TAMPERE UNIVERSITY OF TECHNOLOGY
Department of Communications Engineering

ZHANG JIE

Delay Trackers for Galileo CBOC Modulated Signals and Their Simulink-based Implementations

Master of Science Thesis

Subject Approved by Department Council
November 4, 2009
Examiners: Docent Elena-Simona Lohan
M.Sc. Mohammad Zahidul Hasan Bhuiyan

Abstract

Tampere University of Technology
Master's Degree Program in Radio Frequency Electronics
Department of Communications Engineering
Zhang, Jie: Delay Trackers for Galileo CBOC Modulated Signals and Their Simulink-based Implementations
Master of Science Thesis, 76 pages
February, 2010
Examiners: Dr. Docent Elena-Simona Lohan
M.Sc. Mohammad Zahidul Hasan Bhuiyan
Keywords: Galileo, CBOC modulation, delay tracker, Simulink

Galileo will be the future European Global Navigation Satellite Systems (GNSSs), which is going to provide high availability, increased accuracy and various location services. This new satellite system proposes the use of a new modulation, namely the Composite Binary Offset Carrier (CBOC) modulation, which motivates the research on GNSS receiver with this new modulation.

Code tracking is one of the main functions in a GNSS receiver and its task is to give an accurate estimation of the code delay. The accuracy of this code delay estimation is strictly connected with the accuracy of user position computation. One typical code tracking structure is the code tracking loop. The code tracking algorithms or delay trackers used in code tracking loop are the main aspect, which affects the performance of code tracking loop. Various typical delay trackers are studied in this thesis.

Simulation is one important issue in the design and analysis of any communication system or navigation system. One method for testing delay trackers and effects from different tracking algorithms can be realized in the simulation tool, such as a software receiver. The simulation tool makes it convenient to test various algorithms used in the receiver and to investigate the receiver performance before the algorithms are built in the real devices. On the other hand, the implementation of delay trackers in a software receiver can be also helpful for further developing the simulation tool.

The goal of this thesis has been to develop and analyze the implementations of various code delay trackers for Galileo systems via Simulink tool. The analysis has also helped to further develop the model in order to include realistic receiver constraints for mass-market application. The performance of the delay trackers is measured in terms of Root Mean Square Error (RMSE), tracking error variance and Multipath Error Envelopes (MEEs).

Preface

This Master of Science Thesis has been written for Department of Communications Engineering (DCE) at the Tampere University of Technology (TUT), Tampere, Finland. This thesis work has been carried out within two research projects of the Department of Communications Engineering at TUT: the Academy of Finland-funded project "Digital Processing Algorithms for Indoor Positioning Systems (ACAPO)" and the EU FP7-funded project under grant agreement number 227890 "Galileo Ready Advanced Mass MArket Receiver (GRAMMAR) ".

I would like to express gratitude to my thesis supervisor Docent Elena Simona Lohan and Mohammad Zahidul Hasan Bhuiyan for their support throughout the research work and their useful comments on my thesis. I would also like to express my appreciation to my colleagues Bashir Siddiqui and Danai Skournetou for their friendly supporting and offering me their help whenever I needed.

I am thankful to all of my Chinese friends in Tampere for their great concerning, encouragement and company during my M.Sc studies.

Finally, I express my gratitude to my parents for their endless love and inspiration and my love, Toni, for helping me, believing in me and loving me.

Tampere, Finland

Zhang Jie

Orivedenkatu 8 C 71
33720 Tampere
Finland
jie.zhang@tut.fi
Tel. +358468477485

Contents

List of Abbreviations

ACF	Auto Correlation Function
AltBOC	Alternative Binary Offset Carrier
AME	Absolute Mean Error
ARM	Advanced RISC Machine
ARNS	Aeronautical Radio Navigation Services
AWGN	Additive White Gaussian Noise
BOC	Binary Offset Carrier
BPSK	Binary Phase Shift Keying
C/A	Coarse/Acquisition
CBOC	Composite Binary Offset Carrier
CDMA	Code Division Multiple Access
CNR	Carrier-to-Noise-Ratio
CosBOC	Cosine Binary Offset Carrier
CS	Commercial Service
DISG	Digitized IF Signal Generator
DLL	Delay Lock Loop
DP	Dot Product discriminator
DSP	Digital Signal Processing
E	Early correlator
EGNOS	European Geostationary Navigation Overlay System
ESA	European Space Agency
ETRI	Electronics and Telecommunications Research Institute
EU	European Union
FFT	Fast Fourier Transform
FLL	Frequency Lock Loop
FPGA	Field Proframmable Gate Array
GDISS	GNSS Digitized IF Signal Simulation
GIOVE-A	Galileo In-Orbit Validation Element-A
GIOVE-B	Galileo In-Orbit Validation Element-B
GLONASS	GLobal Orbiting NAvigation Satellite System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GRAMMAR	Galileo Ready Advanced Mass MArket Receiver
GRANADA	Galileo Receiver ANalysis And Design Application
HRC	High Resolution Correlator
I	In-phase

IF	Intermediate Frequency
L	Late correlator
LOS	Line-Of-Sight
MBOC	Multiplexed Binary Offset Carrier
MEE	Multipath Error Envelope
MEO	Medium Earth Orbit
MGD	Multiple Gate Delay
MP	Multipath error used in SBME
MTSAT	Multi-function Transport SATellite system
NavSAS	Navigation Signal Analysis and Simulation
NCO	Numerically Controlled Oscillator
NLOS	Non Line-Of-Sight
OS	Open Service
OS SIS ICD	Open Service Signal-In-Space Interference Control Document
PLAN	Position Location And Navigation
PLL	Phase Lock Loop
PRN	Pseudo-Random Noise
PRS	Public Regular Service
PSD	Power Spectral Density
Q	Quadrature
QoS	Quality of Service
QZSS	Quasi-Zenith Satellites System
RDG	Raw Data Generation
RMSE	Root Mean Square Error
RNSS	Radio Navigation Satellite Service
sps	Symbols per second
SAR	Search And Rescue service
SBME	Slope Based Multipath Estimation
SinBOC	Sine Binary Offset Carrier
SoL	Safety-of-Life-Service
TMBOC	Time Multiplexed Binary Offset Carrier
TTFF	Time To First Fix
TUT	Tampere University of Technology
US	United States
USSR	Union of Soviet Socialist Republic
VE	Very Early correlator
VL	Very Late correlator
VVE	Very Very Early correlator
VVL	Very Very Late correlator
WCDMA	Wideband Code Division Multiple Access

List of Symbols

w	Amplitude weighting factor
n	AWGN noise
r_{E1_BB}	Baseband of received E1 signal
T_C	Chip period
f_C	Chip rate
E_b	Code symbol energy
$\otimes$	Convolution operator
I_0	Correlation value at prompt
I_{+2}	Correlation value at very late
Δ	Correlator spacing
$\delta(t)$	Dirac pulse
I_{VE}	Early correlator output
$\hat{f}$	Estimated carrier frequency
$\hat{\tau}_{LOS}^{(i)}$	Estimated LOS delay in second
I_E	In-phase component for Early correlator
I_L	In-phase component for Late correlator
I_P	In-phase component for Prompt correlator
I_{VL}	In-phase component for Very Late correlator
I_{VE}	In-phase component for Very Early correlator
I_{VVE}	In-phase component for Very Very Early correlator
I_{VVL}	In-phase component for Very Very Late correlator
$c_{k,n}$	k^{th} chip corresponding to n^{th} symbol
m_L	Late slope of normalized ideal correlation function
$d(t)$	Modulation waveform
b_n	n^{th} complex data symbol
N	Number of point
a_{SBME}	Optimized coefficient of SBME
τ_i	Delay for i^{th} path
α_i	Gain for i^{th} path
G_{MBOC}	Power spectral density of MBOC
ρ_i	Pseudorange measurement
Q_E	Quadrature-phase component for Early correlator
Q_L	Quadrature-phase component for Late correlator
Q_P	Quadrature-phase component for Prompt correlator
Q_{VE}	Quadrature-phase component for Very Early correlator

Q_{VL}	Quadrature-phase component for Very Late correlator
Q_{VVE}	Quadrature-phase component for Very Very Early correlator
Q_{VVL}	Quadrature-phase component for Very Very Late correlator
r_{E1}	Received E1 signal
$P_{T_{BOC2}}(\bullet)$	Rectangular pulse of support T_C / N_{BOC2}
$SVNi$	Satellite Vehicle i
$sign(\cdot)$	Signum operator
$S_{SinBOC}(t)$	SinBOC signal wave
N_{BOC1}	SinBOC(1,1) modulation order
N_{BOC2}	SinBOC(6,1) modulation order
d	Spacing between early and late correlator
c	Speed of light
S_F	Spreading factor
f_{SC}	Sub-Carrier frequency
T_{sym}	Symbol period
$\tau_{LOS}^{(i)}$	True LOS delay in second
$G_{SinBOC(m,n)}$	Unit power PSD of a sine-phased BOC modulation
a_i	Weight coefficient for i^{th} correlator pair
N_{BOC}	BOC modulation order

1. Introduction

1.1 Background and motivation

During the second half of the last century, Global Navigation Satellite Systems (GNSSs) became an important part of wireless communication. It is widely used in personal devices, public transportation and industries. It can point out the exact location of any user on the surface of the earth anytime anywhere. The first GNSS, Global Positioning System (GPS), was started by the United States (US) government in the 1980's. It was primarily developed for military application, but it has been widely used in civilian applications and it is currently the only fully operational GNSS. Galileo is another GNSS under development, which will be the future European satellite system. It is independent from GPS, but fully compatible with GPS for civilian use worldwide. It is going to provide higher accuracy, better availability and more services compared with GPS. Galileo is expected to be operational commercially by 2015.

In 2004, the European Union (EU) and the US got an agreement that establishing a common baseline signal modulation, named Binary Offset Carrier or BOC(1,1) modulation for modernized civil GPS signal on L1 band and Galileo Open Service (OS) on E1 band. The new BOC modulation reduces the interference level caused by the existing GPS L1 C/A signal, since it splits the power spectra away from the center frequency [1]. In 2007, as the result of close working relation, the EU and US announced that a new modulation type, Multiplexed BOC or MBOC, as the common GPS-Galileo modulation for civilian use. This new modulation allocates a wide band signal BOC(6,1) in E1/L1 band without interfering with other existing signals and realizes the compatibility and interoperability between GPS and Galileo [2]. The Galileo E1 OS signal modulation, Composite BOC or CBOC is a variant of MBOC modulation. CBOC(6,1,1/11) is formed by a wideband signal, BOC(6,1) and a narrow-band signal, BOC(1,1), in such a way that it consists of 1/11 of the BOC(6,1) power and 10/11 of the BOC(1,1) power. Galileo satellites will transmit the modulated signal to the GNSS receivers by making use of Code Division Multiple Access (CDMA) technique.

After the signal is transmitted from the satellites and it propagates through the space, the synchronization of the received signal is done in Digital Signal Processing (DSP) part of a GNSS receiver. Code acquisition and tracking are two main DSP functions in a GNSS receiver that play crucial role in the accuracy of the position determination. In acquisition, it searches the presence of signals from satellites, producing a coarse estimation on frequency and time of the detected signal. After the signal is acquired, the tracking stage performs a fine estimation of carrier frequency and code delay of the

present satellite signals. The estimation of Line-Of-Sight (LOS) code delay is used to calculate the pseudorange and consequently the position of the receiver. The accuracy of the final value of code delay is therefore strictly related to the accuracy of a receiver's position calculation.
One typical code tracking structure is a code tracking loop, which continuously shifts the reference code generated within the receiver to align the received signal until the alignment is achieved. The shifting is determined by the estimated code delay from the code delay discriminator. Therefore, the algorithms used in the code delay discriminator will affect the accuracy of code delay estimation. Several code tracking algorithms (or delay trackers), such as in [3], [4] and [5], have been proposed in the literature for the BOC(1,1) modulated signal both for GPS and Galileo. However, the performance assessment of these typical delay trackers for MBOC modulated signals has not been studied much. Therefore, the researchers need to test these algorithms and evaluate the performance of a receiver with this new modulation type. The presence of pilot channel in Galileo E1 band brings the possibility to track either the data channel or the pilot channel or the combined data/pilot channel. Moreover, from the receiver point of view, the locally generated reference code in a receiver for the MBOC modulation also has more choices, which can be either MBOC modulated reference code or the BOC(1,1) modulated reference code. The effects of those possibilities and choices are also needed to evaluate in order to optimize the tracking performance.

One method for testing delay trackers and effects from different tracking algorithms can be realized in the simulation tool, such as software receivers. The simulation tool makes it convenient to test various algorithms used in the receiver and to investigate the receiver performance before the algorithms are built in the real devices. On the other hand, the implementation of algorithms in a software receiver can be also helpful on further developing the software receiver.

1.2 Thesis objectives

This thesis work has been carried out within two research projects of the Department of Communications Engineering (DCE) at Tampere University of Technology: the Academy of Finland-funded project "Digital Processing Algorithms for Indoor Positioning Systems (ACAPO)" and the EU FP7-funded project under grant agreement number 227890 "Galileo Ready Advanced Mass MArket Receiver (GRAMMAR) ". One common target in both projects is to find efficient code tracking algorithms for satellite receivers, which are able to operate also in adverse channel conditions (e.g., low Carrier-to-Noise ratios and in the presence of multipath).

The aim of this thesis has been to develop and analyze the implementation of various code tracking algorithms for Galileo E1 signal via Simulink tool. The analysis results have also helped to further develop the model in order to include realistic receiver constraints for mass-market application.

1.3 Thesis contributions

The main contributions of this thesis are enumerated below:

- The development of Simulink-based software receiver at TUT and implementation of various delay trackers in the software receiver.
- The optimization of a particular delay tracking structure, namely Multiple Gate Delay for Galileo E1 signals (i.e., using MBOC modulation).
- The derivation and implementation of a two-stage delay estimator based on Narrow Correlator and High Resolution Correlator.
- Detailed analysis of delay trackers with Simulink model in terms of Root Mean Square Error (RMSE) and tracking error variance.
- The optimization of code tracking loop bandwidth and evaluation of data/pilot tracking.
- Code tracking performance comparison between SinBOC(1,1) tracking and CBOC tracking.
- The partial study of bandwidth limitation on code delay tracking algorithms.

1.4 Thesis outline

The thesis is structured in the following manner:

Chapter 2 gives an overview of GNSSs and GNSS receiver operation.

Chapter 3 discusses the concept of BOC and MBOC modulations for Galileo E1 signal and a brief description of Galileo E5 signal.

Chapter 4 presents typical delay trackers: Narrow Correlator, High Resolution Correlator, Multiple Gate Delay, Dot Product discriminator, Slope Based Multipath Estimation and two-stage estimator, and the effects of normalization factors as well.

Chapter 5 shows the Multiple Gate Delay optimization for MBOC modulated signals with infinite and limited receiver front-end bandwidth.

Chapter 6 discusses about several GNSS software-defined receiver simulators available for commercial and academic use, in order to justify the development of a Simulink-based Galileo signal simulator at TUT.

Chapter 7 shows the GNSS Simulink software receiver at TUT, which is used for simulations in the thesis.

Chapter 8 describes own developments on the Simulink software receiver at TUT.

Chapter 9 presents the simulation results together with the performance analysis.

Finally, **Chapter 10** draws conclusions from this thesis work and presents suggestions for future works.

