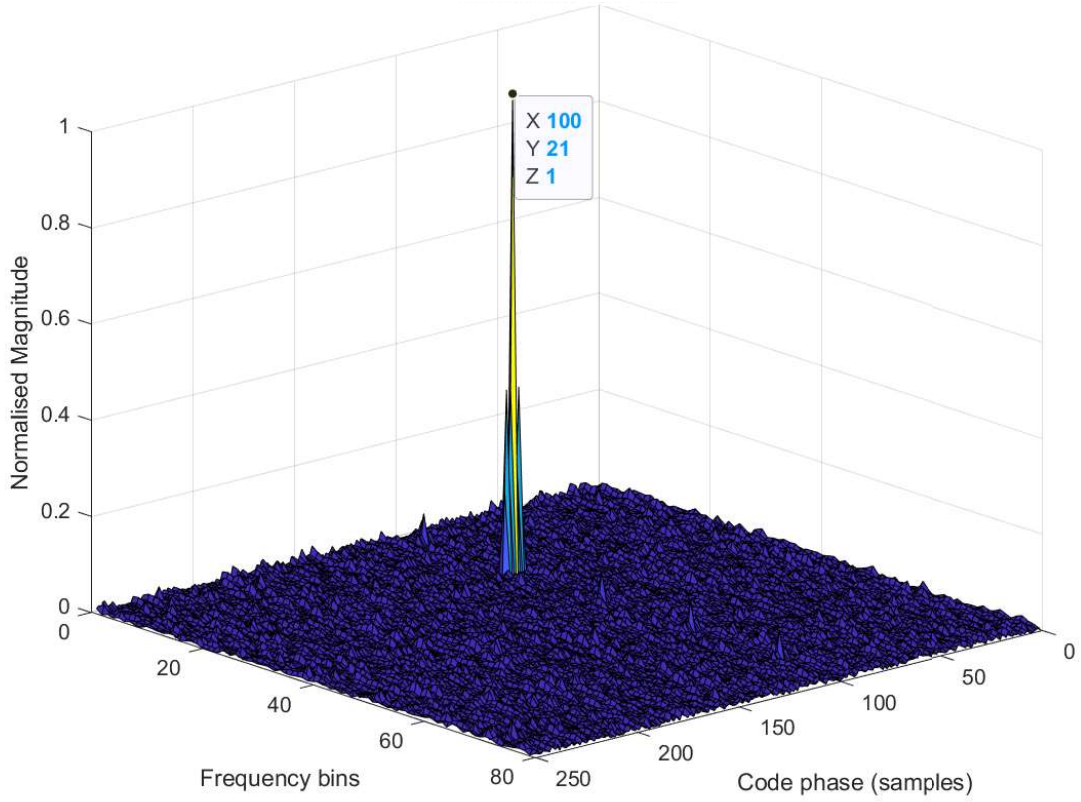


Figure 6.18: Galileo SV1, E1b acquisition matrix with a peak at code phase of 100 samples and frequency bin 21, detected in the permutation cycle 0, simulated in ModelSim and plotted in MATLAB.

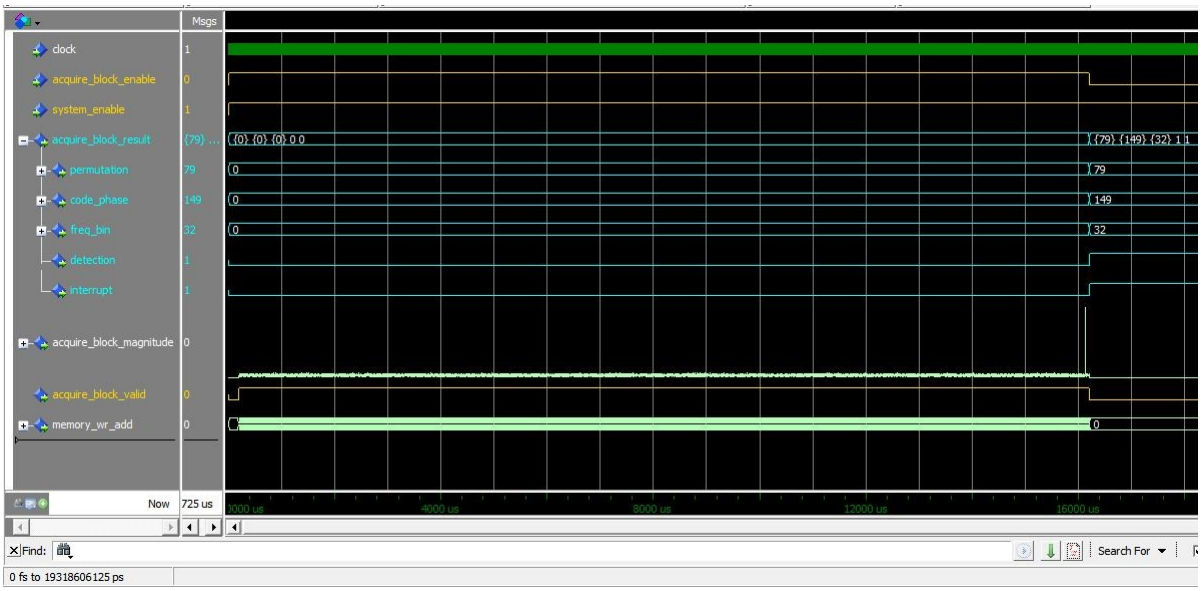


Figure 6.19: Acquisition Block simulation waveform in the ModelSim using the second initialised pattern in the Correlator Sampler with a peak at code phase of 149 samples and frequency bin 32, detected in permutation cycle 79.

In the figure, under the result waveform, the permutation indicates 79, code-phase 149, frequency-bin 32, detection '1' and the interrupt is high 1. The result reflects the

information in the second pattern, initialised in the Correlator Memory. The peak can be seen in the magnitude waveform, and the 20000 x 80 block data points are re-written to a memory repeatedly (20000 samples each time), using memory-wr-add and valid signals. Only the set with detection is maintained for the subsequent processing. Note that the processing time of 80 sequences (permutation cycles 0 to 79) is 16,206 ms, including a latency of 206 μs, assuming the system clock is at 100 MHz. This indicates that the Acquisition Block detection probability time is between 0.406 and 16,206 ms for each satellite search. The choice of the system clock depends on the FPGA speed performance. In the high-end chips, the chip can be clocked at 200 MHz, which can reduce the detection probability time between 0.203 and 8,103 ms for each satellite search.

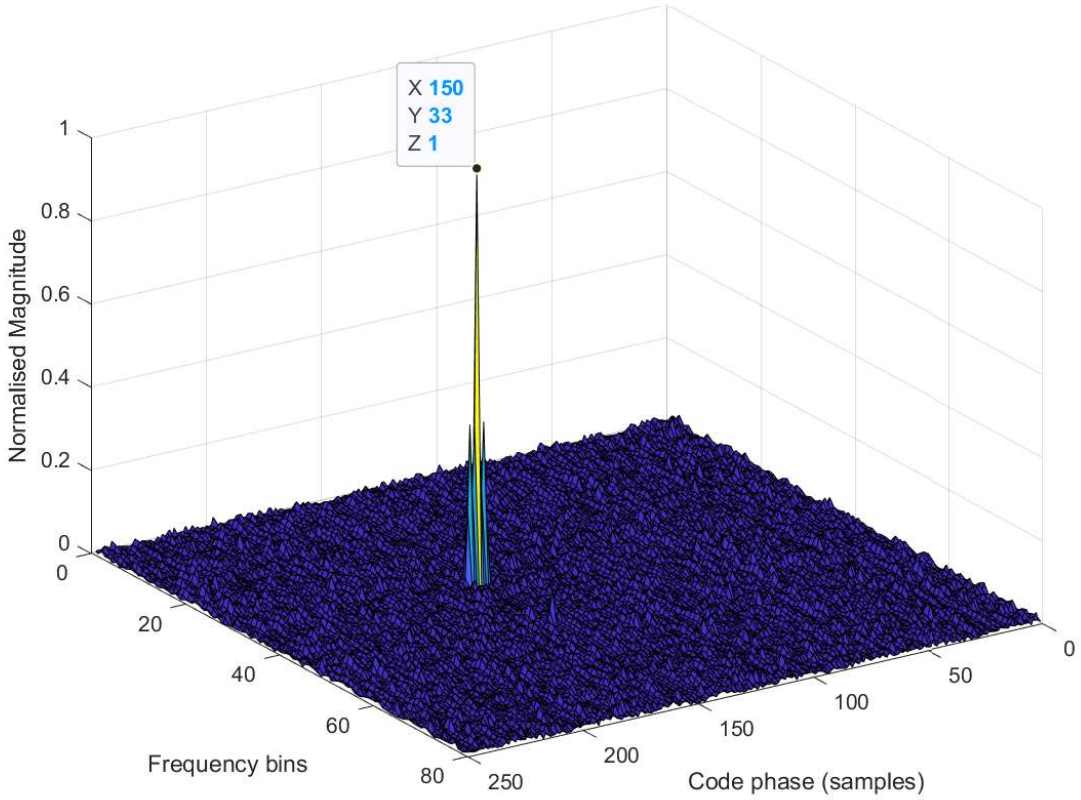


Figure 6.20: Galileo SV1, E1b acquisition matrix with a peak at code phase of 150 samples and frequency bin 33, detected in the last block (80), simulated in ModelSim and plotted in MATLAB.

The approximated magnitude of the correlation matrix exploiting the second pattern in the Correlator Sampler is written into a response file for comparison with the expected performance in MATLAB. Figure 6.20 shows a 3 dimensional plot of the normalised magnitude against the code-phase and frequency bins. Again, the main peak has two mini peaks on both sides, which indicates the BOC(1,1) modulation scheme, code phase at 150 samples, and frequency bin 33 which indicates an 8 kHz Doppler shift. It is important to clarify that the first bin is the zero frequency bin (DC); therefore, the Doppler frequency is calculated as (250 x 32) which produces 8 KHz.

To investigate the cross-correlation performance in the Acquisition Block, the Correlator Sampler is initialised with a third pattern of the data. The third pattern is a synthesised Galileo BOC(1,1) with the E1b code for SV2. The code phase is shifted by 100 samples in the first block and 8 kHz Doppler shift frequency. The length of the pat-

tern is 4 ms, sampled at 5 MHz, results in 20000 samples. during the simulation, the system clock is considered at 100 MHz.

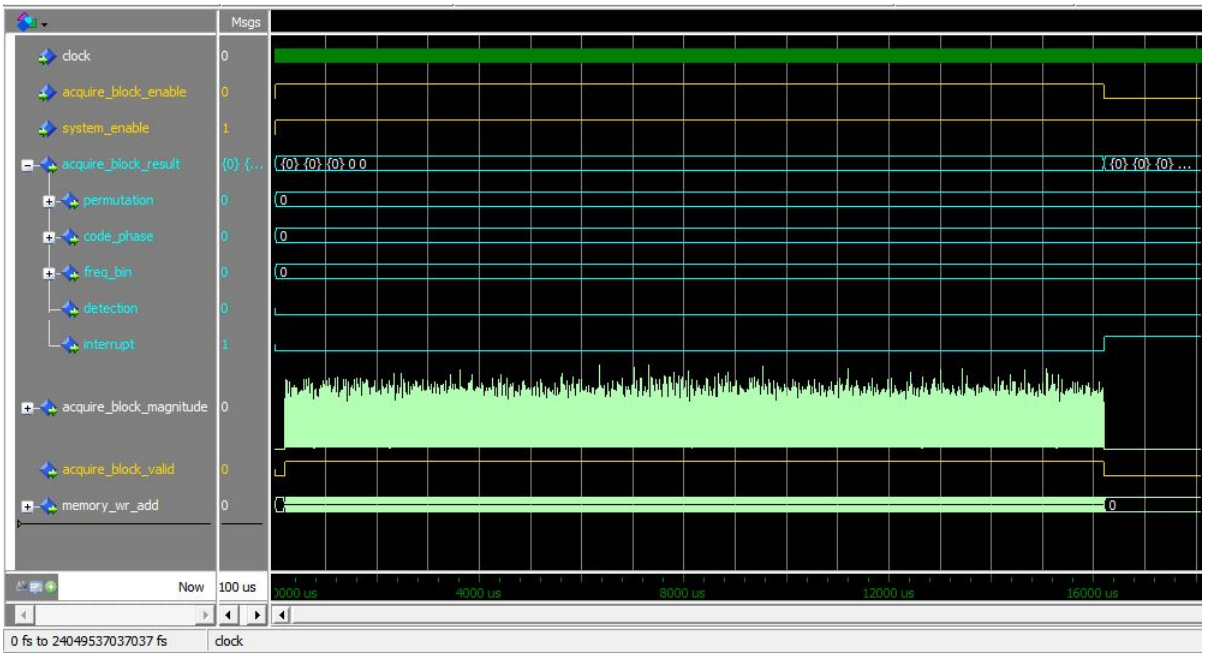


Figure 6.21: Acquisition Block simulation waveform in the ModelSim using the third initialised pattern in the Correlator Sampler.

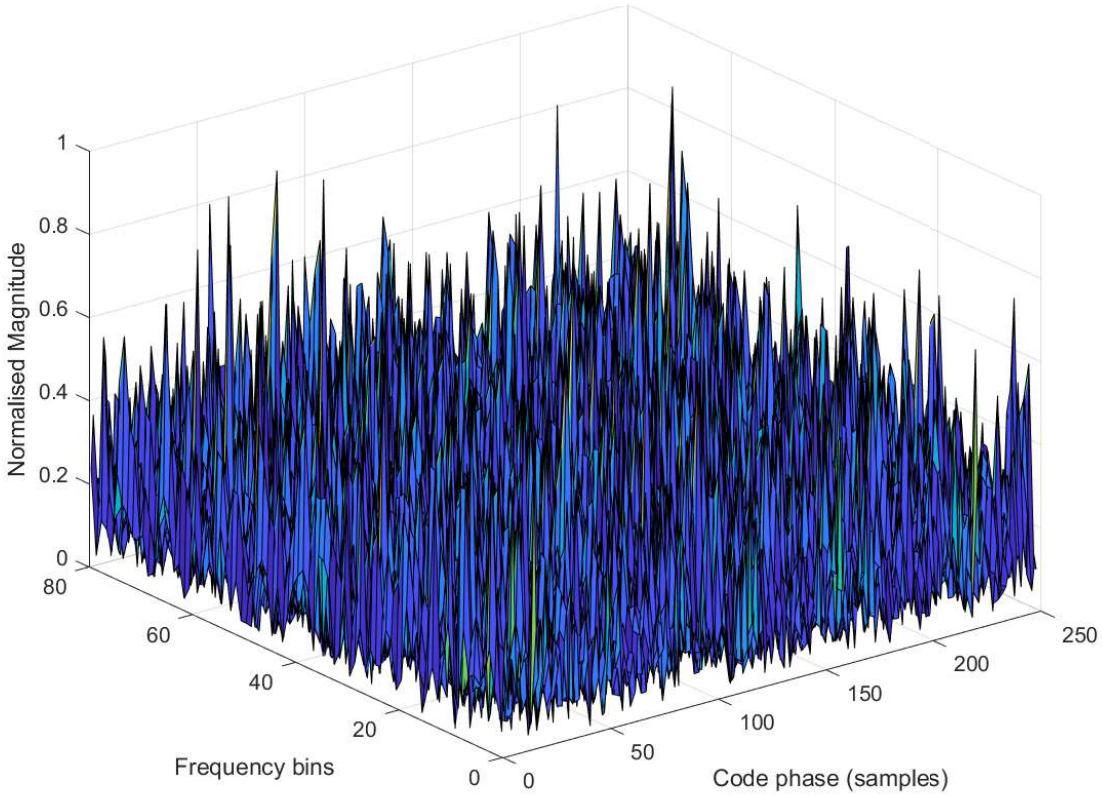


Figure 6.22: Galileo SV1, E1b acquisition matrix with no detected peak, simulated in ModelSim and plotted in MATLAB.

Figure 6.21 shows the simulation waveform of the Acquisition Block in ModelSim utilising the third initialised pattern in the Correlator Memory. In the figure, under the result waveform, the permutation indicates 0', code-phase 0', frequency-bin 0', detection '0', and the interrupt is high '1' at the end of the computation of the FDBZP algorithm. The result reflects the information in the third pattern, initialised in the Correlator Memory. No peak can be seen in the magnitude waveform, and the output is pure noise as a result of the cross-correlation between the SV1 and SV2 BOC(1,1) sequences. The 20000 x 80 block data points are re-written to a memory repeatedly (20000 samples each time) using memory-wr-add and valid signals. Note that the processing time of 80 sequences (permutation cycles 0 to 79) is 16,206 ms, including a latency of 206 μs, assuming the system clock is at 100 MHz. The interrupt signal continues to be high to indicate the end of the computation regardless of the detection result to disable the Acquisition Block and prompt the user to search for other potentially available satellites in the user scope.
Figure 6.22 shows the 3 dimensional plot of the normalised magnitude against the code-phase and the frequency bins in the last 20000 data samples, exploiting the third initialised pattern in the Correlator Sampler. It can be observed that the content of the plot is pure noise.

6.2.1.1 Bit sign transition effect

Galileo E1 OS deploys navigation and secondary code signals with equal spreading code periods, resulting in a potential bit sign transition in each primary code period of the received signal segments. To investigate the Acquisition Block performance under bit sign transition, the Correlator Sampler is initialised with a fourth pattern of the data. This pattern is a synthesised Galileo BOC(1,1) with E1b code for SV1. The code phase is shifted by 10200 samples, 6 kHz Doppler shift frequency and with a bit transition at 50% of the sampled signal. The length of the pattern is 4 ms, sampled at 5 MHz, results in 20000 samples.

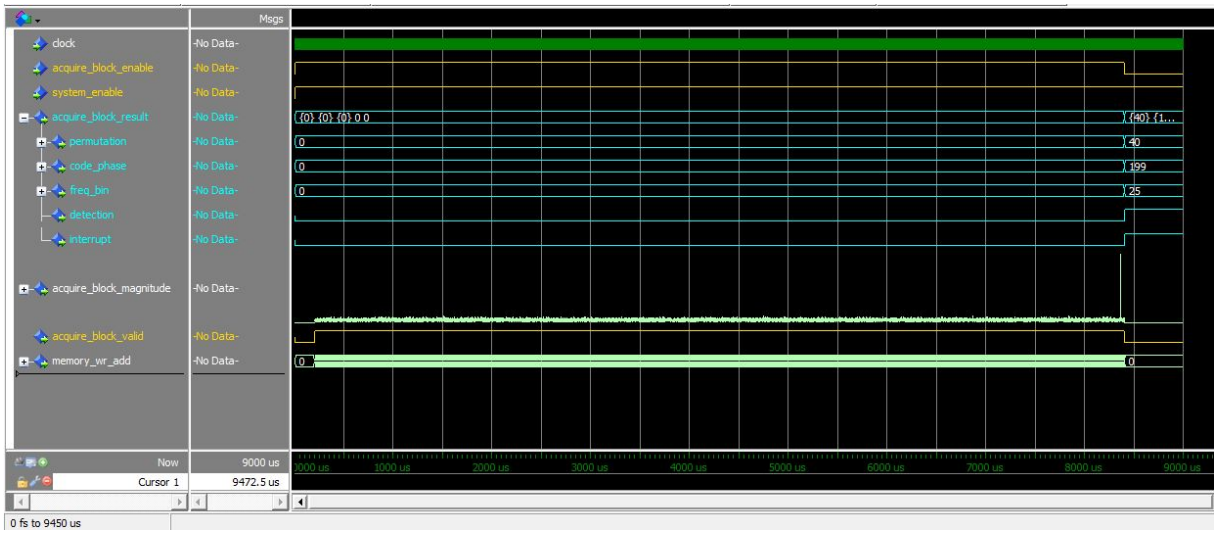


Figure 6.23: Acquisition Block simulation waveform in the ModelSim using the fourth initialised pattern in the Correlator Sampler with a peak at code phase of 199 samples and frequency bin 25, detected in permutation cycle 40.

Figure 6.23 shows the simulation waveform of the Acquisition Block in ModelSim utilising the fourth initialised pattern in the Correlator Memory. In the figure, under

the result waveform, the permutation indicates 40, code-phase 199, frequency-bin 25, detection '1', and the interrupt is high '1' at the end of the computation of the FDBZP algorithm. The code-phase shift is in the block number 41, which indicates (40 x 250 + 200) 10200 samples. The processing time of 41 sequences (permutation cycles 0 to 40) is 8,406 ms, including a latency of 206 μs, assuming the system clock is at 100 MHz. The Doppler frequency shift is (25 x 250) 6250 Hz. The Doppler frequency shift result does not reflect the information in the fourth pattern, initialised in the Correlator Memory. To investigate the error in the Doppler frequency shift estimation, the approximated magnitude of the correlation matrix exploiting the fourth pattern in the Correlator Sampler is written into a response file for comparison with the expected performance in MATLAB.