2. GNSS overview

Since the second half of the last century, when the US and the former Union of Soviet Socialist Republic (USSR) have introduced the concept of satellite-based positioning; the satellite navigation systems have become an important part of wireless communication technology. Several similar systems and satellite-based augmentation are being developed now. From 1980's, the US has started Global Positioning System (GPS). At the same time, the former USSR developed a GPS-like system named GLobal Orbiting NAvigation Satellite System (GLONASS). Also two stand-alone satellite navigation systems are developed nowadays, such as the future European satellite system, Galileo and Compass (former Beidou) system in China. In addition, both Japan and the European Space Agency (ESA) are working on GPS augmentation systems, such as Multifunction Transport Satellite Space based Augmentation System (MTSAT), European Geostationary Navigation Overlay System (EGNOS). A generic name given to these systems is *Global Navigation Satellite Systems* (GNSS). The goal in this chapter is simply to offer a brief overview of GNSSs. The next section is a brief introduction of the non-European GNSSs. Section 2.3 and 2.4 are focusing on the current status of the European GNSS, Galileo system and its physical layer characteristics. The operation of GNSS receivers will be discussed in Section 2.5.

2.1 Non-European GNSS

Global Position System (GPS) is the first and currently only fully operational navigation system. Although GPS was primarily developed for military purposes, it has been widely used in civilian applications as well during past few decades. However, GPS performance still needs further improvement for applications like surveying, geodesy, monitoring and automated machine control, which always demand more accuracy [6]. In the late 1990s, the US government started GPS modernization program, which will upgrade GPS performance for both military and civilian applications [7].

When GPS was under development, the former USSR developed a similar system called GLONASS. Like GPS, GLONASS was designed primarily for the military application. However, GLONASS has suffered from lack of resources in the changed political and economic climate and has been only recently updated to an almost full constellation. The user group and receiver manufactures are smaller than that based on GPS. [8]

China is also developing a regional satellite navigation system called *BeiDou*, which is currently extended to its global counterpart, *Compass.* It is designed to provide

positioning, fleet-management and precision-time dissemination to Chinese military and civilian users [7]. Unlike GPS interacts with user, in *Compass* system, the Mission Control Center determines a position estimate and transmits it to each user.

Japanese government is also undertaking development of a navigation system, called Quasi-Zenith Satellites System (QZSS), which is a regional system able to transmit ranging signals over Japan from satellites and to transmit differential correction to GPS and other GNSS systems.

2.2 European GNSS—Galileo

A new promising GNSS under development is the future European navigation system, Galileo, which is the focus of this thesis. Galileo is meant to interoperate with US GPS and Russian GLONASS, the two other global satellite navigation systems currently operational. Two test satellites Galileo In-Orbit Validation Element-A (GIOVE-A) and Galileo In-Orbit Validation Element-B (GIOVE-B) have been launched in 2005 and 2008, respectively. The Galileo system is expected to be fully operational by 2015.

Galileo will consist of 30 (27+3) satellites, positioned in three circular Medium Earth Orbit (MEO) planes at 23222 km altitude [9]. There will be a spare satellite in each plane [8].

Galileo will provide worldwide services depending on user needs [10]:

- The *Open Service* (OS) is designed for mass-market. It will be free of user charge.
- The *Safety-of-Life-Service* (SoL) is designed for use in most transport application where the degraded navigation information will endanger lives.
- The *Commercial Service* (CS), whose targets is the markets where more accuracy is required than offered by the OS. It uses two additional signals.
- The *Public Regulated Service* (PRS) is intended for groups such as police and customs. It is encrypted and operational at all times and circumstances.
- The *Search And Rescue Service* (SAR) is to be used for worldwide humanitarian search and rescue.

2.3 Physical layer characteristics of Galileo system

In Galileo system, four different frequency bands are assigned in order to transmit the navigation signals. These four frequency bands are: E5a and E5b with carrier frequencies at 1176.45 MHz and 1207.14 MHz, respectively, E6 frequency band with carrier frequency 1278.75 MHz, and E1 band with carrier frequency 1575.42 MHz as shown in Figure 2.1. [9]

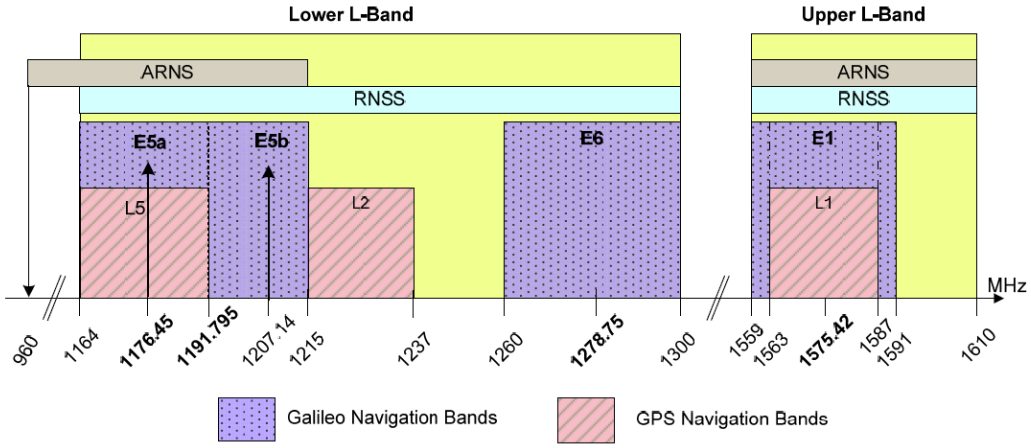

Figure 2.1: Galileo and GPS frequency plan [9]

The Galileo frequency bands have been selected in the allocated spectrum for Radio Navigation Satellite Services (RNSS) and E5a, E5b and E1 bands are included in the allocated spectrum for Aeronautical Radio Navigation Services (ARNS), employed by Civil-aviation users and allowing dedicated safety-critical application [9].

All the Galileo satellites will share the same nominal frequency, making use of Code Division Multiple Access (CDMA) compatible with the GPS approach. Six signals, including three data-less channels, or the so-called pilot channel (i.e., ranging codes not modulated by data), will be accessible to all Galileo users on the E5a, E5b and E1 carrier frequencies for OS and SoL services. Two signals on E6 with encrypted ranging codes, including one data-less channel will be accessible only to some dedicated users that gain access through a given CS provider. Finally, two signals (one in E6 band and on in E2-L1-E1 band) with encrypted ranging codes and data are accessible to authorized users of PRS. [11]

Galileo satellite transmits six different navigation signals: L1F, L1P, E6C, E6P, E5a and E5b signals. L1F signal (open access) and L1P signal (restricted access) are transmitted in the E1 band. E6C signal is a commercial access signal transmitted in E6 and E6P signal is a restricted access signal transmitted in E6A signal channel. E5a and E5b are open access signals transmitted in the E5 band. [12]

The receiver reference bandwidths centered on the carrier frequencies are specified in Table 2.1 [12]. Those reference bandwidths take into account the correlation losses. Table 2.2 and Table 2.3 show the carrier frequencies and signal definition of Galileo and GPS signals, respectively.

Table 2.1: Galileo signal receiver reference bandwidths [9]

Signal	**Receiver reference bandwidth (MHz)**
E1	24.552
E6	40.920
E5a	20.46
E5b	20.46

Table 2.2: Galileo signals and applied modulations [2], [9]

Band	Carrier Frequency [MHz]	Modulation	Chip rate [Mcps]	Code length (chips)	Data rate [sps]	Presence of pilot channel
E5a	1176.45	AltBOC(15,10)	10.23	10230	50	yes
E5b	1207.14			10230	250	yes
E6	1278.75	BPSK	5.115	5115	1000	yes
E1	1575.42	CBOC(+) in E1B CBOC(-) in E1C	1.023	4092	250 in E1B No data in E1C	yes

Table 2.3: GPS signal and applied modulations (Civil use only) [13]

Band	Carrier Frequency [MHz]	Modulation	Chip rate [Mcps]	Code length (chips)	Data rate [sps]	Presence of pilot channel
L1 C/A	1575.42	BPSK	1.023	1023	50	yes
L1C	1575.42	TMBOC	1.023	10230	100	yes
L2C	1227.60	BPSK	1.023	10230	50	yes
L5	1176.45	BPSK	10.23	10230	100	no

It can be ssen from the Table 2.2 and Table 2.3 that, the E1 band in Galileo and L1 band in GPS have the same center frequency at 1575.42 MHz, but the signal transmitted in E1 and L1 band do not interfere significantly with each other because of the use of difference modulations, as shown in the tables. Galileo introduces longer codes and new types of modulation. For several years, SinBOC(1,1) modulation has been the baseline for Galileo OS signal and modernized GPS L1C signal. Recently, GPS and Galileo working group has recommended MBOC modulation. As one of the MBOC implementations, CBOC will be used by Galileo OS and another MBOC implementations, TMBOC will be used by GPS for its L1C signal [2]. Both SinBOC(1,1) and MBOC modulations are described in Chapter 3. The Pseudo Random Noise (PRN) code sequences used for the Galileo navigation signals determine important properties of the system. The Galileo code design includes the code length and its correlation properties of the code sequence. The performance of Galileo codes is also dependent on the cold start acquisition time [11]. The code length of Galileo E1 band OS signal is 4092 chips, which is four times higher than the GPS C/A code length

of 1023 chips. For E5 signals, the code length is decided to be as high as 10230 chips [9].

For Galileo bands, the following chip rates are considered [9]:

- 10.23 Mcps for E5 band
- 5.115 Mcps for E6 band
- 1.023 Mcps for E1 band

As channel coding, a ½ rate convolutional coding scheme with constraint length 7 is used for all transmitted signals. There are different navigation messages transmitted in different bands, with the symbol rate at 50, 250, and 1000 symbols per second (sps). In GPS, the possible symbol rates are 50 and 100 sps. [14]

2.4 GNSS receiver operation overview

After the signal is transmitted from a satellite and it propagates through space, it is incident on a user's antenna of GNSS receiver. The radio front-end utilizes a combination of amplifier(s), mixer(s), filter(s), and its own oscillator to digitalize the incoming signal. The resulting sampled data will be used in signal processing to determine the position of the receiver. [15]

Figure 2.2 shows the simplified block diagram of a GNSS receiver. Passing through the radio front-end, the sampled data enters into signal processing stage. Acquisition and tracking are the two main tasks in signal processing stage. The purpose of acquisition is to identify all the satellites visible to user. If a satellite is visible, the acquisition must determine the frequency of the signal from a specific satellite, and the code phase, which denotes where the code begins in the current data block. The main purpose of tracking is to refine the coarse value of code phase and frequency and to keep track of these as the signal properties change over time. Code tracking and carrier frequency tracking are done here. The accuracy of the final value of the code phase is connected with the accuracy of pseudorange computed later on. If the receiver loses track of a satellite, a new acquisition must be performed for that particular satellite. [15]

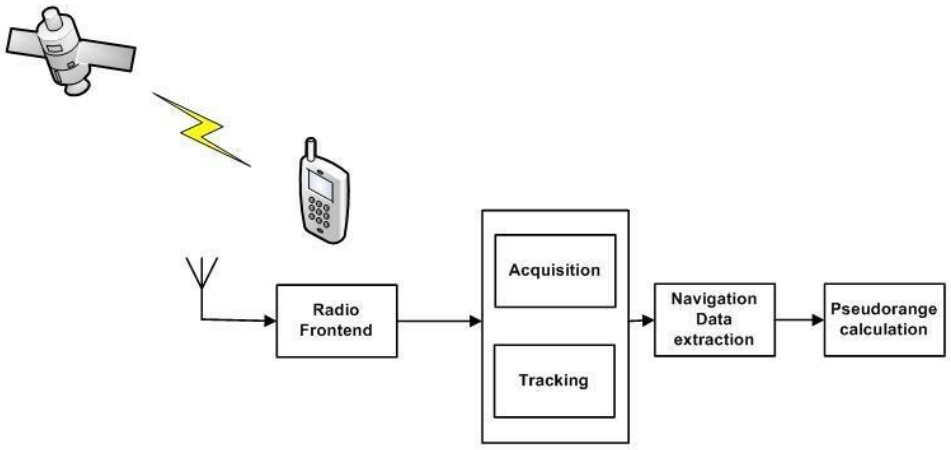

Figure 2.2: Example of transmitter and receiver channel for GNSS [15]

When the signal is properly tracked, the code and carrier will be removed from the signal, only leaving the navigation data bit. The navigation data bit is used to find when the data was transmitted from the satellite and which is the satellite positions on the sky at a given time (i.e., to extract the almanac or coarse data and ephemeris or accurate data about the satellites' positions). Another step before computing the position is to compute the pseudoranges. The pseudoranges are determined based on the time of transmission from the satellites and the time of arrival at the receiver. The time of arrival is based on the beginning of a subframe in the navigation data bit.

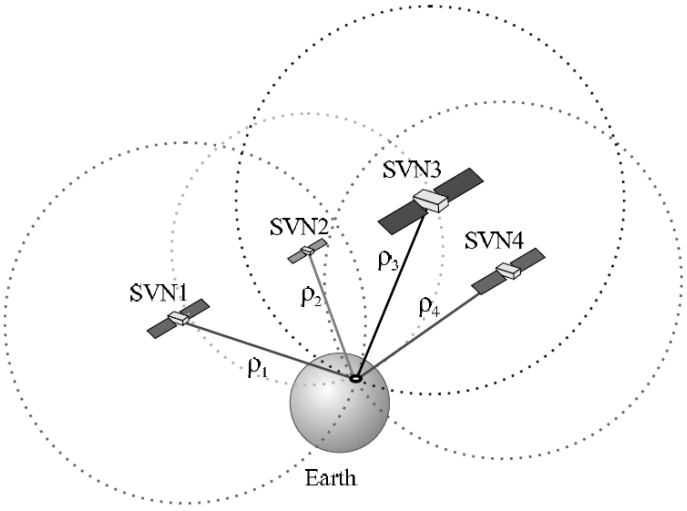

Figure 2.3: The basic principle of GNSS positioning. With known position of four Satellites SVNi and the signal travel distance ρ_i, the user position can be computed [15]

The final task is to compute the position of a user. Position determination is based on so-called triangulation principle as shown in Figure 2.3. It means that the position of a user is found at the intersection of 4 spheres, each with radius equal to the pseudorange measurement ρ_i.

The accuracy of pseudorange computation directly affects the accuracy of a user's position. The pseudorange is computed as the travel time from satellite to receiver multiplied by the speed of light in vacuum. The receiver has to estimate exactly when the signal is received. Therefore, an accurate estimation of the code phase in the tracking stage is very important, which is also one of the main topic addressed in this thesis.

3. GNSS modulation types

In 2004, the United States of America and the European Community reached an agreement that BOC(1,1) is to be the baseline for Galileo E1 OS signals and modernized GPS L1C signals. However, the optimization of that modulation has not stopped. The experts from US and Europe have produced a more recent recommendation for L1C and Galileo E1 OS signal, which is MBOC(6,1,1/11). The basic concept of BOC and MBOC will be introduced in this chapter.

3.1 Galileo E1 OS and GPS L1C signal modulation

In Galileo E1 band, the signals have the same carrier frequency as GPS L1 band. Therefore, a new type of modulation, namely CBOC or MBOC (in fact CBOC is an implementation of MBOC, as discussed later in this chapter), is used in order to minimize interference from GPS L1 signal. The concept of this new modulation will be introduced in the following sections.

3.1.1 BOC and MBOC

The concept of Binary Offset Carrier (BOC) modulation was first introduced by Betz as an effort for GPS modernization. It provides a simple and efficient way of shifting signal energy away from the band center. BOC modulation is a square sub-carrier modulation, where a signal is multiplied by a rectangular sub-carrier of frequency f_{SC}, which splits the spectrum of the signal into two parts. [16]

A BOC modulation is defined via two parameters BOC(m,n), related to reference frequency 1.023MHz, m= f_{SC} /1.023 and n= f_C /1.023, where f_C is chip rate [16]. From the point of view of the equivalent baseband signal, the BOC modulation can be defined via a single parameter, denoted as the BOC modulation order:

$$N_{BOC} \triangleq 2\frac{m}{n} = 2\frac{f_{SC}}{f_C} \tag{1}$$

where m and n should be chosen in such a way that N_{BOC} remains an integer.

BOC modulation has two main variants: sine-BOC (SinBOC) and cosine-BOC (CosBOC). SinBOC modulated signal $x(t)$ can be seen as the convolution between SinBOC waveform $S_{SinBOC}(t)$ and a modulating waveform $d(t)$, as [17]:

$$x(t) = \sum_{n=-\infty}^{\infty} b_n \sum_{k=1}^{S_F} c_{k,n} s_{SinBOC}(t - nT_{sym} - kT_C)$$
$$= s_{SinBOC}(t) \otimes \sum_{n=-\infty}^{\infty} \sum_{k=1}^{S_F} b_n c_{k,n} \delta(t - nT_{sym} - kT_C) \triangleq s_{SinBOC}(t) \otimes d(t) \quad (2)$$

where $\otimes$ is the convolution operator; $d(t)$ is the spread data sequence; b_n is the n^{th} complex data symbol; T_{sym} is the symbol period; $c_{k,n}$ is the k^{th} chip corresponding to the n^{th} symbol; $T_C = 1 / f_C$ is the chip period; S_F is the spreading factor ($S_F = T_{sym} / T_C = 1023$ for Galileo E1 OS and GPS L1 signals), and $\delta(t)$ is the discrete Dirac pulse, which is has the value of infinity for $t = 0$, the value zero elsewhere [18].

According to its original definition in [16], the SinBOC signal waveform $s_{SinBOC}(t)$ is defined as:

$$s_{SinBOC}(t) = sign\left(\sin\left(\frac{N_{BOC}\pi t}{T_C}\right)\right), 0 \le t \le T_C \quad (3)$$

where $sign(\bullet)$ is the signum operator. Figure 3.1 gives an example of time domain waveform for SinBOC(1,1) modulated chip sequence with 5 chips [1, -1, 1, 1, 1, 1].

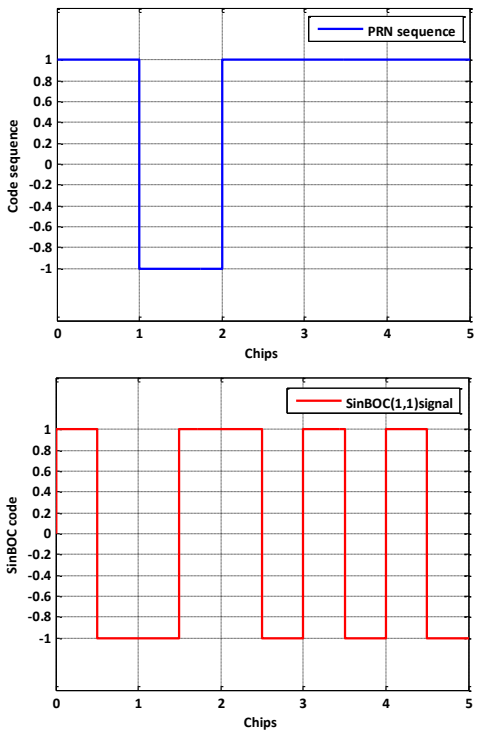

Figure 3.1: Example of SinBOC(1,1) modulated signal in time domain (lower plot). The upper plot shows the original PRN sequence before modulation

Similarly, the CosBOC-modulated signal is the convolution between the modulating signal and the following waveform [16]:

$$s_{CosBOC}(t) = sign\left(\cos\left(\frac{N_{BOC}\pi t}{T_C}\right)\right), 0 \le t \le T_C \tag{4}$$

For Public Regulated Service (PRS) in Galileo, CosBOC(15,2.5) proposed in [19] performs better with respect to multipath mitigation and seems to remain the most likely candidate. However, CosBOC(15,2.5) modulation is out of the scope of this thesis.

The normalized Power Spectral Density (PSD) of a SinBOC(m,n) modulated PRN code with even N_{BOC} is given by [16], [17]:

$$G_{SinBOC(m,n)}(f) = \frac{1}{T_C}\left(\frac{\sin\left(\pi f \frac{T_C}{N_{BOC}}\right)\sin(\pi f T_C)}{\pi f \cos\left(\pi f \frac{T_C}{N_{BOC}}\right)}\right) \tag{5}$$

An example of PSD of SinBOC(1,1) is presented in Figure 3.2. It can be seen that the power is away from the center frequency.

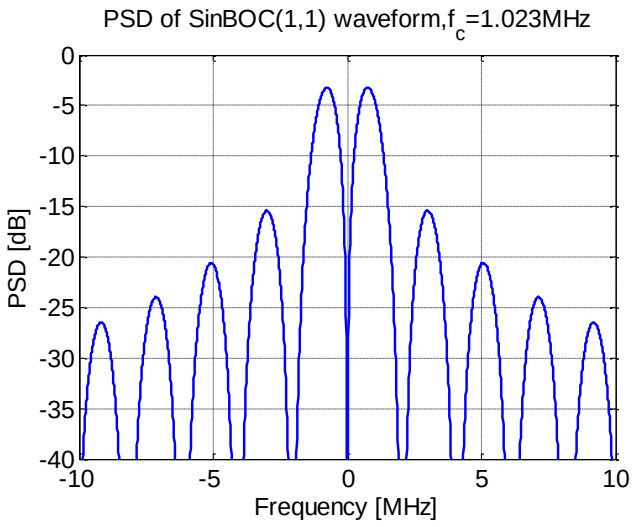

Figure 3.2: Example of PSD of SinBOC(1,1) modulated signal

Multiplexed BOC (MBOC) introduces more power on higher frequencies compared with SinBOC(1,1) case, by adding a high frequency BOC component, which improves the performance in tracking [1]. The PSD of MBOC (6,1,1/11) is the sum of weighted PSD of BOC (1,1) and BOC (6,1). The PSD of MBOC(6,1,1/11) is shown to be:

$$G_{MBOC}(f) = \frac{10}{11} G_{SinBOC(1,1)}(f) + \frac{1}{11} G_{SinBOC(6,1)}(f) \tag{6}$$

where $G_{SinBOC(m,n)}$ is the unit-power PSD of a sine-phased BOC modulation [1]. The PSD of MBOC(6,1,1/11) is shown in Figure 3.3.

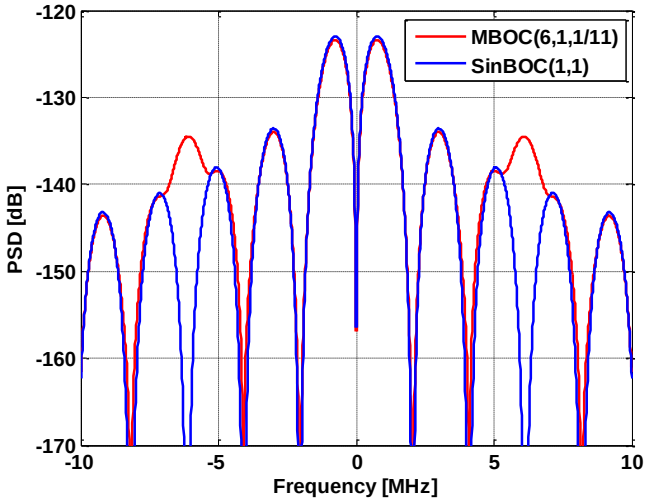

Figure 3.3: Example of PSD of MBOC(6,1,1/11) modulated signal

Since the definition of MBOC is in frequency domain, different implementations in time domain will fit into the definition above. Two main implementations of MBOC modulation are Composite BOC (CBOC) and Time-Multiplexed BOC (TMBOC) [1], [9].

3.1.2 MBOC implementation—TMBOC

TMBOC is the main candidate for the modernized GPS L1C signal. In TMBOC, the whole signal is divided into block of N code symbols and $M<N$ of N code symbols are SinBOC(1,1) modulated, while N-M code symbols are SinBOC(6,1) modulated. The waveform of TMBOC can be written as [20]:

$$s_{TMBOC}(t) = \sqrt{E_b}\sum_{n\in S} b_n \sum_{m=1}^{S_F} c_{m,n} \sum_{i=0}^{N_{BOC1}} \sum_{k=0}^{\frac{N_{BOC2}}{N_{BOC1}}-1} (-1)^i P_{T_{BOC2}}\left(t - i\frac{T_C}{N_{BOC1}} - k\frac{T_C}{N_{BOC2}}\right) + \sqrt{E_b}\sum_{n\notin S} b_n \sum_{m=1}^{S_F} c_{m,n} \sum_{i=0}^{N_{BOC2}-1} (-1)^i P_{T_{BOC2}}\left(t - i\frac{T_C}{N_{BOC2}}\right) \quad (7)$$

where N_{BOC1}=2 is the BOC modulation order for SinBOC(1,1), N_{BOC2}=12 is the BOC modulation order for SinBOC(6,1); S is the set of chips which are SinBOC(1,1) modulated; E_b is the code symbol energy; b_n is the n^{th} code symbol; $c_{m,n}$ is the m^{th} chip corresponding to the n^{th} symbol; $P_{T_{BOC2}}(\bullet)$ is a rectangular pulse of support T_C / N_{BOC2} and unit amplitude. An example of time domain waveform of TMBOC is shown in Figure 3.4.

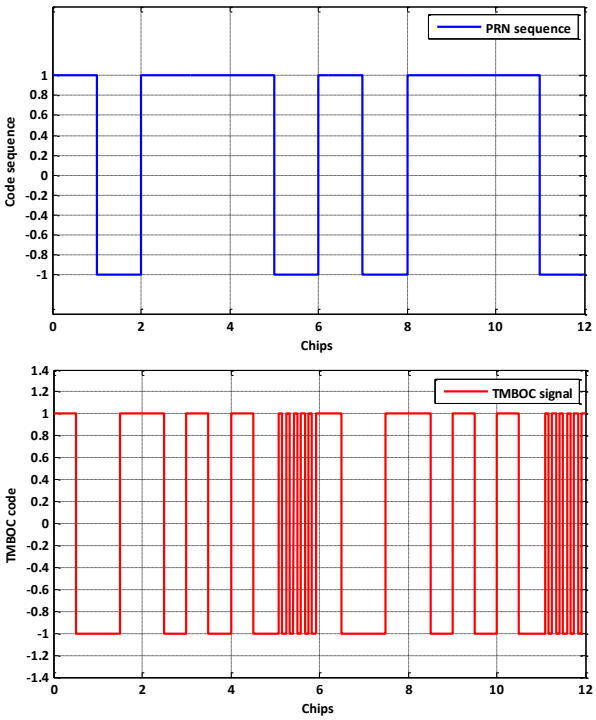

Figure 3.4: Example of waveform of TMBOC in time domain. Upper plot: PRN sequence; Lower plot: TMBOC modulated waveform

Since the pilot and data components of a signal can be formed using different spreading time series and the total signal power can be divided differently between the pilot and data components, many different TMBOC-based implementations are possible. One candidate implementation of TMBOC for a signal with 75% power on the pilot component and 25% power on the data component, could use all SinBOC(1,1) spreading symbols on the data component and 29/33 SinBOC(1,1) spreading symbols and 4/33 SinBOC(6,1) spreading symbols on the pilot component as in Equation (8) [1].

$$
\begin{aligned}
G_{Pilot}(f) &= \frac{29}{33} G_{SinBOC(1,1)}(f) + \frac{4}{33} G_{SinBOC(6,1)}(f) \\
G_{Data}(f) &= G_{SinBOC(1,1)}(f) \\
G_{MBOC(6,1,1/11)}(f) &= \frac{3}{4} G_{Pilot}(f) + \frac{1}{4} G_{Data}(f) \\
&= \frac{10}{11} G_{SinBOC(1,1)}(f) + \frac{1}{11} G_{SinBOC(6,1)}(f)
\end{aligned}
\tag{8}
$$

Another candidate for TMBOC implementation is using all SinBOC(1,1) spreading symbols on the data component, and 2/11 SinBOC(6,1) spreading symbols on the pilot component. The power split between the data and pilot component in a signal is half as in Equation (9). [1]

$$
\begin{aligned}
G_{Pilot}(f) &= \frac{9}{11} G_{SinBOC(1,1)}(f) + \frac{2}{11} G_{SinBOC(6,1)}(f) \\
G_{Data}(f) &= G_{SinBOC(1,1)}(f) \\
G_{MBOC(6,1,1/11)}(f) &= \frac{1}{2} G_{Pilot}(f) + \frac{1}{2} G_{Data}(f) \\
&= \frac{10}{11} G_{SinBOC(1,1)}(f) + \frac{1}{11} G_{SinBOC(6,1)}(f)
\end{aligned}
\tag{9}
$$

Receiver implementation will be the simplest if SinBOC(6,1) symbols are placed in the same location in both pilot and data components. Proper placement can improve the autocorrelation and cross correlation properties of spreading code, compared to these properties with all SinBOC(1,1) spreading symbols [1].

3.1.3 MBOC implementation—CBOC

A possible CBOC implementation is based on the four level spreading symbols formed by weighted sum of SinBOC(1,1) and SinBOC(6,1) symbols. Two different implementation of CBOC can be considered for a fifty-fifty power split between data and pilot components. [21]

- CBOC symbols are used in both data and pilot components, formed from the sum of $\sqrt{10/11}$ SinBOC (1, 1) symbols and $\sqrt{1/11}$ SinBOC (6, 1) symbols.
- CBOC symbols are used only on pilot component, formed from the sum of $\sqrt{9/11}$ SinBOC (1, 1) and $\sqrt{2/11}$ SinBOC (6, 1).
-

According to [21], there are three signal models that can be used to implement CBOC:

- CBOC(‘+’)
- CBOC(‘-’)
- CBOC(‘+/-’)

The examples of CBOC(‘+’), CBOC(‘-’) and CBOC(‘+/-’) time domain waveforms along with the original PRN code sequence are presented in Figure 3.5.