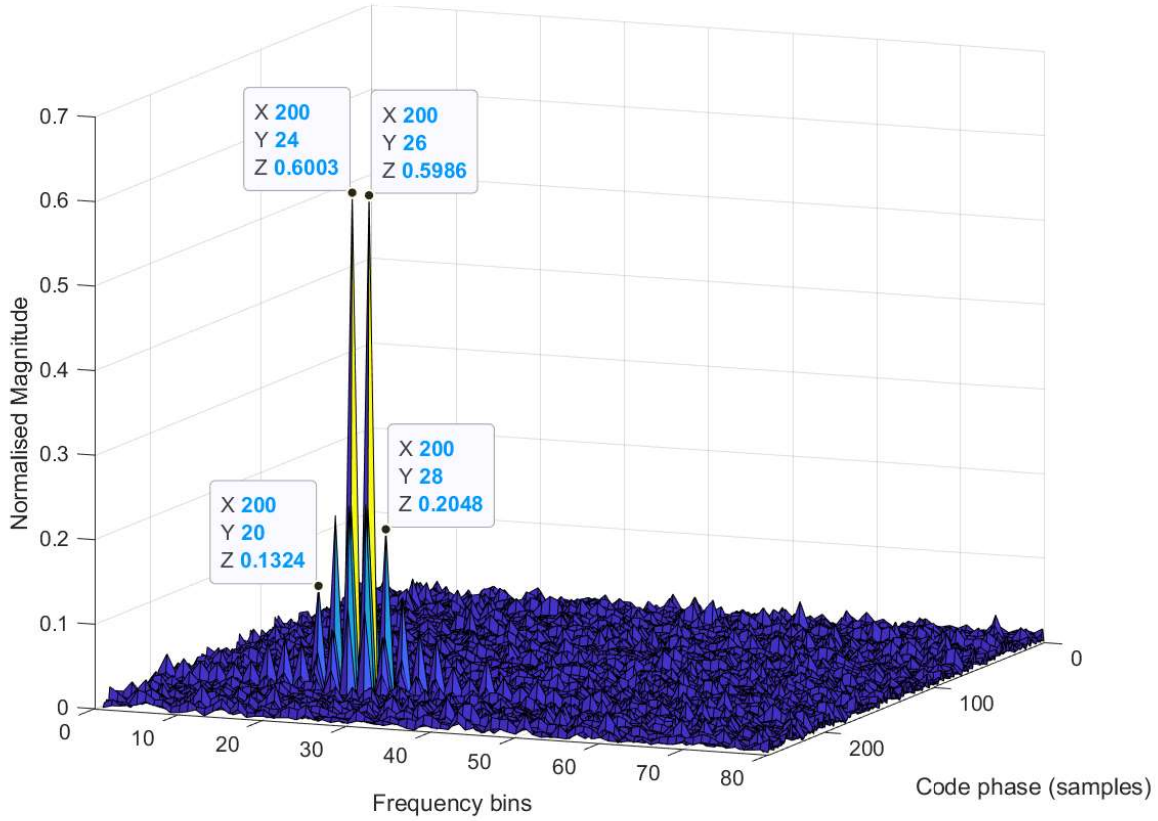


Figure 6.24: Galileo SV1, E1b acquisition matrix with bit sign transitionat approximatly at 50% of the 4 ms long sampled signal (fourth pattern), simulated in ModelSim and plotted in MATLAB.

Figure 6.24 shows the 3 dimensional plot of the normalised magnitude against the code-phase and frequency bins. It is important to point out that the magnitude is normalised using the maximum value in the acquisition matrix when there is no bit sign transition available. It can be seen in the figure that the bit sign transition splits the main peak into two adjacent peaks, approximately each with 60% of the main peak power. Note that the Doppler bin of the upper peak is shifted by +1 bin and the lower peak is shifted by -1 bin while the code-phase of both peaks indicates the true value (10200 samples).
To overcome the bit sign transitions issue, (Lin and Tsui, 2003b) proposed using two consecutive data bit periods to ensure that there is at least one accumulation interval that do not contain data bit transitions. $2T_C$ ms of incoming signal is used. Instead of circularly permuting the code blocks to investigate all possible code delays, the incoming signal is shifted by one block in a linear way. As a result of that, when the coherent

integration time is one code period, the correlation interval is free of data (and secondary code) bit sign transition. The permutation and linear shifting method require the same processing period in FPGA, the only difference is that the linear shift method requires doubling the sampled signal period to $2T_C$ ms. As a result the Correlator Sampler size has to be big enough to contain the sampled signal when the record-reply method is utilised. To accomodate $2T_C$ (8 ms) data sampled at 5 KHz on Acquisition Block, the memory size of the Correlator Sampler is doubled. To verify the system under linear shift method using 8 ms data with bit sign transition, the modified Correlator Sampler is initialised with a fifth pattern. The fifth pattern is a synthesised Galileo BOC(1,1) with E1b code for SV1. The code phase is shifted by 10200 samples (200 samples in block 41), 6 kHz Doppler shift frequency and with a bit transition at 50% of the sampled signal. The length of the pattern is 8 ms, sampled at 5 MHz, resulting in 40000 samples. During the simulation, the system clock is considered to be at 100 MHz.

Figure 6.25 shows the simulation waveform of the Acquisition Block in ModelSim utilising the fifth initialised pattern in the Correlator Memory. In the figure, under the result waveform, the permutation indicates 40, code-phase 199, frequency-bin 24, detection '1', and the interrupt is high '1' at the end of the computation of the FDBZP algorithm. The code-phase shift is in the block number 41, which indicates (40 x 250 + 200) 10200 samples and the Doppler frequency shift is (24 x 250) 6 KHz. The processing time of 41 sequences (permutation cycles 0 to 40) is 8,406 ms, including a latency of 206 μs, assuming the system clock is at 100 MHz. Figure 6.26 shows the 3 dimensional plot of

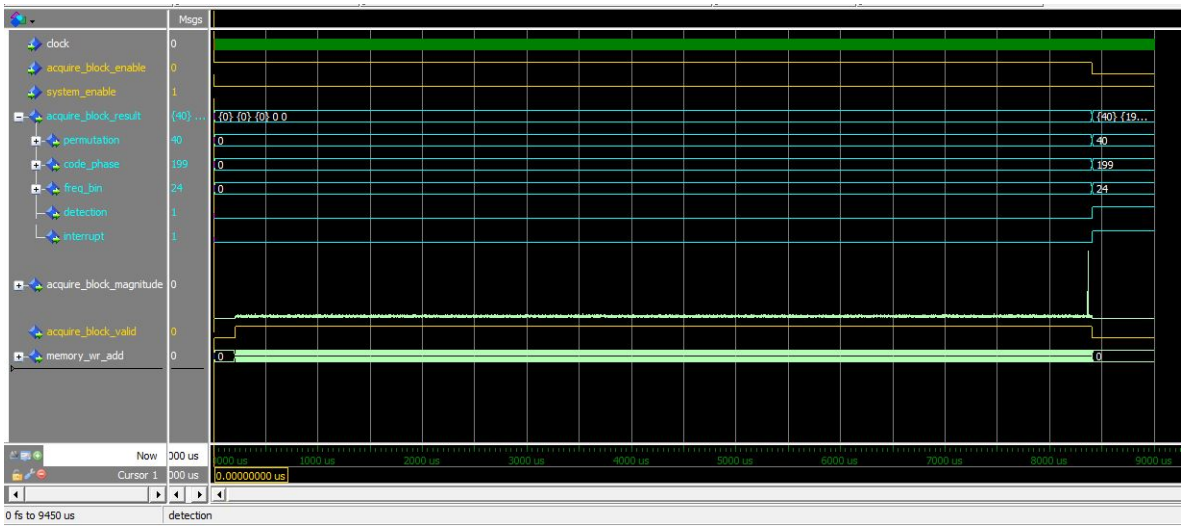


Figure 6.25: Acquisition Block simulation waveform in the ModelSim using the fifth initialised pattern in the Correlator Sampler with a peak at code phase of 199 samples and frequency bin 24, detected in permutation cycle 40.

the normalised magnitude against the code-phase and frequency bins. It can be seen in figure that the bit sign transition does not split the main peak into two adjacent peaks when the linear shift method is implemented.

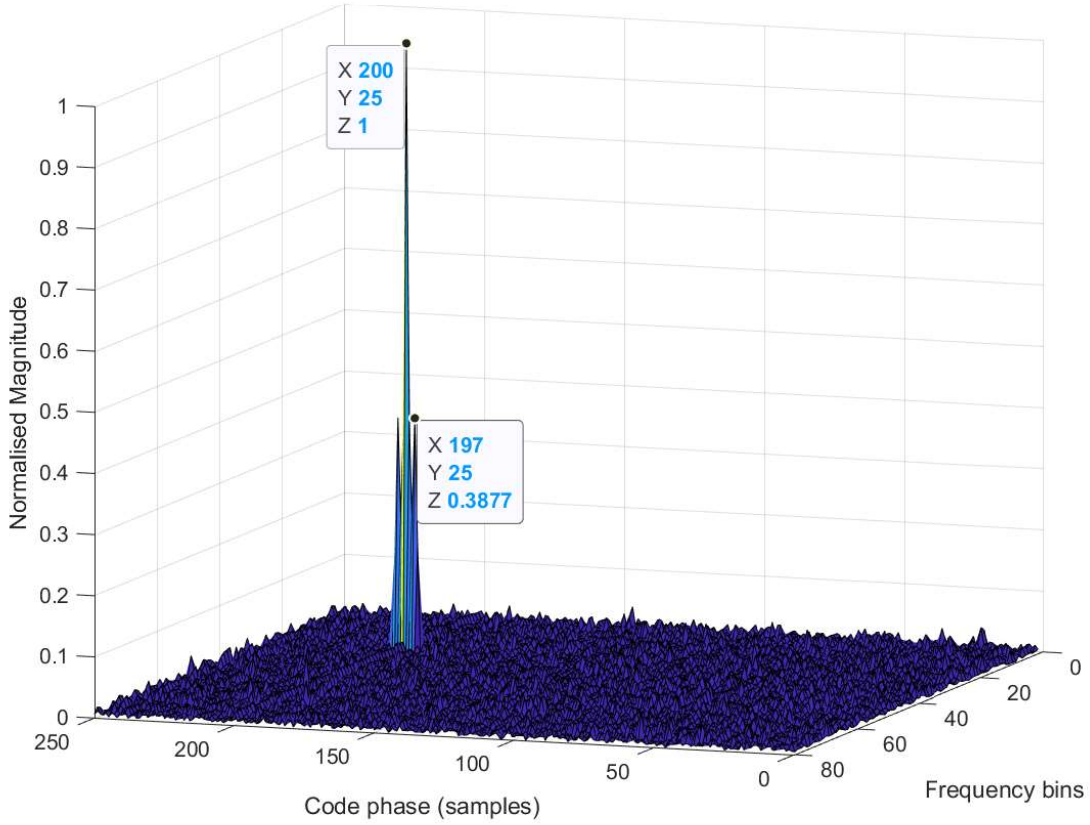
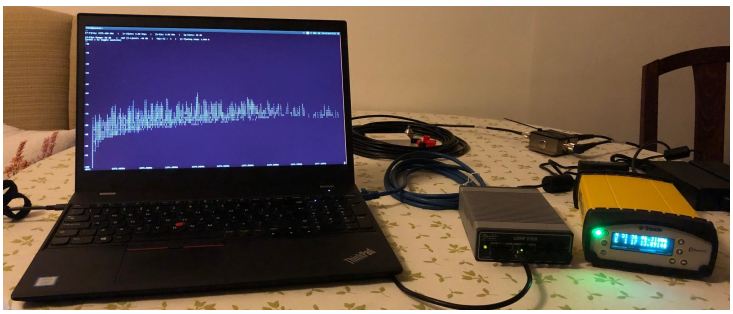


Figure 6.26: Galileo SV1, E1b acquisition matrix with bit sign transitionat 50% of the 8 ms long sampled signal (fifth pattern), simulated in ModelSim and plotted in MATLAB.