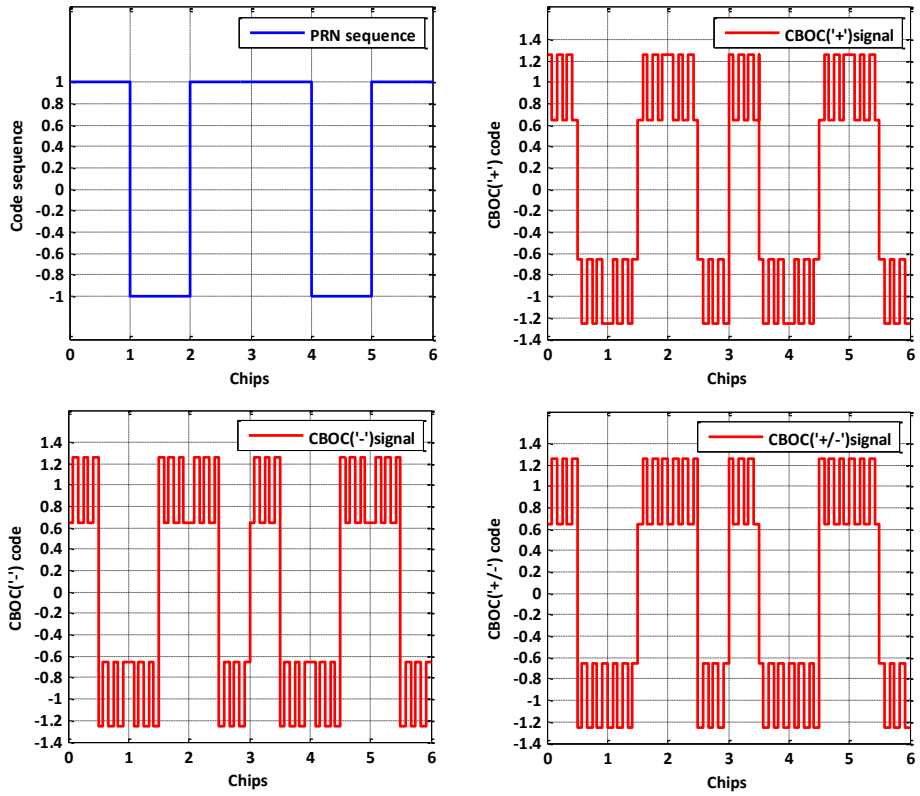

Figure 3.5: Example of CBOC ($w_1 = \sqrt{10/11}$) waveform in time domain

In CBOC(‘+’) modulation, the weighted SinBOC(1,1) modulated symbol is summed by weighted SinBOC(6,1) modulated symbol [21].

$$s_{CBOC('+')}(t) = w_1 s_{SinBOC(1,1)}(t) + w_2 s_{SinBOC(6,1)}(t) \tag{10}$$

where w_1 and w_2 are amplitude weighting factors which need to be chosen in such a way that PSD is as in Equation (6) and $w_1^2+w_2^2=1$. One possible choice is to select $w_1 = \sqrt{10/11}$ and $w_2 = \sqrt{1/11}$, currently used in the standard [9].

In CBOC(‘-’) modulation, the weighted SinBOC(6,1) modulated symbol is subtracted from the weighted SinBOC(1,1) modulated symbol [21]:

$$s_{CBOC('-')}(t) = w_1 s_{SinBOC(1,1)}(t) - w_2 s_{SinBOC(6,1)}(t) \tag{11}$$

In CBOC('+/-') modulation, the weighted SinBOC(1,1) is summed by the weighted SinBOC(6,1) modulated symbols for even chip and the weighted SinBOC(6,1) is subtracted from the weighted SinBOC(1,1) modulated symbols for odd chips [21].

$$s_{CBOC('+/-')}(t) = \begin{cases} w_1 s_{SinBOC(1,1)}(t) + w_2 s_{SinBOC(6,1)}(t) & \text{even chip} \\ w_1 s_{SinBOC(1,1)}(t) - w_2 s_{SinBOC(6,1)}(t) & \text{odd chip} \end{cases} \tag{12}$$

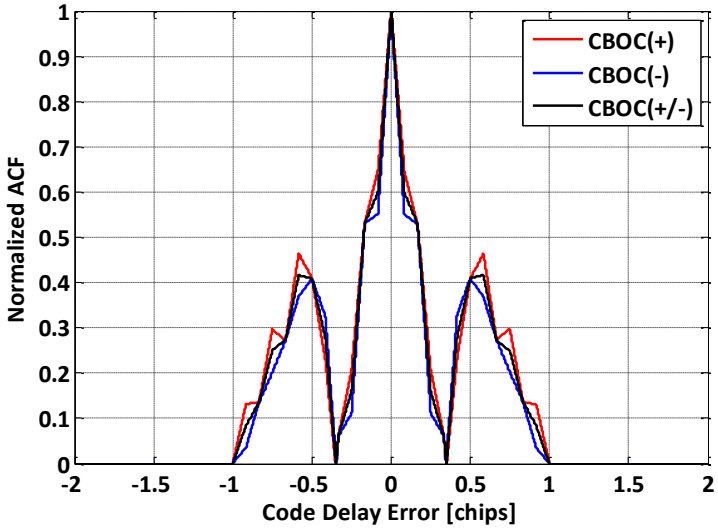

Figure 3.6: Normalized absolute ACFs of CBOC('+'), CBOC('-') and CBOC('+/-')

Figure 3.6 presents the normalized absolute Auto-Correlation Function (ACF) of different CBOC implementations with infinite receiver bandwidth. It can be observed that the main peak of CBOC('-') is narrower than the other implementations. The shape of ACF will affect the tracking performance. The secondary peaks can lead to stable false lock points [21].

Currently, CBOC('+') and CBOC('-') have been proposed for the E1B data channel and E1C pilot channel of Galileo E1 OS signal as described in [9]. This thesis is focusing on these two modulation type.

3.2 E5 signal

Galileo transmits four different signals in the E5 band. Two of them will carry navigation messages and the remaining two are data-free pilot channels.

Table 3.1: Signal properties of E5 band [22] [23]

Signal component	Modulation	Data	Chip rate [Mchip/s]	Center frequency
E5aI	BPSK(10)	Yes	10.23	1176.45MHz
E5aQ	BPSK(10)	No	10.23	
E5bI	BPSK(10)	Yes	10.23	1207.14 MHz
E5bQ	BPSK(10)	No	10.23	

These four signal components in the E5 band can be modulated as a wideband signal generated by AltBOC(15,10) 8-PSK modulation as described in [9] and [22]. The wideband signal is at center frequency of 1191.795 MHz [23]. The AltBOC modulation provides such advantage that E5a (I/Q) and E5b (I/Q) can be processed independently, as traditional BPSK(10) signal, or together, leading to a better tracking performances in terms of noise and multipath mitigation at the cost of a larger front-end bandwidth and increased complexity [23]. However, the study of E5 signals is beyond the scope of this thesis.

4. Code tracking loops

As mentioned in the Section 2.4, the tracking is needed in order to provide the fine code delay estimation and the accuracy of this estimation is directly related to the accuracy of users' position computation. In this chapter, a common tracking structure and some typical algorithms used in code tracking structure will be introduced.

4.1 Discriminators for code tracking loops

One common structure used in GNSS receiver for code tracking is based on a feedback loop. The received signal is correlated with an early and a late shifted locally generated reference code. The correlation outputs are then used in the discriminator function in order to detect the code phase difference between the received signal and the reference code. The output of discriminator function is fed into the Numerically Controlled Oscillator (NCO) in order to generate a precise reference code. Before feeding the output of a discriminator function into NCO, it passes through a loop filter, which is used to reduce noise in order to produce an accurate estimate for an original signal at its output [7]. The code tracking is maintained through a feedback loop where the error signal is formed by discriminator function. In the following sections, some typical discriminator functions and the impact of normalization of discriminator functions on the tracking performance are studied.

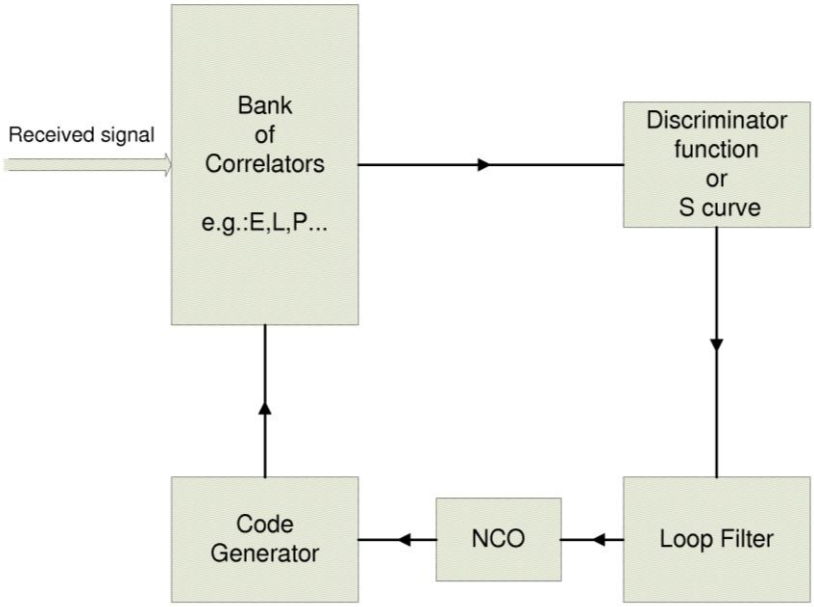

Figure 4.1: Generic block diagram of a code tracking loop

4.1.1 Narrow Correlator

The Narrow Correlator (narrow-EML or nEML) is one of the most popular multipath mitigation approaches. It is based on narrowing the large Early-Late spacing (e.g., $\Delta=xT_C$, $x<=1$ is the early-late spacing in chips) of a classical early-minus-late code tracking. This reduces the tracking errors in the presence of noise and multipath [3]. The nEML requires three complex correlators: one early, one late and one in-prompt. One complex correlator is equivalent to two real correlators; one is for In-phase (I) branch and one for Quadrature-phase (Q) branch. Its output has a characteristic shape, commonly referred to as S-curve or discriminator output, denoted by *D* in what follows. The correct code phase can be found in zero-crossing. The shape of S-curve also depends on the early-late spacing. There are several nEML implementations and the most common are the coherent nEML and the non-coherent early-minus-late power. The discriminator function for un-normalized absolute of early-minus-absolute of late, which will be used in what follows, is:

$$D = |I_E| + |Q_E| - |I_L| - |Q_L| \tag{13}$$

where I_E, I_L, Q_E and Q_L are the I and Q components for early and late correlators. An example of un-normalized nEML with 0.08 chip E-L spacing is shown in Figure 4.2. The normalization issue will be discussed in Section 4.2.

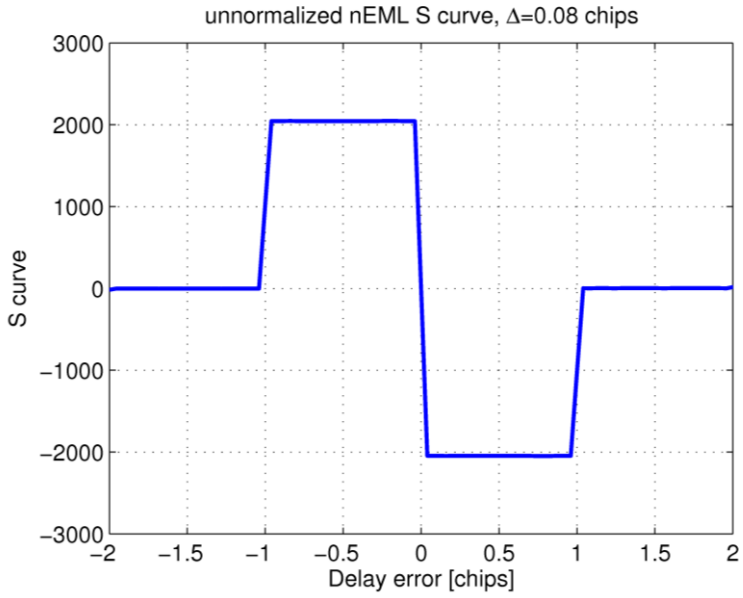

Figure 4.2: Example of S-curve for unnormalized in single path propagation and infinite receiver bandwidth

4.1.2 HRC

The High Resolution Correlator (HRC) was introduced in [5]. It has two more correlators compared with nEML. The unnormalized discriminator function of HRC with output D is presented as in [5]:

$$\begin{aligned} D &= a_1(|I_E|+|Q_E|-|I_L|-|Q_L|) \\ &\quad + a_2(|I_{VE}|+|Q_{VE}|-|I_{VL}|-|Q_{VL}|),\ a_1 = 1, a_2 = -0.5 \end{aligned} \tag{14}$$

here I_E, I_L, I_{VE}, I_{VL}, Q_E, Q_L, Q_{VE} and Q_{VL} are the I and Q components of Early (E), Late (L), Very Early (VE) and Very Late (VL) correlators. If the E-L correlator spacing is Δ, then VE-VL correlator spacing is 2Δ. As mentioned in [5], HRC provides significant multipath mitigation for medium and long-delay multipath compared with nEML, but it cannot reject the short delay multipath effects and suffers from significant degradation in noise performance. As we can see from Figure 4.3, the existence of extra zero crossing in S-curve increases the possibility of locking to a false point. Moreover, HRC is under patent protection [24].

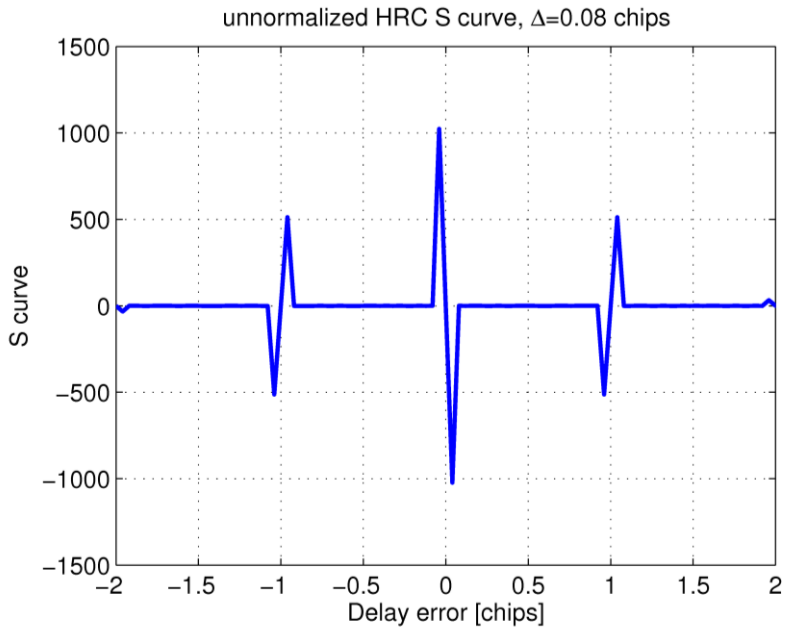

Figure 4.3: Example of S-curve for unnormalized HRC in single path propagation and infinite receiver bandwidth

4.1.3 MGD

Another code tracking discriminator function is called Multiple Gate Delay (MGD), which has been first introduced in [25]. It has a variable number of weighted early-late correlator pairs. The error output D of an unnormalized MGD discriminator with three pairs of absolute early-late correlators, which will be used in what follows, is given by:

$$\begin{aligned} D = {} & a_1(|I_E| + |Q_E| - |I_L| - |Q_L|) \\ & + a_2(|I_{VE}| + |Q_{VE}| - |I_{VL}| - |Q_{VL}|) \\ & + a_3(|I_{VVE}| + |Q_{VVE}| - |I_{VVL}| - |Q_{VVL}|) \end{aligned} \tag{15}$$

where a_1, a_2 and a_3 are the weight coefficients for each correlator pair. Usually, a_1 is normalized to 1 without loss of generality; I_E, I_L, I_{VE}, I_{VL}, I_{VVE}, I_{VVL}, Q_E, Q_L, Q_{VE}, Q_{VL}, Q_{VVE} and Q_{VVL} are the I and Q components of early (E), late (L), Very Early (VE), Very Late (VL), Very Very Early (VVE) and Very Very Late (VVL). In [25], the results showed that MGD performs significantly worse than nEML. The main reason for that is due to the fact that the weighting factors were not optimized. The results of MGD with optimum parameters in [4], [26] and [27] showed that the optimum MGD gives better performance than nEML and HRC under the infinite receiver front-end bandwidth. Nevertheless, the main advantage of MGD is that it offers a large set of unpatented choices, which can be used in the design of mass-market GPS or Galileo receivers [26]. In this thesis, the optimization for MGD structure for MBOC modulation is studied, which will be presented in Chapter 5.

4.1.4 Dot Product (DP) discriminator

The unnormalized Dot Product is defined as follows:

$$D = (I_E - I_L)I_P + (Q_E - Q_L)Q_P \tag{16}$$

where I_E, I_L, I_P, Q_E, Q_L and Q_P are the I and Q components for Early, Late and Prompt correlators. From the point of view of S-curve, the Dot Product discriminator has similar S-curve shape as the nEML but it slightly outperforms nEML [7]. Figure 4.4 presents an example of S-curve for unnormalized Dot Product discriminator.

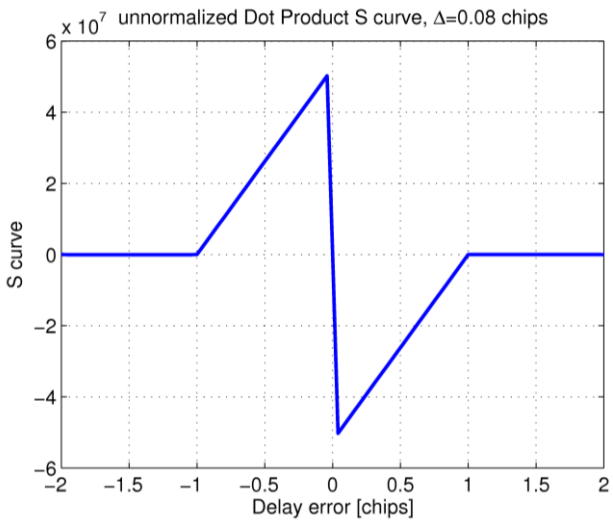

Figure 4.4: Example of S-curve for unnormalized Dot Product in single path propagation and infinite receiver bandwidth

4.1.5 SBME

An A-Posteriori Multipath Estimation (APME) was proposed in [28]. It utilizes a posteriori-estimation of the multipath error affecting the code tracking. The tracking is done in a conventional nEML. The multipath error affecting the nEML tracking is estimated in an independent module on the basis of different signal amplitude measurements. Subtracting this estimation from the code-phase measurement yields a substantial reduction of the error, especially for short-delay multipath. Therefore, a modified APME was developed in co-operation with colleagues at TUT and named as Slope Based Multipath Estimation (SBME) technique [29]. It uses nEML in tracking as in APME, but estimated the multipath error is calculated as in Equation (17):

$$MP = -a_{SBME}\left(1 - \frac{I_{+2} - m_L d}{I_0}\right) \tag{17}$$

here m_L is the late slope of the normalized ideal correlation function (i.e. m_L= -1 for BPSK, and m_L= -3 for SinBOC(1,1) modulated signal); d is the spacing between early and late correlator pair; I_0 and I_{+2} are the correlation values at prompt and at 2d late from the prompt correlation, respectively; a_{SBME} is the optimized coefficient in least square sense by utilizing the theoretical MEE curves (e.g., a_{SBME} is 0.42 in case of BPSK). Here, it is to be mentioned that, the late slope totally depends on the correlation shape. The parameters used in this thesis are summarized in Table 4.1. These parameters in the table were derived via Least Square optimization, in joint co-operation with colleagues at TUT.

Table 4.1: Parameters used in SBME

	SinBOC(1,1) tracking		MBOC tracking	
	m_L	a_{SBME}	m_L	a_{SBME}
E1B channel	-2.68377	0.2	-4.2297	0.07
E1C channel	-3.31623	0.14	-5.3847	0.05

4.1.6 Two-stage estimator

The two-stage estimator runs the first stage for certain time duration in order to tracking the error around the main peak of correlation shape. The nEML is used in the first stage because it has wide uncertainty region, which provides high possibility to track on the main peak of the correlation shape. The second stage is activated after the first stage is finished. The second stage is to make the fine estimation of the code delay. HRC is chosen in the second stage since it has smaller uncertainty region compared with nEML, which can provide more accurate code delay estimation than nEML.

As shown in Figure 4.3, the extra zero-crossing in the discriminator function of HRC increases the possibility of locking to a false point and sensitivity to noise. Therefore, a separate Carrier-to-Noise Ratio (CNR) estimator module is implemented, which is working with two stage estimator in order to improve the tracking performance.

The CNR estimator is based on the theory in [30]. It considers the measurement of total power in $1/T$ (wide-band power) and $1/MT$ (narrow-band power) noise bandwidth of the following form:

$$WBP_k = \left(\sum_{i=1}^{M} (I_i^2 + Q_i^2) \right)_k \tag{18}$$

and

$$NBP_k = \left(\sum_{i=1}^{M} I_i \right)_k^2 + \left(\sum_{i=1}^{M} Q_i \right)_k^2 \tag{19}$$

computed over the same M samples. A normalized power defined as follows:

$$NP_k = \frac{NBP_k}{WBP_k} \tag{20}$$

The CNR estimator can be presented as:

$$\frac{\hat{C}}{N} = 10 \log_{10} \left(\frac{1}{T} \frac{\hat{\mu}_{NP} - 1}{M - \hat{\mu}_{NP}} \right) \tag{21}$$

where $\hat{\mu}_{NP} = \frac{1}{K}\sum_{k=1}^{K} NP_k$ is the lock detector measurement.

The estimated CNR from CNR estimator is working between the first and second stage. If the estimated CNR is higher than a threshold, the second stage will be running with HRC. Otherwise, the tracking will run with nEML continuously. The CNR threshold can be set according to the users' requirement (for example, 33 dB-Hz used in this thesis).

4.2 Normalization factor of discriminators

All the code tracking discriminators should be normalized in order to compensate for the gain variances in the code tracking loop. Indeed, in the Simulink-based model, the tracking loops are not operating properly without normalization. The normalization removes the amplitude sensitivity, which helps the code tracking loop to keep track of noisy signals. Therefore, the normalization helps the code tracking performance. However, the question about the choice of a proper normalization factor is not a trivial one and we have addressed this issue in this thesis, as follows.

One way to normalize the discriminator function is by using the prompt correlator. In the ideal cross-correlation function between the incoming signal and local reference code, the prompt correlator output has the maximum amplitude or power. Therefore, it can be used for normalization. An example of nEML normalized by the prompt correlator is shown in Equation (22). Similar formula can be obtained for HRC and MGD by using the same denominator.

$$D = \frac{|I_E| + |Q_E| - |I_L| - |Q_L|}{|I_P| + |Q_P|} \tag{22}$$

Since in the Dot Product discriminator function, the correlators are multiplied by each other, the normalization factor should remain the same order as in the unnormalized discriminator function. Therefore, the power of the prompt correlator is used in Dot Product as shown in Equation (23).

$$D_{DP} = \frac{(I_E - I_L)I_P + (Q_E - Q_L)Q_P}{I_P^2 + Q_P^2} \tag{23}$$

Another normalization method is to use the sum of the signal strengths of early and late correlators as described in [15]. An example of nEML normalized by the sum of early and late correlators is shown in Equation (24).

$$D = \frac{|I_E| + |Q_E| - |I_L| - |Q_L|}{|I_E| + |Q_E| + |I_L| + |Q_L|} \tag{24}$$

Again, similar formula can be obtained for HRC and MGD by using the same denominator. The Dot Product again uses the power of the early and late correlator output as shown in Equation (25).

$$D_{DP} = \frac{(I_E - I_L)I_P + (Q_E - Q_L)Q_P}{I_E^2 + Q_E^2 + I_L^2 + Q_L^2} \tag{25}$$

The normalization by sum of the signal strength of early and late correlator output can be used in any DLL discriminator function. Figure 4.5-Figure 4.8 give examples about how the normalization performs in different CNR conditions. The envelope of the correlator output was used in the simulation and MGD parameters are optimized. As we can see, the normalization by early plus late correlator gives the best tracking performance for nEML, Dot Product and MGD, especially at lower CNR condition. The normalization results of HRC shown in Figure 4.7 are not so clear here. There is not much difference in tracking performance in case of HRC with these three types of normalization. However, in what follows, the normalization via early-plus-late correlation will be used for all the discriminators.

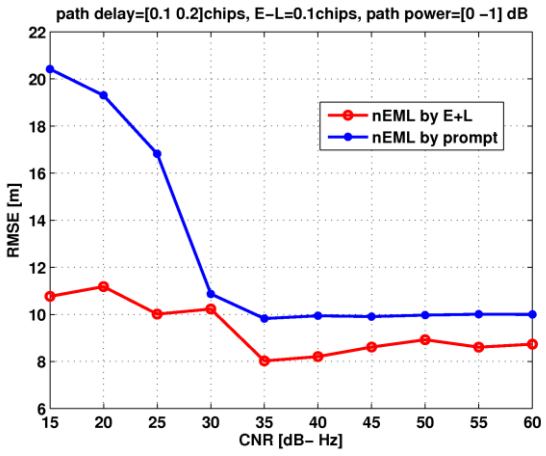

Figure 4.5: RMSE vs. CNR for nEML with two normalization factors

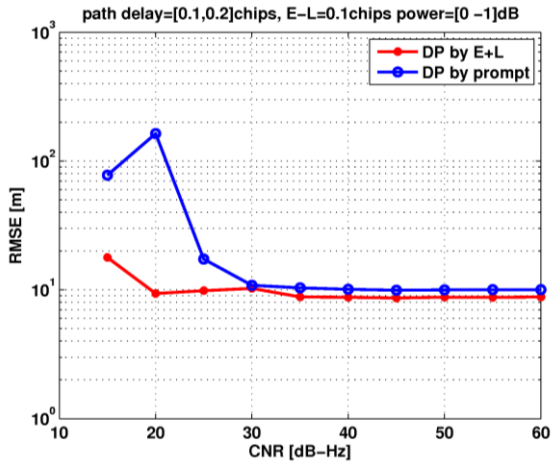

Figure 4.6: RMSE vs.CNR for Dot Product with two normalization factors

As described in the previous section, HRC and MGD algorithms utilize more correlator pairs than that of nEML. Therefore, there is another possibility to normalize HRC and MGD discriminators by using all the correlator pairs and corresponding weight factors in the normalization. A generic formula of the normalized discriminator function can be presented as follows:

$$D=\frac{\begin{array}{l}a_1(|I_E|+|Q_E|-|I_L|-|Q_L|)\\+a_2(|I_{VE}|+|Q_{VE}|-|I_{VL}|-|Q_{VL}|)\\+a_3(|I_{VVE}|+|Q_{VVE}|-|I_{VVL}|-|Q_{VVL}|)\end{array}}{\begin{array}{l}a_1(|I_E|+|Q_E|+|I_L|+|Q_L|)\\+a_2(|I_{VE}|+|Q_{VE}|+|I_{VL}|+|Q_{VL}|)\\+a_3(|I_{VVE}|+|Q_{VVE}|+|I_{VVL}|+|Q_{VVL}|)\end{array}} \tag{26}$$

where a_1, a_2 and a_3 are the weighting factors for each correlator, for MGD, they are optimized MGD coefficients; for HRC, a_1=1, a_2=-0.5 and a_3=0.

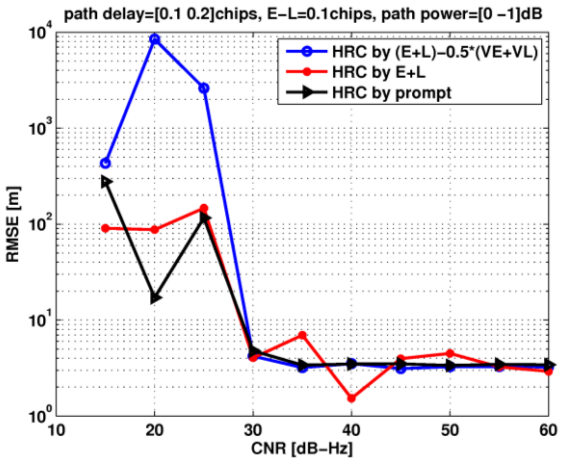

Figure 4.7: RMSE vs.CNR for HRC with three normalization factors

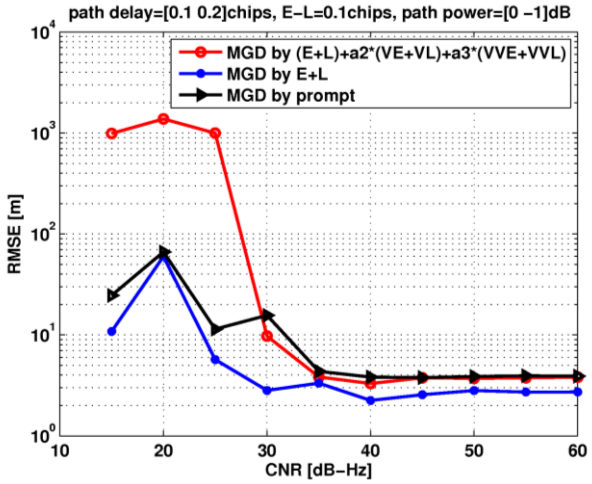

Figure 4.8: RMSE vs.CNR for MGD with three normalization factors

However, as can be seen from Figure 4.7 and Figure 4.8, this type of normalization gives the worst performance in tracking at low CNR.

5. MGD Optimization for MBOC

In this chapter, we discuss about the choice of optimum weighting factors for MGD delay tracking structure. First, the MGD structure is reviewed, then an optimality criterion is defined, and afterwards the optimization results are shown.

5.1 MGD structure

Bases on the studies in [4] and [26], the MGD structure was chosen as shown in Figure 5.1. The notation of E (Early), L (Late), VE (Very Early), VL (Very Late), VVE (Very Very Early) and VVL (Very Very Late) stand for E_i and L_i correlator pairs, i=1,2,3 respectively. The spacing between E_i and L_i are uniform $\Delta_i = i\Delta_1$, where Δ_1 is the E-L spacing.

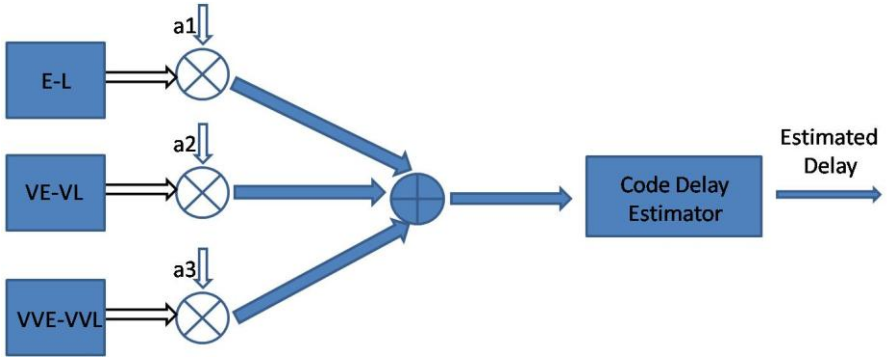

Figure 5.1: MGD discriminator function structure

In this thesis, the delay tracking loop structure shown in Figure 5.2 is used. The in-phase (R_I) and Quadrature (R_Q) correlators are generated and shifted according to the estimated delay from the discriminator. A NCO adjusts the code phase according to the smoothed error coming from the discriminator function. The smoothing is done via the loop filter, here using a code loop bandwidth of 3 Hz.