6.2.2 Simulation results using real samples

To validate the designed system, the system is simulated by using recorded real data. The recording is performed on 25^{th} of November 2021 in the Harwich city around 21:30, by using USRP E320 software defined radio (SDR), at centre frequency of 1575.42 MHz, downconverted to the baseband with 3 bit quantisation and sampled at 5 MHz for a duration of 20 ms. A multi-band active GNSS antenna (Trimble GA830) is utilized; A Trimble SPS361 GNSS receiver has been used to supply power to the active antenna and monitor received signals; A GPS Networking HIALDCBS1x2-TNC high-isolated amplified splitter used for the active antenna as the USRP does not provide DC voltage at the RF ports; A laptop is used to control the USRP via Ethernet/SSH; RF connections are made using RG58-type coaxial cables. Figure 6.27 shows the equipment setup during the recording.

Figure 6.28, shows the sky plot of the Galileo constellation from Trimble GNSS planing online, during the record time. The recording is with non-clear view to the sky. The satellites in the north and eastern sky are almost blocked by the neighbour's. The view to the south and south-west is relatively clear, therefore, only satellites 30, 27 and 15 from Galileo constellation are detected. The recorded data is complex samples with each part expressed as type short (2 Byte numbers), so 4 Bytes per IQ sample. The file contained 20 ms of signal and is therefore 400000 Bytes long (5e6 x 20e-3 x 4). To investigate the available satellites in the recorded samples, the obtained binary file from the USRP is converted to 3 bit signed integer format and written into a text file. An acquisition process has been performed on the first and second 4 ms data samples individually in



(a) Laptop, USRP E320 and Trimble SPS361 GNSS receiver

(b) Trimble GA830 active GNSS antenna.

Figure 6.27: Equipment setup for recording real data from Galileo constellation.

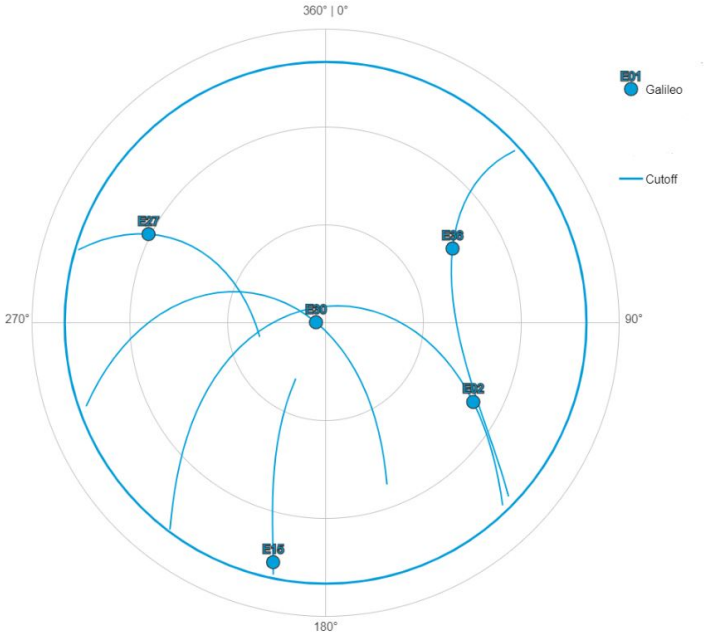
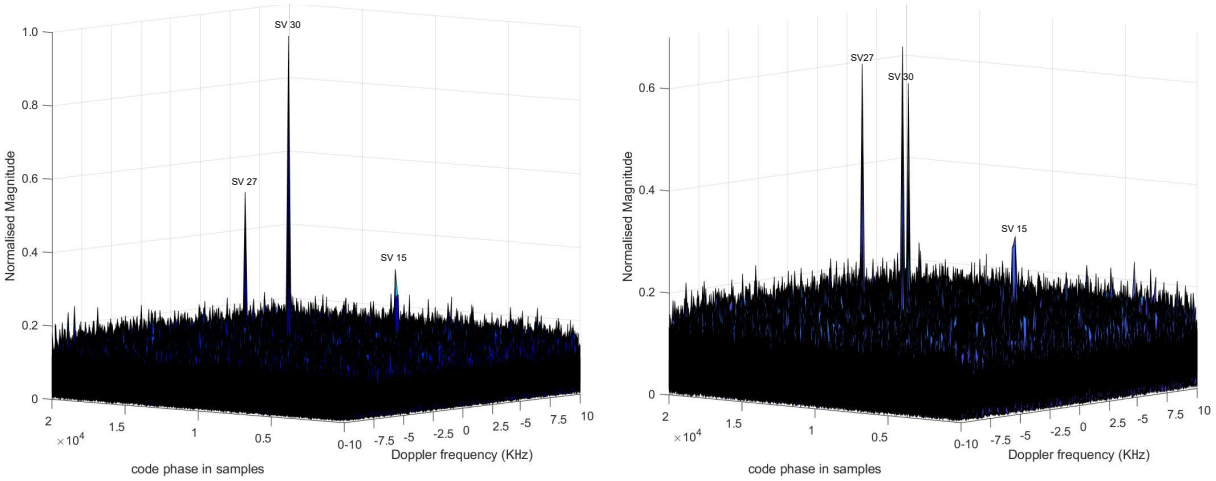


Figure 6.28: Sky plot of Galileo constellation during recording using Trimble GNSS planing online.

MATLAB to find out the available satellites from Galileo constellation.

Figure 6.29a shows the 3D plot of the acquisition matrix performed in MATLAB using the first 4 ms of the total 20 ms recorded data. Three satellites 30, 27 and 15 are identified, while satellites 02 and 36 are not detected, this is due to non-clear view to the sky on the north and eastern side. Figure 6.29b shows the acquisition matrix using the second 4 ms of the total 20 ms data, it can be seen that the correlation peak of the SV30 is split into two adjacent peak and the power is reduced by approximately 40%, this is due to the bit sign transition in the middle of the second 4 ms data samples.

To validate the system under linear shift method using 8 ms data with bit sign transition, the modified Correlator Sampler is initialised with second and third 4 ms of the recorded real data which produces 40000 concatenated I and Q samples and SV 30 is acquired in the simulation. Figure 6.30 shows the simulation waveform of the Acquisition Block in ModelSim utilising 8 ms of the real data samples initialised in the Correlator Sampler. In the figure, under the result waveform, the permutation indicates 40, code-phase 99, frequency-bin 69, detection '1' and the interrupt is high 1.



(a) Acquisition matrix using first 4 ms. (b) Acquisition matrix using second 4 ms.

Figure 6.29: Acquisition matrix obtained from Galileo satellites search using the first and the second 4 ms of the recorded total 20 ms data.

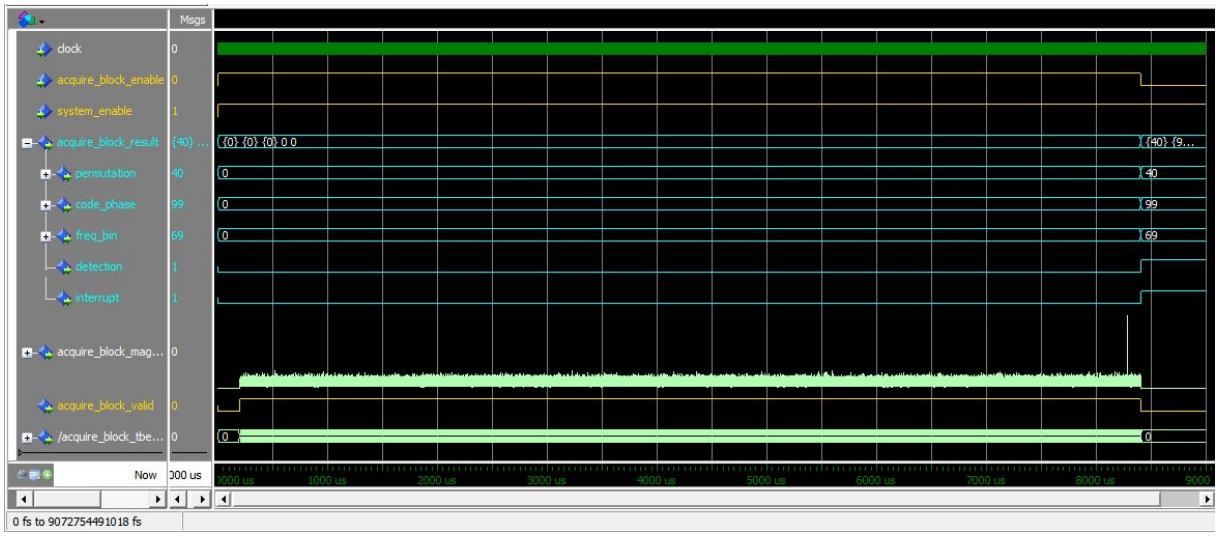


Figure 6.30: Acquisition Block simulation waveform in the ModelSim using the real recorded data with bit sign transition at 50% of the first 4ms of the total 8ms long real sampled signal. The peak is at code phase of 99 samples and frequency bin 69, detected in permutation cycle 40.

Figure 6.31 shows the 3 dimensional plot of the normalised magnitude against the code-phase and frequency bins. It can be seen in the figure that the peak is not splited due to bit sign transition when the linear shift method of $2T_c$ is utilised.

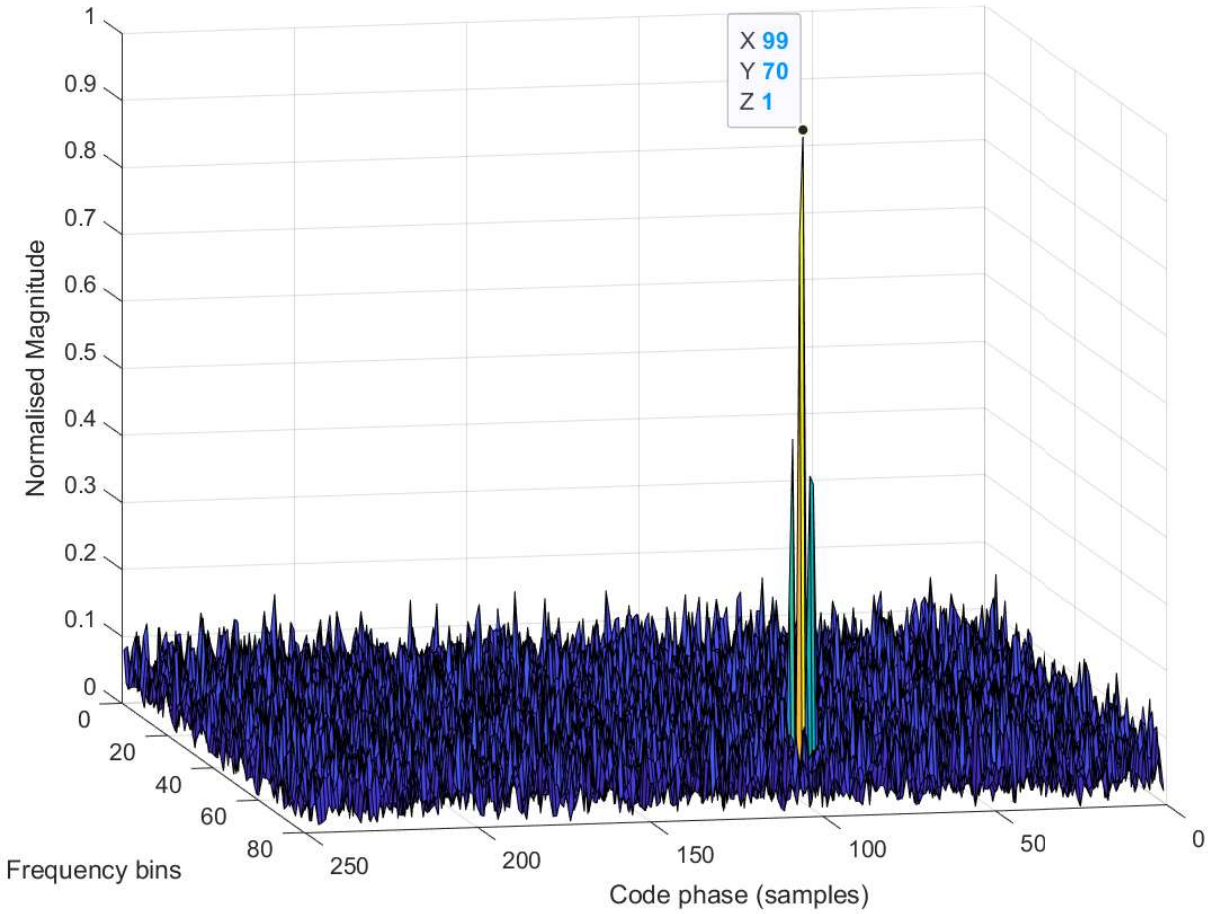


Figure 6.31: Galileo SV30, E1b acquisition matrix with bit sign transition at 50% of the first 4ms of the total 8ms long real sampled signal, simulated in ModelSim and plotted in MATLAB.