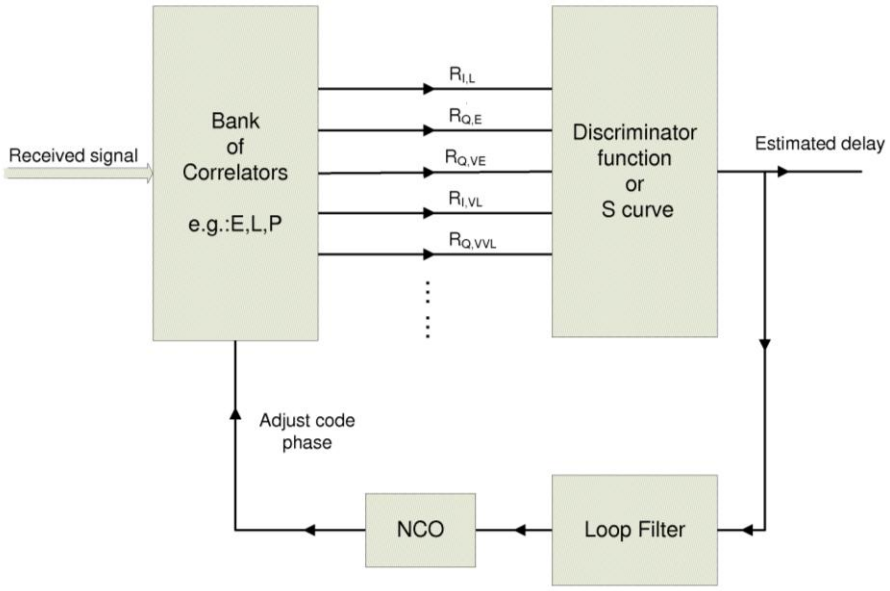

Figure 5.2: The block diagram of tracking loop used

5.2 Optimization criterion and theoretical analysis

The optimization is analyzed under two hypotheses: infinite bandwidth (mostly used in theoretical analysis) and limited front-end bandwidth (realistic case). With infinite front-end bandwidth, the optimum coefficients were obtained for five MBOC signal implementations, namely CBOC(+), CBOC(-), CBOC(+/-), TMBOC(6,1,9/11) and TMBOC(6,1,29/33). The reference code has the same modulation as the transmitted signal, because for infinite bandwidth, we can get better results with same reference code as transmitted signal and we are not limited by the bandwidth constraint. With limited front-end bandwidth, two receiver options are studied here: 1) the reference code uses the same modulation as the transmitted signal; 2) the reference code is SinBOC(1,1) modulated (this approach is typically useful under low bandwidth constraints, which are specific to mass-market receivers).

As shown in Figure 5.1, three correlator pairs are used, because the MGDs with more than three early-late correlator pairs proved to bring only a little benefit in performance and increased the complexity significantly [26]. In order to decide on the optimum coefficient, we used an optimality criterion based on so-called Multipath Error Envelope (MEE) [4], which shown the performance under a two-path channel, in the absence of other noise sources. The MEE is widely used for illustrating the multipath performance of different code tracking algorithms [27]. The smaller the enclosed area is, the better the multipath performance is. The MGD with optimum coefficients should have minimum enclosed area. The illustration of the enclosed MEE area principle is shown in Figure 5.3 for a nEML structure with 3 MHz double-sided bandwidth.

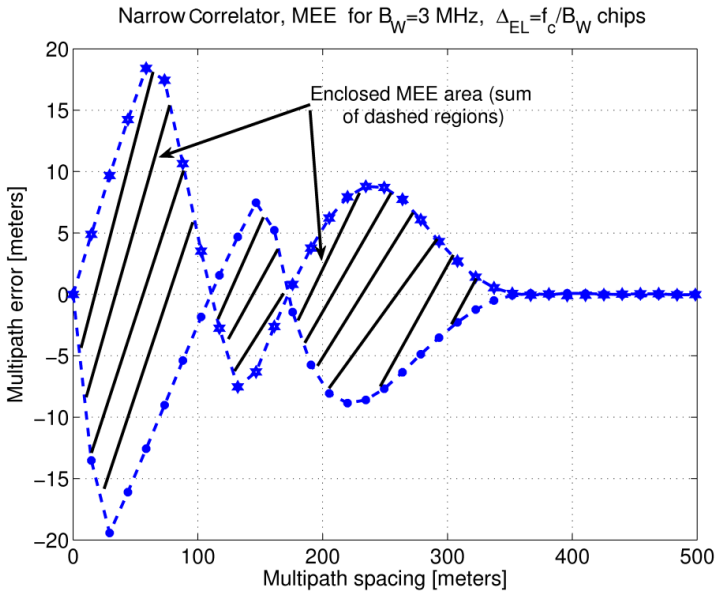

Figure 5.3: Illustration of enclosed MEE area for Narrow Correlator case. $\Delta_1=\Delta_{EL}=0.34$chips

In order to search for the optimized coefficients a_1, a_2 and a_3, a vector v_i= [-1:0.1:1] is defined, with a resolution of 0.1 and values between -1 and 1, which contains the possible values of the optimum coefficient. a_1 is set to one without loss generality. Therefore, a_2 and a_3 would be searched in the vector. The channel is considered as a two-path static channel with the first path having unit amplitude and the second path having amplitudes from 0.5 to 0.95 with a step of 0.05. The final MEE will be obtained as an average of all MEEs for each channel profile. Based on the above assumptions, the optimum coefficient values a_2 and a_3 are found via a two dimensional search for the second and third correlator pairs, respectively. The discriminator function of MGD structure is given by:

$$D(\tau)=\begin{cases}\sum_{i=1}^{N} a_i \left(\left| R_I\left(\tau+\frac{\Delta_i}{2}\right)+jR_Q\left(\tau+\frac{\Delta_i}{2}\right)\right|^P - \left| R_I\left(\tau-\frac{\Delta_i}{2}\right)+jR_Q\left(\tau-\frac{\Delta_i}{2}\right)\right|^P \right), & P=1,2 \\ \sum_{i=1}^{N} a_i \left(\left| R_I\left(\tau+\frac{\Delta_i}{2}\right)\right|+\left|R_Q\left(\tau+\frac{\Delta_i}{2}\right)\right| - \left| R_I\left(\tau-\frac{\Delta_i}{2}\right)\right|-\left|R_Q\left(\tau-\frac{\Delta_i}{2}\right)\right| \right), & P=-1 \end{cases} \quad (27)$$

where N is the number of correlator pairs; $R_I(\cdot)$ and $R_Q(\cdot)$ are the In-phase and Quadrature-phase correlator functions between the received signal and the reference code, respectively; the spacing between the i^{th} early and late correlator is equal to Δ_i; uniform spacing is used, which means $\Delta_i=i\Delta_1$. The factor P determines the type of non-

linearity: *P*=2 (square of envelope) and *P*=1 (envelope). We also introduce the notation *P*=-1, which stands for the sum of absolute value of real part and imaginary part of correlation function.

5.2.1 The analysis under infinite bandwidth of the front-end filter

We noticed that the enclosed area and the optimum coefficient of MGD with *P*=-1 are exactly the same as that with *P*=1. Therefore, we will not list those parameters for *P*=-1 separately.

If we compare Table 5.1 with Table 5.2, the use of envelopes (*P*=1) gives smaller or equal enclosure area as compared to the use of squaring envelopes (*P*=2) for all the MBOC implementations.

Table 5.1: Average enclosed MEE areas [chips] with optimum coefficients for MBOC implementations when P=1or P=-1, infinite bandwidth case

P=1 or P=-1					
E-L Spacing (Δ_1) (chips)	CBOC(+)	CBOC(-)	CBOC(+/-)	TMBOC(9/11)	TMBOC(29/33)
0.08	0.0040	0.0031	0.0033	0.0045	0.0044
0.16	0.0071	0.0055	0.0056	0.0049	0.0063
0.32	0.0381	0.0350	0.0364	0.0404	0.0394
0.5	0.0360	0.0318	0.0318	0.0312	0.0344

Table 5.2: Average enclosed MEE areas [chips] with optimum coefficients for MBOC implementations when P=2, infinite bandwidth case

P=2					
E-L Spacing (Δ_1) (chips)	CBOC(+)	CBOC(-)	CBOC(+/-)	TMBOC(9/11)	TMBOC(29/33)
0.08	0.0041	0.0033	0.0034	0.0046	0.0044
0.16	0.0069	0.0061	0.0059	0.0057	0.0064
0.32	0.0385	0.0366	0.0378	0.0414	0.0393
0.5	0.0375	0.0294	0.0330	0.0327	0.0355

Table 5.3: Optimum coefficients for MBOC signals, infinite bandwidth case

P=1															
E-L spacing (Δ_1) (chips)	CBOC(+)			CBOC(-)			CBOC(+/-)			TMBOC(9/11)			TMBOC(29/33)		
	a_1	a_2	a_3	a_1	a_2	a_3	a_1	a_2	a_3	a_1	a_2	a_3	a_1	a_2	a_3
0.08	1	-0.5	0	1	-0.2	-0.2	1	-0.2	-0.2	1	0.7	-0.7	1	0	-0.3
0.16	1	-0.5	0	1	-0.6	0.1	1	-0.6	0.1	1	-0.6	0.1	1	-0.6	0.1
0.32	1	1	0.2	1	1	0	1	1	0.1	1	-1	-0.4	1	-0.8	-0.3
0.5	1	0.4	-0.2	1	0.3	0.1	1	0.4	-0.2	1	0.3	-0.2	1	0.4	-0.2

The optimum coefficients for three early-late correlator pairs with different E-L spacing are shown in Table 5.3. As can be seen in Table 5.3, the optimum coefficients are highly dependent on the early-late spacing and the modulation type. Therefore they should be derived according to the desired MGD structure and modulation type. However, in order to have a simple and uniform receiver model, which means the parameters used in one tracking algorithm will not change with early-late spacing, only one optimum coefficient can be used. Since the optimum coefficients [1 -0.6 0.1] appears more frequently, it will be used in the Simulink-based simulations, which will be presented in Chapter 9.

Figure 5.4 shows an example of MEE curve for nEML, HRC and optimized MGD with [1 -0.6 0.1] when Δ_1=0.16 chips for CBOC(-) signal.

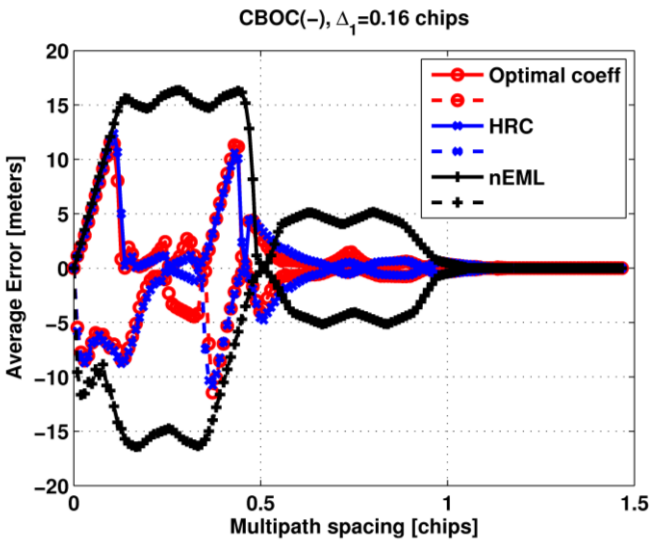

Figure 5.4: Average MEE curve of nEML, HRC and optimum MGD with [1 -0.6 0.1]

5.2.2 The analysis under limited bandwidth of the front-end filter

In reality, the received signal passes through the Radio Frequency (RF) front-end and the filter in the receiver's front-end filters out some part of the signal spectrum. The

effects on the MGD optimization from this limited front-end bandwidth are studied here.

One difference from the infinite bandwidth analysis is that, the SinBOC modulated reference codes are also considered here. This is because the SinBOC(1,1) receiver is more likely to be used in mass market receiver design, having a lower complexity than the MBOC receiver. The optimum coefficients were found for four scenarios:

- CBOC(+) signal with reference CBOC(+) receiver
- CBOC(+) signal with reference SinBOC(1,1) receiver
- CBOC(-) signal with reference CBOC(-) receiver
- CBOC(-) signal with reference SinBOC(1,1) receiver

The correlator spacing Δ_1 is determined by the receiver front-end bandwidth by Equation (28), according to [31].

$$\Delta_1 = f_C / B_W \tag{28}$$

where f_C is chip rate in MHz, for Galileo E1 signal, f_C is 1.023 MHz; B_W is receiver double-sided front-end bandwidth in MHz. Four bandwidths are considered in the process of optimization. They are B_W=20.46 MHz and B_W=24.552 MHz, which are defined in the latest Galileo OS SIS ICD, and B_W=3 MHz, B_W=4 MHz, which are considered as low front-end bandwidth in receiver design for mass market.

Table 5.4: Average enclosed MEE area [chips] for MBOC implementation when P=1 and P=-1, limited bandwidth case

P=1 or P=-1				
Tx	CBOC(-)		CBOC(+)	
Rx	CBOC(-)	SinBOC(1,1)	CBOC(+)	SinBOC(1,1)
3 MHz	0.0296	0.0298	0.0312	0.0302
4 MHz	0.0261	0.0266	0.0301	0.0288
20.46 MHz	0.0024	0.0033	0.0035	0.0035
24.552 MHz	0.0021	0.0023	0.0029	0.0024

Table 5.5: Averaged enclosed MEE area [chips] for MBOC implementation when P=2, limited bandwidth case

P=2				
Tx	CBOC(-)		CBOC(+)	
Rx	CBOC(-)	SinBOC(1,1)	CBOC(+)	SinBOC(1,1)
3 MHz	0.0297	0.03	0.0317	0.03
4 MHz	0.0287	0.0294	0.0327	0.032
20.46 MHz	0.0025	0.0035	0.0037	0.0038
24.552 MHz	0.0024	0.0027	0.0031	0.0024

Similar with the results under infinite bandwidth assumption, the average enclosed MEE area with P=1 gives better results than that of P=2 as shown in Table 5.4 and Table 5.5.

Moreover, the results under P=1 and P=-1 are equal, which means that any of the two design options based on envelopes are equally suitable. Comparing the results in Table 5.4 and Table 5.5 with the results of CBOC(+) and CBOC(-) under the infinite bandwidth assumption, it is found that for low bandwidths (e.g., 3 or 4 MHz), the averaged enclosed MEE area is as big as those with wide early-late spacing (i.e., 0.32 or 0.5 chips). This is because the narrow front-end bandwidth restricts the minimum possible correlator spacing for a specific front-end bandwidth.