6.2.3 Validation test results using FPGA and real data

To validate the designed system in the FPGA, the developed VHDL code is targeted to a FPGA board and tested with the same recorded real Galileo E1 data samples. The hardware included the following components:

- DE10-standard development kit.
  This kit provided with Intel Cyclone V SE 5CSXFC6D6F31C6N system on chip. The chip includes 110K programmable logic elements and 5,761 Kbits embedded memory.
  Note: This kit is chosen due to its availability at the university. In addition, the design can be targeted to any other FPGA chip regardless to the vendor.
- USB cable (Type A to B) for FPGA programming and control.
- 12V DC power adapter.
- TTL-232R-RPi. USB to TTL level serial UART converter cable incorporating FTDI's FT232RQ USB to Serial UART interface IC device which handles all the USB signalling and protocols.
- Laptop.

The software included the following components:

- Quartus Prime version 18.0.0 standard edition.
- NIOS 2 software build tools for eclipse.
- MATLAB.

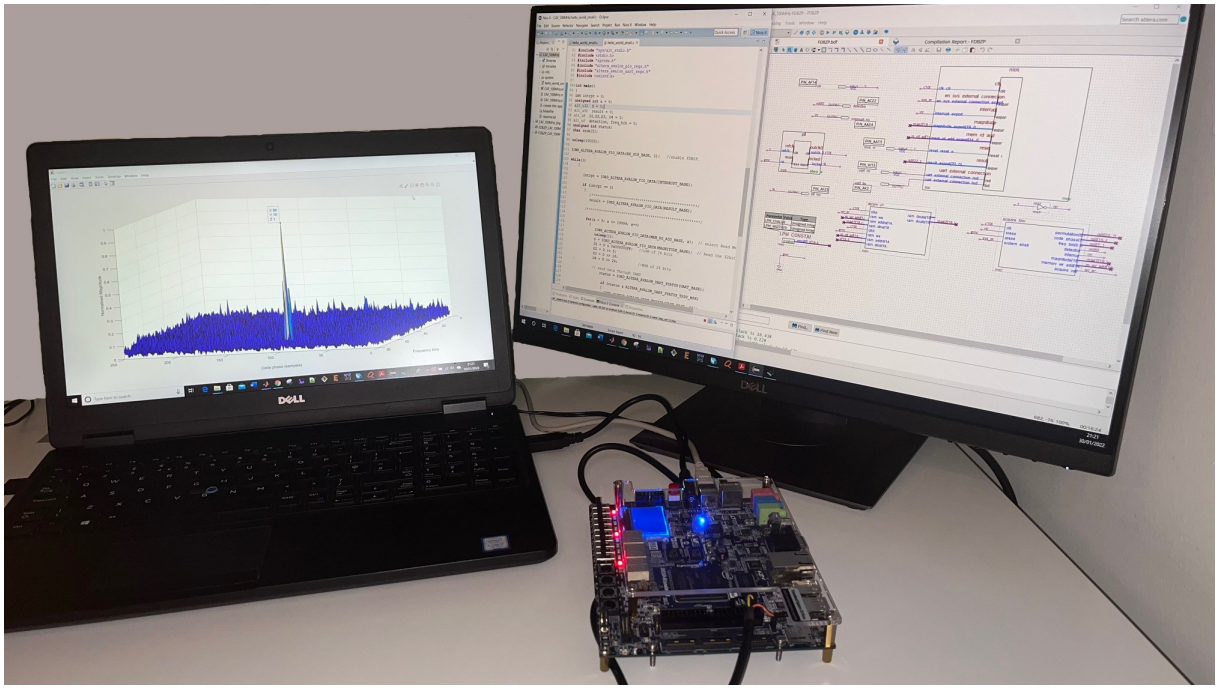


Figure 6.32: Hardware setup for validating the prposed design in FPGA.

Figure 6.32, shows the hardware set up during the test execution. The DE10-Standard development kit connected to the laptop using two links; the TTL-232R-RPi cable to transfer the data from the development kit to the laptop and USB cable (Type A to B) for programming the FPGA.
The design on FPGA composed of the following components:

- Acquisition Block, this is the incarnation of the FDBZP which represents the developed proposed system in VHDL.
- PLL intel FPGA IP, to convert the 50 MHz on-board clock to 100 MHz.
- Dpram_u19 internal on-chip dual port memory, this is a 32767 x 20 bit wide memory to contain the acquisition matrix output from Acquisition Block component.
  Note: This internal memory might be replaced with an external on board memory to reduce the internal memory utilisation.
- Nios 2 soft processor, to manage all internal units, control the Acquisition Block, and send the saved acquisition matrix on the Dpram_u19 to the laptop using UART-232 protocol.

The FPGA configuration file is generated by compiling the design on Quartus Prime software and downloaded to the FPFA on the DE10-Standard development kit. Figure

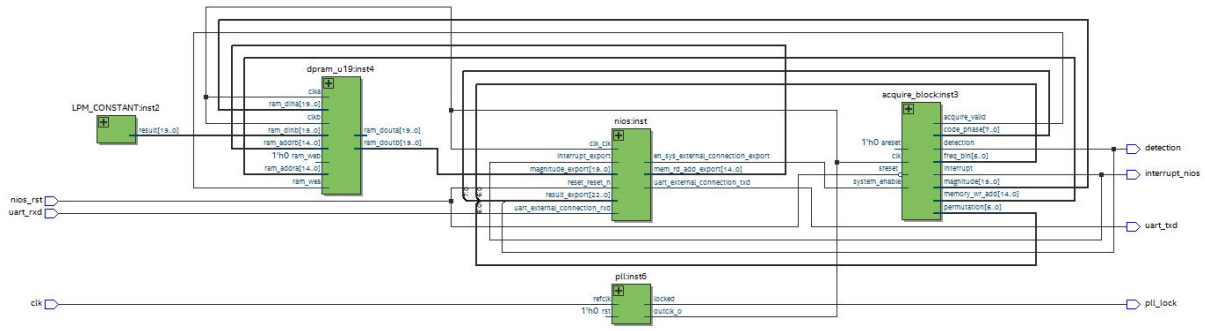


Figure 6.33: Register transfer level description of the proposed system in Quartus.

6.33, shows the register transfer level (RTL) description of the design generated from Quartus Prime software. To investigate the utilisation of Acquisition Block (FDBZP) on the FPGA, the block is compiled separetly to see the compilation report which describes the utilisation of the registers, DSP blocks and internal memory on the FPGA. Figure 6.34, shows the compilation report of the proposed design, it can be seen that the design utilises 27100 registers, 1,998.848 Kbits from total 5,761 Kbits memory bits and no DSP blocks as expected.

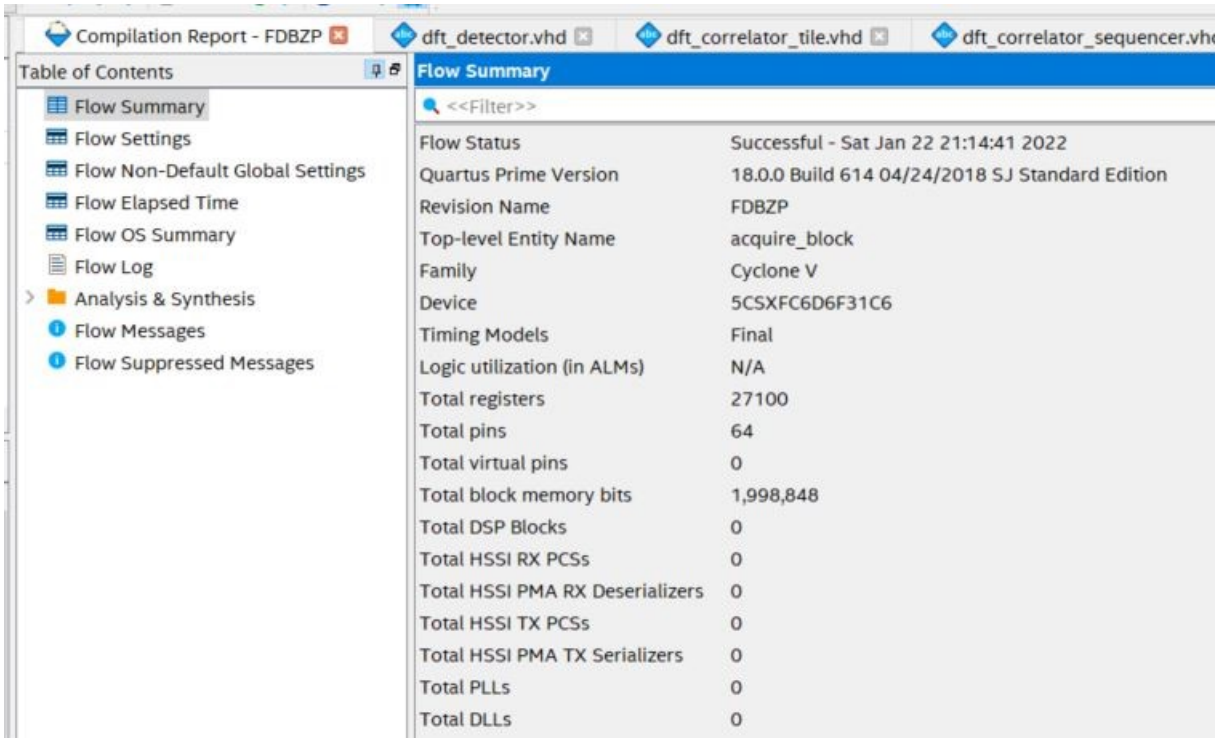


Figure 6.34: Compilation report of the proposed system in Quartus.

The Nios 2 soft processor included the following perepherals:

- UART (RS-232 serial port), to transfer the data from FPGA to the laptop.
- A one bit output port, to enable the Acquisition Block.
- A one bit input port, to interrupt the processor.
- A 23 bit input port, to read the result registers.

- A memory read port, to read the acquisition matrix from Dpram_u19.

The Nios 2 is programmed using "NIOS 2 software build tools for eclipse". This software is embedded in Quartus Prime. To avoid external connections, the same 8 ms real data (I&Q) which is used in simulation is initialised in Correlator Sampler. The 8 ms samples are truncated from 4 ms to 12 ms of total 20 ms recorded data, with a bit transition at the middle of the first 4 ms.

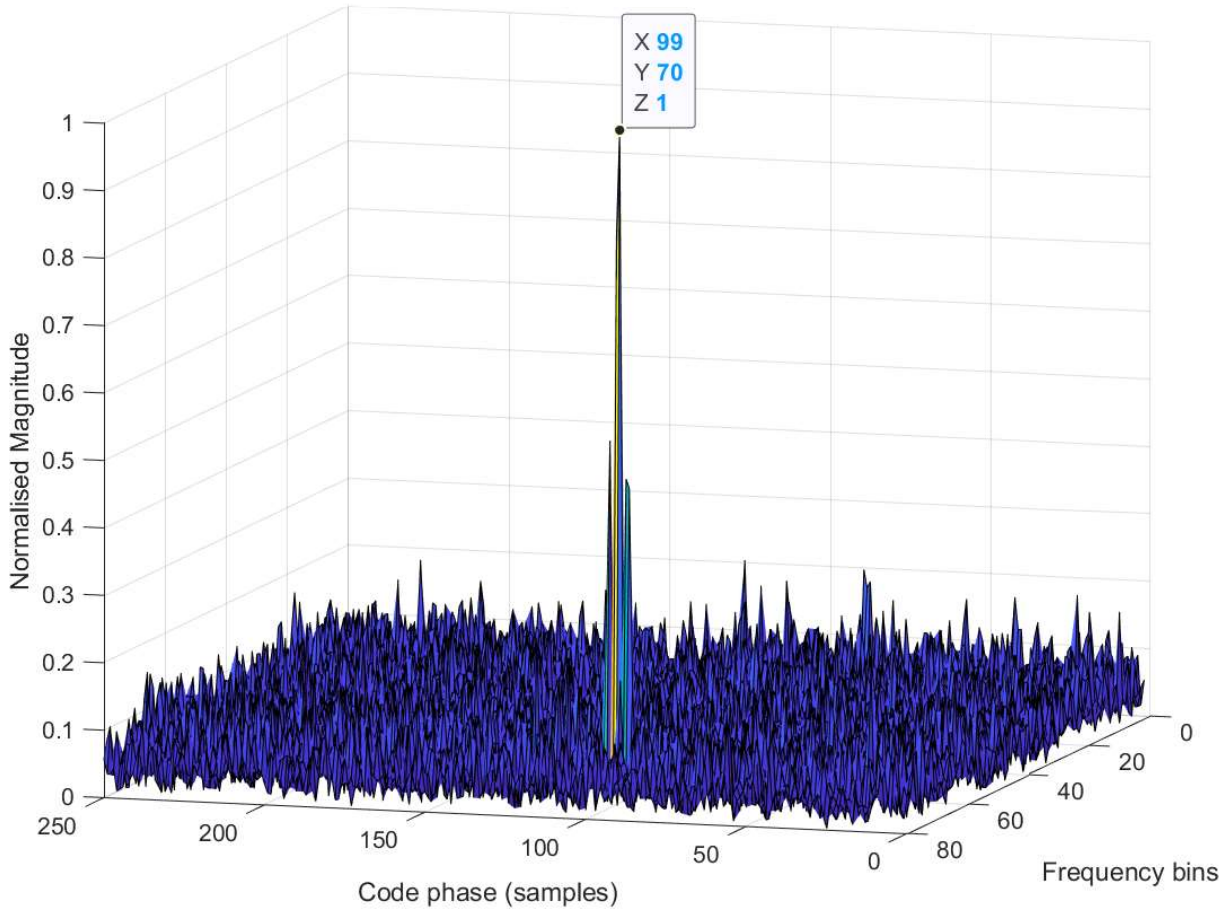


Figure 6.35: Galileo SV30, E1b acquisition matrix with bit sign transitionat 50% of the first 4ms of the total 8ms long real sampled signal, acquired in FPGA and plotted in MATLAB.

When the program is executed, the Nios 2 enables the Acquisition Block and wait for an interrupt signal from the block. The acquisition Block will acquire Galileo SV 30 independently, write the acquisition matrix to Dpram_u19, updates the result registers and inform the processor through interrupt pin. When the processor is interrupted, it reads first the 23 bit result registers. The result register is a concatenated version of the following variables:

- Detection = result[0].
- Frequency-bin = result[7..1].
- Code-phase = result[15..8].
- Permutation = result[22..16].

The processor will investigate the detection, if the value is '1', this indicate that the SV 30 is detected, the processor will read the 20000 samples of acquisition matrix from Dpram_u19 and send the matrix to the laptop through the UART port. On the laptop, the MATLAB reads the samples from the UART port and plot the three dimensional acquisition matrix as shown in Figure 6.35. The result from the FPGA is matching the simulation with the same real data. The code-phse is 99, frequency bin is 70 and the permutation in the result register is 40.

6.3 Summary

Acquisition Block is designed and developed using system engineering. System-level requirements are developed, then the system is broken down into subsystems, which are further broken down into lower-level entities until a complete understanding of the system is achieved from top to bottom. The lower-level entities are designed and built using VHDL in the Notepad++ code editor. Each described entity in VHDL is debugged and simulated using the ModelSim tool independently. The design is modified iteratively until the entities met the desired performance. The smaller entities are then combined into assemblies and then into subsystems from which the system is obtained. At each stage of the integration, debugging and simulation are conducted to verify successful integration. The system is then simulated for the desired properties, and the design is modified until the system met the desired performance. The system is designed without using any third-party fast Fourier transform FFT IP cores, DSP blocks, or multipliers. Avoiding double blocking and zero padding and utilising the parallel processing capability of the FPGA led to reducing the computation number and complexity, which led to a substantial improvement in the efficiency and the acquisition process. The feasibility and performance of the proposed approach are investigated by developing a structural test bench containing the main core under test and a stimulus generator, simulated in ModelSim. The complex DFT is generated by utilising the symmetry property of the real and imaginary parts of the DFT. This reduced the correlation computation to half, that is, only the first half (N/2) of the basis functions being used to obtain the complex DFT. This approach reduced the basis function matrix size from 80 x 240 to 40 x 120 for 3 bit quantisation, in other words, 75% fewer memory blocks and 50% fewer correlation tiles are required. The absolute value of the complex (I and Q) output of the Acquisition Block is approximated by using the JPL approximation algorithm to reduce the computational burden of obtaining the magnitudes of the I and Q vectors.

To avoid cycling through different allocated codes to different SVs and using potentially corrupted data during the recording process, the Correlator Sampler is initialised with five different pre-known patterns for system verification and debugging. The patterns are synthesised in MATLAB in a double-precision format and converted into a 3 bit fixed-point format. This step simplified the verification process of the system. The design is validated in the hardware using DE10-standard development kit. This kit provided with Intel Cyclone V SE 5CSXFC6D6F31C6N system on chip.

Finally, Acquisition Block detection probability time is between 0.406 and 16,206 ms for each satellite search. The choice of the system clock depends on the FPGA speed performance. In the high-end chips, the chip can be clocked at 200 MHz, which can reduce the detection probability time between 0.203 and 8,103 ms for each satellite search.

Chapter 7

Conclusion and future work

7.1 Thesis achievements

In this thesis, an FDBZP acquisition method for moderen GNSS signals is presented. The detailed design of FDBZP in FPGA is provided. The proposed method is implemented with no need to form the double blocks and zero padding the code, without using any third-party IPs such as FFT, DSP, or multipliers. The simulation results showed that the FDBZP method introduces less computation delay and power consumption by 50% during the partial correlation process owing to performing correlation on single blocks instead of double blocks (unlike the conventional DBZP method). The need to form double blocks and zero padding the code for continuity purposes is avoided by performing a partial correlation in the time domain instead of circular correlation using FFT cores. This provided more flexibility in selecting the sampling frequency, unlike correlation using the circular approach which the user must comply to radix 2 when utilising FFT blocks in FPGA. The complex DFT is computed in an efficient way by utilising the symmetry property about N/2. This reduced the correlation computation in the DFT block to half, that is, only the first half (N/2) of the basis functions used to obtain the complex DFT. This approach reduced the basis function matrix size from 80 x 240 to 40 x 120 for 3 bit quantisation, that is, 75% fewer memory blocks and 50% less correlation tiles.

When the correlation result lies in the first block, the computation requires 40600 clock cycles. For 80 permutations (including the starting block), the computation required 1.620.600 clock cycles. Assuming a 100 MHz system clock, the detection probability time was 0.406 ms when the code-phase was within the first block and 16,206 ms when the code-phase was within the last block (block 80), for each satellite search.

7.2 Recommendations for future work

- Development of an adaptive acquisition module
  The presented acquisition system is for the acquisition of different satellites of Galileo E1b signal, in other words it is suitable for single frequency acquisition. Given the advantages of the multi-constellation GNSS receiver, it may be preferable to adapt the designed acquisition system to acquire different satellites from various constellation with different code lengths.

- Development of tracking module
  In GNSS receivers, the obtained results from acquisition process should be passed

to the tracking system, to track the signal contentiously. For the seek of the development of the entire GNSS receiver in FPGA, the tracking system should be developed. Unlike GPS L1 signal, BOC(1,1) signal require a specific tracking method due to the presence of side peaks in the autocorrelation function which adds potential ambiguity.

- Development of position solution
  Calculating the user position was out of the scope of this work. For the seek of the development of the entire GNSS receiver, the user position should be calculated. This can be achieved by utilising the SoC processor.

- Minimizing FPGA resource utilization
  The logic resources and the embedded memory blocks remains the dominant factor in the size, cost and power consumption of the FPGA. A future work to optimise the utilisation of the resources and memory blocks is recommended.

Bibliography

Jérôme Leclère. Resource-efficient parallel acquisition architectures for modernized gnss signals. 2014.

Ávila Rodríguez and José Ángel. *On generalized signal waveforms for satellite navigation.* PhD thesis, München, Univ. der Bundeswehr, Diss., 2008, 2008.

DJR Van Nee and AJRM Coenen. New fast gps code-acquisition technique using fft. *Electronics Letters*, 27(2):158–160, 1991.

David M Lin and James BY Tsui. Comparison of acquisition methods for software gps receiver. In *Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2000)*, pages 2385–2390, 2000.

David Akopian. A fast satellite acquisition method. In *Proceedings of the 14th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2001)*, pages 2871–2881, 2001.

David M. Lin and James B Y Tsui. Acquisition through circular (56) references cited correlation by partition for gps c/a code and p(y) code, 2003a.

Nesreen I Ziedan and James L Garrison. Unaided acquisition of weak gps signals using circular correlation or double-block zero padding. In *PLANS 2004. Position Location and Navigation Symposium (IEEE Cat. No. 04CH37556)*, pages 461–470. IEEE, 2004.