Table 5.6: Averaged enclosed MEE area [chips] for optimum MGD, HRC and nEML when P=1 for CBOC(-) modulated signal, limited bandwidth case

P=1						
Tx	CBOC(-)					
Rx	CBOC(-)			SinBOC(1,1)		
	MGD	HRC	nEML	MGD	HRC	nEML
3 MHz	0.0296	0.0307	0.0306	0.0298	0.0308	0.0313
4 MHz	0.0261	0.0357	0.0326	0.0266	0.039	0.0335
20.46 MHz	0.0024	0.0031	0.0047	0.0033	0.0033	0.0096
24.552 MHz	0.0021	0.0024	0.0043	0.0023	0.0025	0.0084

Table 5.7: Averaged enclosed MEE area [chips] for optimum MGD, HRC and nEML when P=1 for CBOC(+) modulated signal, limited bandwidth case

P=1						
Tx	CBOC(+)					
Rx	CBOC(+)			SinBOC(1,1)		
	MGD	HRC	nEML	MGD	HRC	nEML
3 MHz	0.0312	0.0338	0.0335	0.0302	0.0324	0.0328
4 MHz	0.0301	0.0428	0.0359	0.0288	0.0415	0.0349
20.46 MHz	0.0035	0.004	0.0067	0.0035	0.0038	0.0112
24.552 MHz	0.0029	0.0029	0.0059	0.0024	0.0024	0.0097

In all the cases with different bandwidths as shown in Table 5.6 and Table 5.7, the enclosed area of MGD is smaller than the enclosed area of HRC and nEML. It was also found that the HRC has bigger enclosed area than nEML when front-end bandwidth is narrow. This points out the fact that HRC is not robust enough for narrow receiver front-end bandwidths. The optimum coefficients for MGD with three types of non-linearity are shown in Table 5.8 and Table 5.9.

Figure 5.5 and Figure 5.6 show the average MEE (over varying second path amplitude) for the nEML, HRC and optimum MGD with B_W=3 MHz and CBOC(-) signal with SinBOC(1,1) reference code and CBOC(-) reference code, respectively. The slight variations in the MEE curves can be explained by the fact that, some spurious peaks might be obtained under certain second path amplitude (e.g., in the cases when the second path amplitude are much closer to the first path amplitude), and these spurious peaks make the averaged MEE less smooth than what is usually reported in literate under fixed second path amplitude.

Figure 5.7 and Figure 5.8 show the average MEE for the nEML, HRC and optimum MGD with B_W=24.552 MHz and CBOC(-) signal with SinBOC(1,1) reference code and CBOC(-) reference code, respectively. We remark that, for lower bandwidth (i.e., 3 or 4 MHz), which is typical for mass-market receiver, it makes sense to use a reference SinBOC(1,1) receiver for a low complexity implementation, while for higher front-end bandwidth (e.g., 24.552 MHz as specified in Galileo OS SIS ICD), a reference CBOC receiver will achieve the best performance.

Table 5.8: Optimum Coefficients for five MBOC implementations with different E-L spacing, limited bandwidth case

P=1								
Tx	CBOC(-)				CBOC(+)			
Rx	CBOC(-)		SinBOC(1,1)		CBOC(+)		SinBOC(1,1)	
Opt_coeff	a_2	a_3	a_2	a_3	a_2	a_3	a_2	a_3
3 MHz	-0.1	-0.1	-0.1	-0.2	-0.2	-0.2	0.1	-0.3
4 MHz	0.3	-0.6	1	-1	0.4	-0.6	1	-1
20.46 MHz	0.1	-0.4	-0.2	-0.2	0.1	-0.4	-0.2	-0.2
24.552 MHz	0.3	-0.5	-0.8	0.2	-0.5	0	-0.8	0.2

Table 5.9: Optimum Coefficients for five MBOC implementations with different E-L spacing, limited bandwidth case

P=2								
Tx	CBOC(-)				CBOC(+)			
Rx	CBOC(-)		SinBOC(1,1)		CBOC(+)		SinBOC(1,1)	
Opt_coeff	a_2	a_3	a_2	a_3	a_2	a_3	a_2	a_3
3 MHz	0.1	-0.2	0.1	-0.2	0.1	-0.4	0.1	-0.3
4 MHz	0.9	-1	0.9	-1	1	-1	1	-1
20.46 MHz	0.2	-0.6	-0.3	-0.2	0.1	-0.5	-0.3	-0.2
24.552 MHz	0.3	-0.6	-0.8	0.2	-0.8	0.2	-0.8	0.2

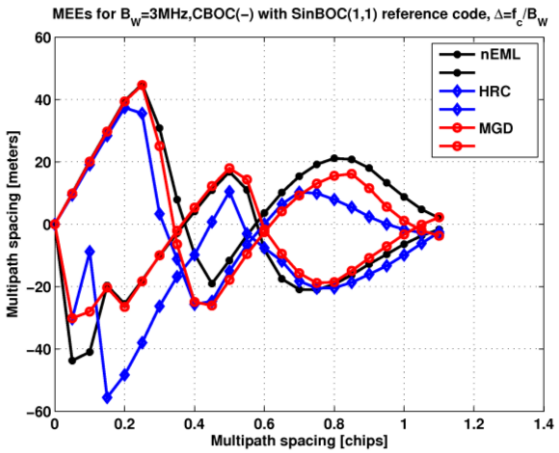

Figure 5.5: The average MEE of nEML, HRC and optimum MGD with parameters a=[1 -0.1 -0.2]. P=1, B_W=3 MHz, CBOC(-) signal with SinBOC(1,1) reference code

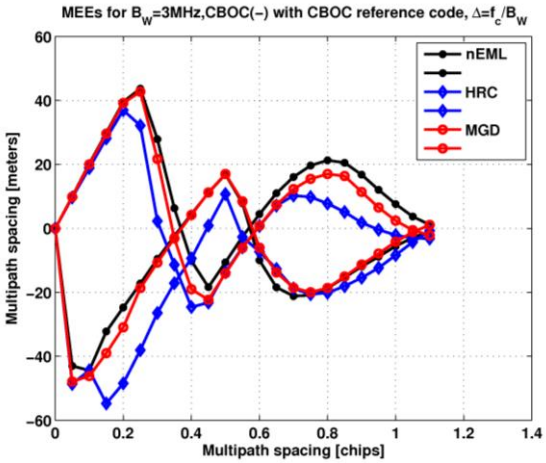

Figure 5.6: The average MEE of nEML, HRC and optimum MGD with parameters a=[1 -0.1 -0.1]. P=1, B_W=3 MHz, CBOC(-) signal with CBOC(-) reference code

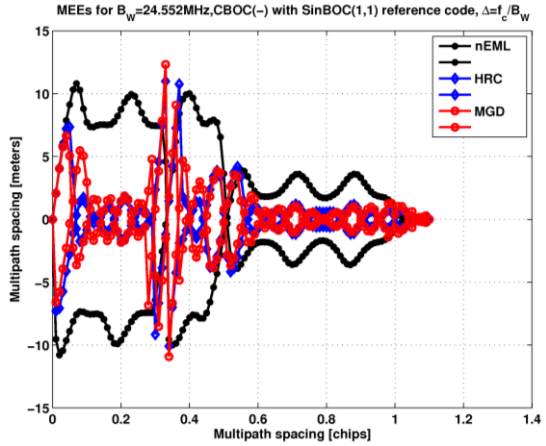

Figure 5.7: The average MEE of nEML, HRC and optimum MGD with parameters a=[1 -0.8 0.2]. P=1, B_W=24.552 MHz, CBOC(-) signal with SinBOC(1,1) reference code

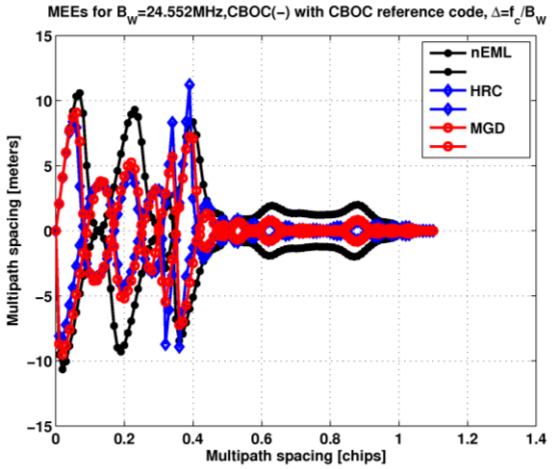

Figure 5.8: The average MEE of nEML, HRC and optimum MGD with parameters a=[1 0.3 -0.5]. P=1, BW=24.552 MHz, CBOC(-) signal with CBOC(-) reference code

The results from Figure 5.5 to Figure 5.8 showed that HRC is clearly not a good option in terms of MEE performance at low receiver bandwidth. For low bandwidths, MGD is slightly better than nEML, but the gap is not significant. For high bandwidths, HRC and MGD outperforms the nEML, while having a very similar performance. It seems that, in terms of MEE, the only advantage of using MGD versus nEML at low bandwidth and HRC at high bandwidth is its higher flexibility and ability to offer a patent-free solution, adjusted to the designer's needs (e.g., according to desired correlator spacing and sampling frequency for a specific front-end bandwidth).

6. GNSS simulators

This chapter tries to justify the need for building a Simulink-based software tool for Galileo signals and to emphasize the missing parts or drawbacks in existing GNSS simulators nowadays.

During the past few years, several PC-based real-time software receivers have been developed in both academic world and industrial world. The basic idea is to position a wideband A/D converter as close to an antenna as it is convenient, then transfer those samples into a programmable element, and apply digital signal processing techniques to obtain the desired results [15]. It removes the analog components and their nonlinear, temperature-based, age-based characteristics and provides ultimate simulation environment [32]. As stated in [33], '*A GNSS development tool nowadays has to be upgradeable, flexible, expandable and open when it comes to your challenges of modern GNSS signals'*, which justifies the software approach. Six main software-defined receiver simulator prototypes/projects found in the literature will be described in this chapter.

6.1 IRGAL software receiver

IRGAL software receiver was developed by the Navigation Signal Analysis and Simulation (NavSAS) research group, which is a joint team of Istituto Superiore Mario Boella (ISMB) and Politecnico di Totino that acts in the satellite navigation and localization sectors in the Galileo Lab located in ISMB. One of their main research topics is to design and develop a GPS/Galileo receiver in software radio technology [32].

IRGAL software receiver was developed between 2006 and 2008 in Italy. It consists of a hardware RF front-end and software receiver. They are connected by an USB interface. The software receiver can work with any front end, which has USB interface. The receiver is working for GPS L1 and Galileo E1 frequency band and MBOC was not implemented, but it is upgradable. Acquisition and tracking blocks are optimized in C language on Field Programmable Gate Array (FPGA) + Advanced RISC Machine (ARM) prototyping hardware. The acquisition is FFT based and the tracking is using second order tracking loops, which can be changed by users [32]. According to the test reporting in [32], the position accuracy Root Mean Square Error (RMSE) is less than 10 m using code-based measurement and Time To First Fix (TTFF) in cold start mode is less than 45 seconds. The biggest disadvantage of this receiver is that there is not much

information available. Many key parameters cannot be found, such as operational CNR and receiver bandwidth.

6.2 GSNRx™

GSNRx™ is an on-going development in Position Location and Navigation (PLAN) group at University of Calgary. The whole modular design is written in C++. The entire receiver processing is implemented in software. Sampling rate and intermediate frequency are user-selectable. Both acquisition and tracking are implemented in software and capable for GPS L1C, L2C and L5, Galileo E1 and E5a, and GLONASS L1 and L2. However, it is patent-protected and not available for general use. [34]

6.3 IpexSR SW Rx

PC-based Experimental Software Receiver (ipexSR) developed in Institute of Geodesy and Navigation is a high-bandwidth dual-frequency L1/L2 C/A code software receiver. The implemented receiver is in C++ mixed with assembler code to increase performance on standard PCs running under Windows [35]. The ipexSR performs signal acquisition based on FFT techniques. It can be repeated for a user-defined number of times if acquisition is not successful. Tracking is also implemented in the receiver. Users can control the tracking loop bandwidth and some multipath mitigation algorithms are also utilized in Delay Lock Loop (DLL).

The comparison of three simulators mentioned above is summarized in Table 6.1.

Table 6.1: Comparison of IRGAL, GNSR and IpexSR simulators. N/F= not found

Feature	IRGAL SW rx	GSNRx	IpexSR
Signal type	GPS L1 & Galileo E1; MBOC not implemented	GPS L1 C/A, L1C, L2C & L5, Galileo E1&E5a, E5b, GLONASS L1&L2; MBOC not implemented	GPS L1, L2 & L5;Galileo signal not implemented
Sampling frequency	17.5103 MHz	40 MHz (20 MHz per I/Q channel)	40.96 MHz
IF frequency	4.5102 MHz	User-selectable	8.087/8.287 MHz
CNR	N/F	>=40 dB-Hz for acquisition >=35 dB-Hz for tracking	N/F
Quantization	1-8 bits	3 bits (per I and Q samples)	2~4 bits
Acquisition/tracking modules	Included	Included	Included
Bandwidth	N/F	Adjustable	15~20 MHz
Software platform	N/F	C++	C++ and assembler code
Multipath mitigation	N/F	Not implemented	Implemented

6.4 GNSS digitized IF signal simulator

GNSS Digitized IF Signal Simulator (GDISS) is developed by Electronics and Telecommunications Research Institute (ETRI) in Daejeon, Korea, as a part of development of software based Test & Evaluation Facility, which provides test and evaluation environment for various software level application and navigation algorithms in GNSS. GDISS provides two main capabilities: Raw Data Generation (RDG), which is used to generate GPS and Galileo observables and Digitized IF Signal Generation (DISG), which is used to generate GPS L1 C/A, L2C and Galileo E1 (E1B and E1C) digitized IF signals. Therefore, this simulator does not include any acquisition and tracking blocks. The main specifications of GDISS prototype for L1 C/A, L2C, and E1 (E1B and E1C) are as follows [36].

Table 6.2: Specifications of GDISS prototype

Feature	Specifications
Signal	GPS L1 C/A, Galileo E1(B&C)
Signal power	-20 dB
Sampling Frequency	5~40 MHz(default: 5.714 MHz)
IF frequency	1~20 MHz (Default:1.134 MHz)
CNR	30~50 dB-Hz
Acquisition/tracking modules	Not implemented
Bandwidth	2 MHz~4 MHz
quantization	2~4 bit
Multipath mitigation	Not implemented

6.5 Software GNSS receiver at Danish GPS center

This software receiver is a single-frequency receiver using C/A code on L1 for GPS, which is implemented in MATLAB. It is able to perform acquisition, code and carrier tracking, navigation bit extraction, navigation data decoding, pseudorange estimation and position computation. The complete receiver is in MATLAB comes with the book "A software-defined GPS and Galileo receiver: A single-frequency approach" [15]. Unlike the commercial software receiver, the code is in open access, and user can modify the code to test different algorithms. By default, the receiver supports GPS L1 and GIOVE-A signal. The specifications are shown in Table 6.3.

6.6 GRANADA Bit-true Receiver simulator

The GRANADA Bit-true software receiver simulator recreates the Galileo/GPS signal-in-space and the receiver signal processing chain using a sample-based simulation approach. It is developed in Matlab/Simulink. The tool enables analysis and simulations

of the receivers' critical algorithms and architecture design, such as acquisition, tracking, signal modulation, multipath and interference analysis. The main features are shown in Table 6.3. However, the GRANADA simulator is not fully functional because it doesn't include a navigation unit. The licenses of GRANADA simulator are expensive and sources are partially encrypted. This is not very suitable for algorithm development in general use.