Gregory W Heckler and James L Garrison. Implementation and testing of an unaided method for the acquisition of weak gps c/a code signals. *NAVIGATION, Journal of the Institute of Navigation*, 56(4):241–259, 2009.

Wen Zhang and Mounir Ghogho. Improved fast modified double-block zero-padding (fmdbzp) algorithm for weak gps signal acquisition. In *2010 18th European Signal Processing Conference*, pages 1617–1621. IEEE, 2010.

WJ Hurd, JI Statman, and VA Vilnrotter. High dynamic gps receiver using maximum likelihood estimationand frequency tracking. *IEEE transactions on aerospace and electronic systems*, (4):425–437, 1987.

Char-Dir Chung. Differentially coherent detection technique for direct-sequence code acquisition in a rayleigh fading mobile channel. *IEEE Transactions on Communications*, 43(2/3/4):1116–1126, 1995.

James Bao-Yen Tsui. *Fundamentals of global positioning system receivers: a software approach*, volume 173. John Wiley & Sons, 2005.

Elliott D Kaplan and Christopher Hegarty. *Understanding GPS/GNSS: Principles and applications*. Artech House, 2017.

Dan Doberstein. *Fundamentals of GPS Receivers: A Hardware Approach*. Springer Science & Business Media, 2011.

Peter Teunissen and Oliver Montenbruck. *Springer handbook of global navigation satellite systems*. Springer, 2017.

B Rama Rao, K McDonald, and Waldemar Kunysz. *GPS/GNSS Antennas*. Artech House, 2013.

Solomon W Golomb and Guang Gong. *Signal design for good correlation: for wireless communication, cryptography, and radar*. Cambridge University Press, 2005.

Simon Haykin. *Communication systems*. John Wiley & Sons, 2008.

Kai Borre, Dennis M Akos, Nicolaj Bertelsen, Peter Rinder, and Søren Holdt Jensen. *A software-defined GPS and Galileo receiver: a single-frequency approach*. Springer Science & Business Media, 2007.

K Sam Shanmugan and Arthur M Breipohl. *Random signals, detection, estimation and data analysis*. 1988.

Bradford W Parkinson, Per Enge, Penina Axelrad, and James J Spilker Jr. *Global positioning system: Theory and applications, Volume I*. American Institute of Aeronautics and Astronautics, 1996.

GPS Navstar. Interface specification is-gps-800d. *US Air Force, Tech. Rep*, 2013.

European Union. *European GNSS (Galileo) open service: Signal in space interface control document*. Office for Official Publications of the European Communities, 2010.

John W Betz. Binary offset carrier modulations for radionavigation. *Navigation*, 48(4): 227–246, 2001.

John G Grimes. Global positioning system standard positioning service performance standard. *Washington, DC, USA: US Department of Defense*, 2008.

Ahao Wang, Junping Chen, Yize Zhang, Jiexian Wang, and Bin Wang. Performance evaluation of the cnav broadcast ephemeris. *The Journal of Navigation*, 72(5):1331–1344, 2019.

Michael J Dunn and DAF DISL. Global positioning system directorate systems engineering & integration interface specification is-gps-200, 2012.

RD Fontana and R Cheung. W., and stansell, t, a.(2001). the new l2 civil signal. *Proceeding of the institute of navigation ION GPS-2001, Salt Lake City, Utah, USA*, pages 617–631.

AJ Van Dierendonck. The new l5 civil gps signal. *Gps World*, 11(9):64–71, 2000.

GPS Directorate. Navstar gps space segment/user segment l5 interfaces (is-gps-705b). Technical report, Technical report, 2011.

Pratap Misra and Per Enge. Global positioning system: signals, measurements and performance second edition. *Global Positioning System: Signals, Measurements And Performance Second Editions,*, 2006.

ICD GLONASS. Global navigation satellite system glonass, interface control document, navigational radio signal in bands l1, l2, 2008.

ICD GLONASS. Global navigation satellite system glonass, interface control document, code division multiple access open service navigation signal in l3 frequency band, 2016.

ICD Galileo. Galileo open service, signal in space interface control document (os sis icd), 2008.

OS SDD Galileo. Galileo, open service, open service service definition document, 2019.

Fabio Dovis, Letizia Lo Presti, Maurizio Fantino, Paolo Mulassano, and Jérémie Godet. Comparison between galileo cboc candidates and boc (1, 1) in terms of detection performance. *International Journal of Navigation and Observation*, 2008, 2008.

Alper Ucar. *A subsampling delta-sigma modulator for global navigation satellite systems*. PhD thesis, University of Westminster, 2010.

Emilie Rebeyrol, Christophe Macabiau, Laurent Lestarquit, Lionel Ries, Jean-Luc Issler, Marie-Laure Boucheret, and Michel Bousquet. Boc power spectrum densities. 2005.

L Ries, F Legrand, L Lestarquit, W Vigneau, and JL Issler. Tracking and multipath performance assessments of boc signals using a bit-level signal processing simulator. In *Proceedings of the 16th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS/GNSS 2003)*, pages 1996–2010, 2001.

J Godet. Technical annex to galileo srd signal plans. *STF annex SRD*, 2003, 2001.

M Soellner and Ph Erhard. Comparison of awgn code tracking accuracy for alternative-boc, complex-loc and complex-boc modulation options in galileo e5-band. In *Proc. of ENC GNSS*, 2003.

OSPS BeiDou. Beidou navigation satellite system open service performance standard (version 2.0), 2018a.

DBDS BeiDou. Development of beidou navigation satellite system (version 4.0), 2019a.

Martin A Rothblatt. *Radiodetermination satellite services and standards*. Artech House, Inc., 1987.

Shaofeng Bian, Jihang Jin, and Zhaobao Fang. The beidou satellite positioning system and its positioning accuracy. *Navigation*, 52(3):123–129, 2005.

ICD BeiDou. Beidou navigation satellite system signal in space interface control document, open service signal (version 2.1), 2016.

ICD BeiDou. Beidou navigation satellite system signal in space interface control document open service signal b1i (version 3.0), 2019b.

ICD BeiDou. Beidou navigation satellite system signal in space interface control document open service signal b1c (version 1.0), 2017a.

Z Yao, M Lu, and ZM Feng. Quadrature multiplexed boc modulation for interoperable gnss signals. *Electronics letters*, 46(17):1234–1236, 2010.

Zheng Yao, Jiayi Zhang, and Mingquan Lu. Ace-boc: dual-frequency constant envelope multiplexing for satellite navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 52(1):466–485, 2016.

Mingquan Lu, Wenyi Li, Zheng Yao, and Xiaowei Cui. Overview of bds iii new signals. *Navigation*, 66(1):19–35, 2019.

ICD BeiDou. Beidou navigation satellite system signal in space interface control document open service signal b2a (version 1.0), 2017b.

ICD BeiDou. Beidou navigation satellite system signal in space interface control document open service signal b1c and b2a (test version), 2017c.

ICD BeiDou. Beidou navigation satellite system signal in space interface control document open service signal b2b (beta version), 2019c.

Kai Zhang. Generalised constant-envelope dualqpsk and altboc modulations for modern gnss signals. *Electronics letters*, 49(21):1335–1337, 2013.

ICD BeiDou. Beidou navigation satellite system signal in space interface control document open service signal b3i (version 1.0), 2018b.

Constantine A Balanis. *Antenna theory: analysis and design*. John wiley & sons, 2016.

Glenn Robb. Circularly polarized antennas explained, without the math. *Antenna Test Lab Co.*, 5:13, 2017.

Korkut Yegin. Antennas and front-end in gnss. *Multifunctional Operation and Application of GPS*, page 165, 2018.

YT Morton, James BY Tsui, David M Lin, Liyeh L Liou, Mikel M Miller, John Schamus, and Qihou Zhou. Assessment and handling of ca code self-interference during weak gps signal acquisition (preprint). Technical report, AIR FORCE RESEARCH LAB WRIGHT-PATTERSON AFB OH SENSORS DIRECTORATE, 2003.

Rodger E Ziemer and William H Tranter. *Principles of communications*. John Wiley & Sons, 2014.

Ramón López La Valle, Javier G García, Pedro A Roncagliolo, and Carlos H Muravchik. A practical rf front-end for high performance gnss receivers. In *2011 International Conference on Localization and GNSS (ICL-GNSS)*, pages 104–109. IEEE, 2011.

Behzad Razavi and Razavi Behzad. *RF microelectronics*, volume 2. Prentice Hall New Jersey, 1998.

Alan Douglas. The legacies of edwin howard armstrong. *Proceedings of the Radio Club of America*, 64(3), 1990.

Fanzhen Meng, Hong Liu, Mingliang Wang, Xiaolin Zhang, and Tong Tian. Rf low power subsampling architecture for wireless communication applications. *EURASIP Journal on Wireless Communications and Networking*, 2016(1):1–15, 2016.

José R García Oya, Andrew Kwan, Fernando Muñoz Chavero, Fadhel M Ghannouchi, Mohamed Helaoui, Fernando Márquez Lasso, Enrique López-Morillo, and Antonio Torralba Silgado. Subsampling receivers with applications to software defined radio systems. *Data Acquisition Applications*, pages 165–194, 2012.

Yonina C Eldar. *Sampling theory: Beyond bandlimited systems*. Cambridge University Press, 2015.

Vinh T Tran, Nagaraj C Shivaramaiah, Thuan D Nguyen, Joon W Cheong, Eamonn P Glennon, and Andrew G Dempster. Generalised theory on the effects of sampling frequency on gnss code tracking. *The Journal of Navigation*, 71(2):257, 2018.

Michael S Braasch and AJ Van Dierendonck. Gps receiver architectures and measurements. *Proceedings of the IEEE*, 87(1):48–64, 1999.

Oliver Garcia, Angel Magallanes, and Qin Hong Lei. Coherent vs. noncoherent integration a deal of gain for gps weak signal acquisition. In *Proceedings of 2010 International Conference on Circuit and Signal Processing & 2010 Second IITA International Joint Conference on Artificial Intelligence (Volume 3)*, 2010.

Myriam Foucras. *PERFORMANCE ANALYSIS OF MODERNIZED GNSS SIGNALS ACQUISITION*. PhD thesis, Institut National Polytechnique de Toulouse (INP Toulouse), 2015.

Daniele Borio and Dennis Akos. Noncoherent integrations for gnss detection: Analysis and comparisons. *IEEE transactions on Aerospace and electronic systems*, 45(1):360–375, 2009.

David M Lin and James BY Tsui. Acquisition through circular correlation by partition for gps c/a code and p (y) code, May 20 2003b. US Patent 6,567,042.

Myriam Foucras, Olivier Julien, Christophe Macabiau, and Bertrand Ekambi. A novel computationally efficient galileo e1 os acquisition method for gnss software receiver. In *ION GNSS 2012, 25th International Technical Meeting of The Satellite Division of the Institute of Navigation*, pages pp–xxxx, 2012.

Ehsan ul Haq, Muhammad Kazim Hafeez, Muhammad Salman Khan, Shoaib Sial, and Arshad Riazuddin. Fpga implementation of a low power, processor-independent and reusable system-on-chip platform. In *2009 International Conference on Emerging Technologies*, pages 337–341. IEEE, 2009.

Chun-Ming Huang, Chien-Ming Wu, Chih-Chyau Yang, Wei-De Chien, Shih-Lun Chen, Chi-Shi Chen, Jiann-Jenn Wang, and Chin-Long Wey. Implementation and prototyping of a complex multi-project system-on-a-chip. In *2009 IEEE International Symposium on Circuits and Systems*, pages 2321–2324. IEEE, 2009.

Goran Jurkovic and Vlado Sruk. Remote firmware update for constrained embedded systems. In *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1019–1023. IEEE, 2014.

Loren K Shade. Implementing secure remote firmware updates. In *Embedded Systems Conference*, 2011.

Gerald Kaiser. *A friendly guide to wavelets*. Springer Science & Business Media, 2010.

Walt Kester. *Mixed-signal and DSP design techniques*, volume 1. Newnes, 2003.

BK Levitt and GA Morris. An improved digital algorithm for fast amplitude approximations of quadrature pairs. *DSN Progress Report*, 42:40, 1977.

Appendix A

Acquisition Block entity

Acquire Block

```
1  library ieee;
   use ieee.std_logic_1164.all;
   use ieee.numeric_std.all;
   -- Design Libraries
   use design_pkg.all;
   use acquire_pkg.all;
   use acquire_mem_pkg.all;

   ------------------------------------------------------------------
10 -- Entity
   ------------------------------------------------------------------
   entity acquire_block is
    port(
     clk              : in  std_logic;
     sreset           : in  std_logic;
     areset           : in  std_logic;
     system_enable    : in std_logic;
     status           : out status_t;
19   magnitude        : out u19_t;
     memory_wr_add    : out u14_t;
     acquire_valid    : out std_logic
    );
   end;

   ------------------------------------------------------------------
   -- Architecture
   ------------------------------------------------------------------
28 architecture struct of acquire_block is

   ------------------------------------------------------------------
   -- Signal Declarations
   ------------------------------------------------------------------
   -- Correlator Block
   signal corr_block_dout : corr_t;
   -- DFT Block
   signal  mem_rd_ad      : u15_t;
37 signal  mem_valid      : std_logic;
   signal sts             : status_t;
   signal mag             : u19_t;
   signal mem_wr_add      : u14_t;
   signal detec_valid     : std_logic;
   -- Correlator Memory
   signal ram_dob       : u23_t;
   -- DFT Block
   signal i               : q11_t;
46 signal q               : q11_t;
   -- Enable signal
   signal sys_en          : std_logic;

   ------------------------------------------------------------------
   begin
```

```
-------------------------------------------------------------------
-- Signal Assignments
-------------------------------------------------------------------
-- Acquire Block Output Signals
status          <= sts;
magnitude       <= mag;
memory_wr_add   <= mem_wr_add;
acquire_valid   <= detec_valid;
sys_en          <= system_enable and (not sts.interrupt);
-- DFT Block
i   <= signed (ram_dob(23 downto 12) );
q   <= signed (ram_dob(11 downto 0)  );

-------------------------------------------------------------------
-- Component Instantiations
-------------------------------------------------------------------
-- Correlator Block
 correlator_0 : correlator_block
  port map(
   clk             => clk,
      sreset          => sreset,
      areset          => areset,
      system_enable   => sys_en,
      -- Output
      corr_block_dout =>  corr_block_dout
  );

-------------------------------------------------------------------
-- 65536x24bit Dual port RAM For Correlator Memory
corr_mem : dpram_u23
 generic map
 (
  addr_bits_g => 16,
  init_data_g => corr_mem_init_c
 )
  port map
  (
      clka      => clk,
   reset     => '0',
      ram_wea   => corr_block_dout.page1_valid,
      ram_addra => corr_block_dout.corr_ram_wr,
      ram_dina  => corr_block_dout.IQ_corr,
      ram_douta => open,
      clkb      => clk,
      ram_web   => '0',
      ram_addrb => mem_rd_ad,
      ram_dinb  => x"000000",
      ram_doutb => ram_dob
  );
-------------------------------------------------------------------
-- DFT Block
 dft_block_0 : dft_block
  port map(
   clk             => clk,
      sreset          => sreset,
      areset          => areset,
      system_enable   => system_enable,
      -- Input Signal
      enable          => corr_block_dout.page2_valid,
      i               => i,
      q               => q,
      -- Output Memory Read Add
      mem_rd_ad       => mem_rd_ad,
      mem_valid       => mem_valid,
      -- Output
      status          => sts,
      magnitude       => mag,
      memory_wr_add   => mem_wr_add,
      detector_valid  => detec_valid
  );

end;
-------------------------------------------------------------------
```

Appendix B

Acquisition Block sub-level entity's

Correlator Block

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
-- Design Libraries
use design_pkg.all;
use acquire_pkg.all;

--------------------------------------------------------------------
-- Entity
--------------------------------------------------------------------
entity correlator_block is
 port(
  clk              : in  std_logic;
  sreset           : in  std_logic;
  areset           : in  std_logic;
  system_enable    : in  std_logic;
  -- Output
  corr_block_dout : out corr_t
 );
end;

--------------------------------------------------------------------
-- Architecture
--------------------------------------------------------------------
architecture struct of correlator_block is

--------------------------------------------------------------------
-- Signal Declarations
--------------------------------------------------------------------
-- Sampler
signal i            : q2_t;
signal q            : q2_t;
signal smpler_valid : std_logic;
-- Coder
signal coder_dout : coder_t;
-- Correlator
signal corr_dout  : corr_t;

--------------------------------------------------------------------
begin

--------------------------------------------------------------------
-- Signal Assignments
--------------------------------------------------------------------
-- Output
corr_block_dout <= corr_dout;
--------------------------------------------------------------------
-- Component Instantiations
--------------------------------------------------------------------
-- Coder
 coder_0 : correlator_coder
```

```
  port map(
       clk              => clk,
       sreset           => sreset,
       areset           => areset,
       enable           => system_enable,
                coder_dout       => coder_dout
  );

-------------------------------------------------------------------
-- Sampler
    sampler_0 : correlator_sampler
     port map(
          clk              => clk,
       sreset           => sreset,
       areset           => areset,
       enable           => coder_dout.valid,
          -- Output data
                i                => i,
          q                => q,
          valid            => smpler_valid
    );

-------------------------------------------------------------------
-- Correlator
 correlator_0 : correlator
  port map(
                clk              => clk,
    sreset           => sreset,
    areset           => areset,
    enable           => smpler_valid,
          -- IF RAM Interface
          i                => i,
          q                => q,
          -- Correlator Coder RAM Interface
          coder            => coder_dout,
          -- Output
          correlator_dout => corr_dout
   );

end;
-------------------------------------------------------------------
```

DFT Block

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
-- Design Libraries
use design_pkg.all;
use acquire_pkg.all;

-------------------------------------------------------------------
-- Entity
-------------------------------------------------------------------
entity dft_block is
 port(
  clk              : in  std_logic;
  sreset           : in  std_logic;
  areset           : in  std_logic;
  system_enable    : in std_logic;
  -- Input Signal
  enable           : in  std_logic;
  i                : in  q11_t;
  q                : in  q11_t;
  -- Output Memory Read Add
  mem_rd_ad        : out u15_t;
  mem_valid        : out std_logic;
  -- Output
  status           : out status_t;
  magnitude        : out u19_t;
  memory_wr_add    : out u14_t;
  detector_valid   : out std_logic
```

```
 );
end;

33 ----------------------------------------------------------------------
-- Architecture
----------------------------------------------------------------------
architecture struct of dft_block is

----------------------------------------------------------------------
-- Signal Declarations
----------------------------------------------------------------------

42 -- base_function_generator
signal baser           : baser_t;
signal mem_rdad        : u15_t;
signal memvalid        : std_logic;

-- dft_correlator_tile_logic
signal ac_i_q          : u39_array_t(0 to 79);
signal ac_i_q_1        : u39_array_t(0 to 40);
signal ac_i_q_2        : u39_array_t(40 to 80);
51 signal valid           : std_logic_vector(0 to 79);
signal ac_i            : q19_t;
signal ac_q            : q19_t;
signal vld             : std_logic;

-- dft_detector
signal ram_wr_add      : u14_t;
signal mag             : u19_t;
signal mem_wr_add      : u14_t;
60 signal detector_vld    : std_logic;
signal sts             : status_t;
----------------------------------------------------------------------
begin

----------------------------------------------------------------------
-- Concurrent signals
----------------------------------------------------------------------
-- Output Signals
69 mem_rd_ad        <= mem_rdad;
mem_valid        <= memvalid ;
magnitude        <= mag;
memory_wr_add    <= mem_wr_add;
detector_valid   <= detector_vld;
status           <= sts;

--
ac_i_q(0 to 40) <= ac_i_q_1(0 to 40);
78 ac_i_q(41 to 79) <= ac_i_q_2(41 to 79);
----------------------------------------------------------------------
-- Components
----------------------------------------------------------------------
-- base_function_generator
 baser_0 : dft_baser
  port map(
   clk         => clk,
      sreset           => sreset,
87    areset           => areset,
      baser_valid      => enable,
   -- Correlator Memory
      mem_rd_ad        => mem_rdad,
      mem_valid        => memvalid,
      -- Output Data
            baser_dout       => baser
  );

96 ----------------------------------------------------------------------
-- Correlator Tiles

gen_tile : for j in 0 to 40 generate
 tile : dft_correlator_tile
  port map(
   clk              => clk,
```

```
      sreset          => sreset,
      areset          => areset,
      -- Input Signal
      i               => i,
         q               => q,
         -- Input Basis Functions
         basis_i         => baser.cs_base_function(j),
         basis_q         => baser.sn_base_function(j),
         baser_cnt       => baser.baser_cnt,
         valid           => baser.valid,
         end_tick        => baser.end_tick,
         -- Output
         acc_i           => open,
            acc_q           => open,
      acc_i_q         => ac_i_q_1(j),
      acc_i_q_2       => ac_i_q_2(80-j),
            corr_valid      => valid(j)
   );
end generate;

-------------------------------------------------------------------
-- Sequencer
sequencer_0 : dft_correlator_sequencer
 port map(
   clk             => clk,
   sreset          => sreset,
   areset          => areset,
      -- Timing
      seq_enable      => valid(0),
      end_tick        => baser.end_tick,
      -- Input from Correlator Tile I&Q
      mram_rd_array   => ac_i_q,
      -- Output from Sequencer
      acc_i           => ac_i,
      acc_q           => ac_q,
   mram_wr_add     => ram_wr_add,
      valid           => vld
  );

-------------------------------------------------------------------
-- Sequencer
detector_0 : dft_detector
 port map(
   clk             => clk,
   sreset          => sreset,
   areset          => areset,
   system_enable   => system_enable,
      -- Input from Sequenser
      i               => ac_i,
      q               => ac_q,
      mram_wr_add     => ram_wr_add,
      valid           => vld,
      -- Detector Output
   status          => sts,
      magnitude       => mag,
      memory_wr_add   => mem_wr_add,
      detector_valid  => detector_vld

  );

end;
-------------------------------------------------------------------
```