Table 6.3: Comparison of software GNSS receiver in Danish GPS center and GRANADA simulator

Feature	**Software-defined GNSS receiver at Danish GPS center**	**GRANADA**
Signal type	GPS L1; Galileo GIOVE-A signals	GPS L1, Galileo E1, E5A, E5B&E6
Input signal	Simulated signal	Simulated signal
Sampling frequency	8.1838 MHz	>=40 MHz
IF frequency	38.4 KHz	Related to sampling frequency
CNR	User defined	>35 dB-Hz
Acquisition/ Tracking modules	Included	Included (encrypted algorithm)
Quantization	2 bits I/Q samples	1~8 bits
Bandwidth	2 MHz	40 MHz
Software platform	Matlab	Simulink
Multipath mitigation	Not implemented	Implemented

From Table 6.1 to Table 6.3, it is remarkable that most of these baseband receiver simulators:

- Typically, they operate at moderate-to-high CNRs (i.e., above 30 or 35 dB-Hz)
- They assume a low IF or very low IF architecture
- They use FFT-based acquisition unit
- No multipath mitigation algorithms are specified for the tracking stage

Regarding the terms of the distribution of these software receivers, there are no clear terms of distribution. Some of them are not even available, but only used locally in the unit, which developed them. All mentioned above motivate the need for building a software tool for Galileo signals, which will be introduced in the next chapter.

7. GNSS Simulink model at TUT

A Galileo simulator for E1 OS signal has been built at the Department of Communications Engineering (DCE) at Tampere University of Technology (TUT). The aim is to simulate the performance of developed link-level algorithms for acquisition and tracking of Galileo signal. The simulator therefore focuses on one link at the time (one satellite and one receiver). The Simulink tool in MATLAB was chosen as the development tool for this receiver. The main reason for this choice is that Simulink has a very intuitive user interface combined with its numerous features. One of the important features is that Simulink provides an interactive graphical environment and a customizable set of block libraries, with which one can design, simulate, implement and test a variety of time-varying systems. The snapshot of this Simulink model is shown in Figure 7.1.

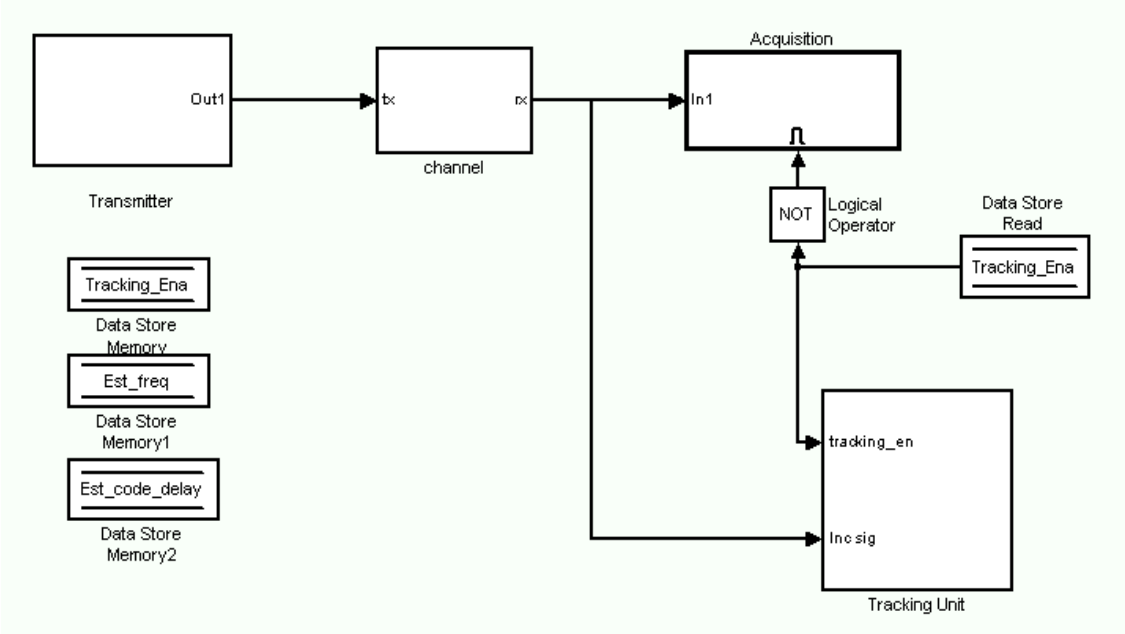

Figure 7.1: The snapshot of Simulink model

This Simulink model simulates the whole E1 OS signal chain, including the E1 transmitter block, the multipath channel, the acquisition block and the tracking unit block. Both E1B and E1C channels are acquired and tracked in the Simulink model. The key functions, such as DLL discriminator and PLL discriminator are implemented using m-language based S-function. This kind of function enables algorithms development. The carrier NCO and the code NCO generators are implemented with C-language based S-function because C-language can speed up simulations. Moreover, variable

Intermediate Frequency (IF) and sampling frequency can be used, according to designers' requirement. For example, we use 3.42 MHz intermediate frequency and 13 MHz sampling rate in the simulations presented in this thesis.

The basic model has been originally created by a former member of DCE group and contained only SinBOC(1,1) tracking and nEML code tracking loop. The thesis author developed the tracking unit of this model, introducing the following contributions:

- Both CBOC and SinBOC(1,1) delay tracking
- Code loop bandwidth optimization
- Additional delay trackers, namely HRC, MGD, DP, SBME and two-stage estimators
- Switching architecture in the transmitter block

7.1 Transmitter

The E1 transmitter is implemented based on CBOC modulation, including primary code and secondary code, in the accordance with the latest Galileo OS SIS ICD. The snapshot of E1 transmitter is shown in Figure 7.2.

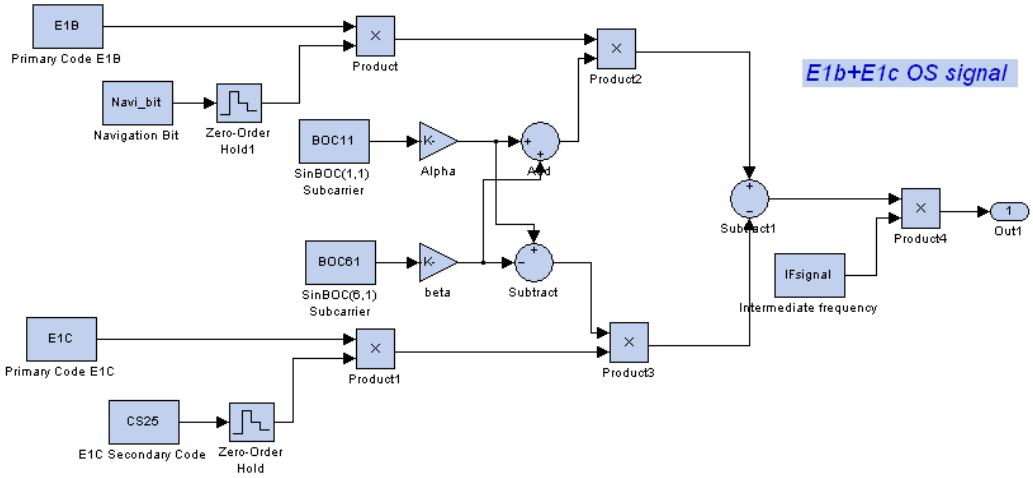

Figure 7.2: The snapshot of E1 transmitter block

In the transmitter block, E1B is CBOC('+') modulated signal with navigation data and E1C is CBOC('-') modulated signal with a pre-defined bit sequence of CS25 (i.e., pilot channel). The E1 signal is formed as the difference between those the two signals, in accordance with the Galileo OS SIS ICD of Feb [9].

7.2 Channel

The Simulink channel model generates the multipath signals in a user-defined CNR and complex noise. The interference from GPS or other sources, excepting noise and multipath are not considered here. The basic function of the channel block can be modeled as follows,

$$r_{E1}(t) = \sum_{i=1}^{L} \alpha_i(t) S_{E1}(t - \tau_i) + n \tag{29}$$

where $r_{E1}(t)$ is the received E1 signal, which is the output of channel block and $\alpha_i(t)$ and τ_i are the path gain and the path delay for the i^{th} path, respectively; $S_{E1}(t)$ is the transmitted signal, which is the output of the transmitter block; L is the number of multipath and n is the AWGN based on the user-defined CNR. Figure 7.3 shows the snapshot of the channel block.

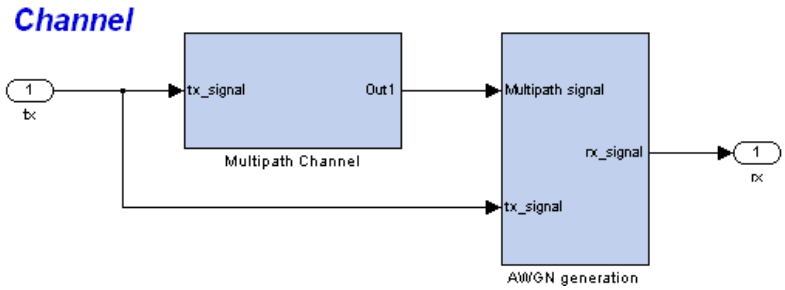

Figure 7.3: The snapshot of channel model

Here, maximum 5 paths, including the Line-Of-Sight (LOS) ray are implemented in 'Multipath Channel' block. After the generation of the multipath signal, a zero mean complex white Gaussian noise is generated in 'AWGN generation' block. The power of noise also needs to be configured before the simulation, and the rang of CNR is typically from 35 to 55 dB-Hz for good GNSS signals. The output of 'AWGN generation' block is the received signal, which is then fed to the acquisition and tracking blocks.

7.3 Receiver

The synchronization of GNSS signal is composed of two stages: i. the acquisition stage, and ii. the tracking stage. Therefore, both acquisition and tracking blocks are implemented in Simulink.

7.3.1 Acquisition unit

When the signal passes through the channel, the 'Acquisition' block is first activated. The acquisition is based on FFT technique. The carrier frequency and the code phase are roughly estimated in the acquisition stage. They are stored in memory, which can be used in the tracking unit to generate a local PRN reference code. Currently, both SinBOC(1,1) and CBOC reference codes can be used in Acquisition block. An example of time-frequency mesh in the acquisition block is shown in Figure 7.4. It shows the time-frequency correlation output and the position of the main peak signal gives the estimated frequency and code delay.

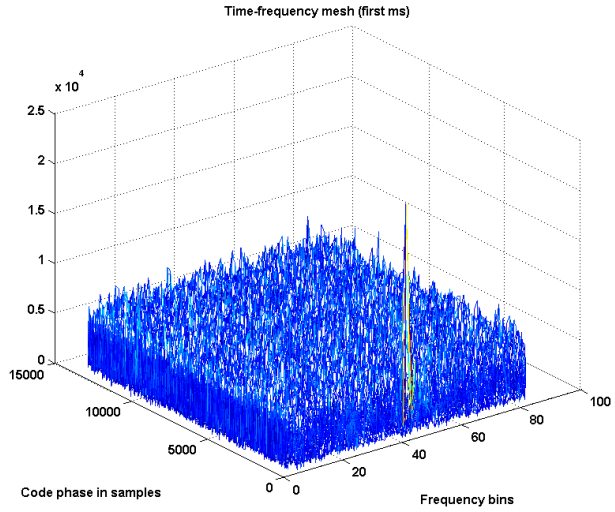

Figure 7.4: Time-Frequency mesh in the E1 acquisition process

7.3.2 Tracking unit

When a signal is detected in the 'Acquisition' block, a control signal 'Tracking_Ena', which is stored in memory will activate the 'Tracking unit'. The tracking unit consists of three main blocks: carrier wipe-off block, code NCO block and dual channel correlation and discriminator block as seen in Figure 7.5.

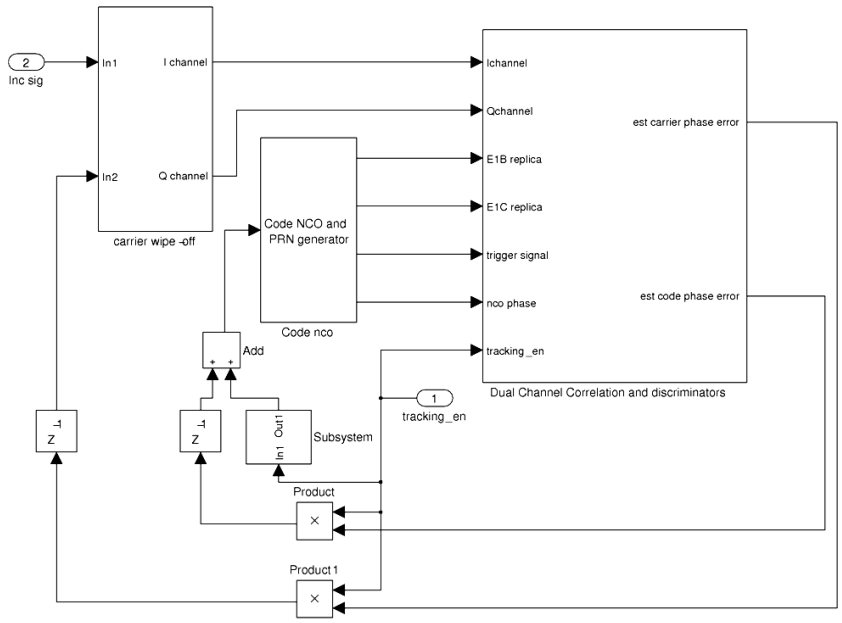

Figure 7.5: The snapshot of tracking unit block

The task of the carrier wipe-off block is to down convert the incoming signal with the help of Equation (30).

$$r_{E1\_BB}(t) = r_{E1}(t)e^{-j(2\pi \hat{f}t+\hat{\phi})} \tag{30}$$

where $\hat{f}$ is the estimated frequency from the acquisition; $\hat{\phi}$ is the estimated phase from PLL or FLL.

The code NCO block is used to generate the local PRN reference code which is shifted by the estimated code phase from DLL. The E1B and E1C reference code sequences are generated separately in dual channel discriminator block. According to the correlator offset and the status of the phase holding shifter, the primary code and the sub-carrier wave can be determined. The spacing between successive correlators is user defined and usually it is not higher than 0.5 chips. Currently, either SinBOC(1,1) or CBOC reference code sequence can be used in the tracking stage.

After converting the incoming signal into baseband, both the incoming signal and the locally generated reference code are used as the inputs to the dual channel correlation and discriminator block. In the dual channel correlation and discriminator block, the E1B and E1C channels are implemented separately, but the structures are approximately the same, except the reference code. In each channel, FLL, PLL and DLL are implemented. The function used in PLL is an atan2 function as mentioned in Equation (31).

$$\phi = \text{atan}(Q_{PS} / I_{PS}) \tag{31}$$

where ϕ is the output phase error, Q_{PS} and I_{PS} are the Q and I components of the prompt signal. FLL discriminator is based on atan2 funtion, as mentioned in Equation (32). The $(\phi_2 - \phi_1)/(t_2 - t_1)$ is the FLL output in degrees.

$$\begin{aligned} &\frac{\phi_2 - \phi_1}{t_2 - t_1} = \frac{\text{atan2}(dot, cross)}{t_2 - t_1}, \\ &where: \\ &dot = I_{PS1} \times I_{PS2} + Q_{PS1} \times Q_{PS2} \\ &cross = I_{PS1} \times Q_{PS2} - I_{PS1} \times Q_{PS2} \end{aligned} \tag{32}$$

In the DLL discriminator block, various DLL discriminator functions can be used for testing and development, such as nEML, HRC, MGD, which have already discussed in Chapter 4.

Before DLL discriminator output is going back to NCO, it passes through a loop filter. Currently, it is a first order loop filter, which is exactly the same as described in [7]. The loop filter bandwidth is chosen empirically as a good trade-off between fast convergence and low noise. The average of filtered estimated code phase from E1B and E1C channels will be fed back to NCO to generate new reference code.

7.4 E5 signal transmitter

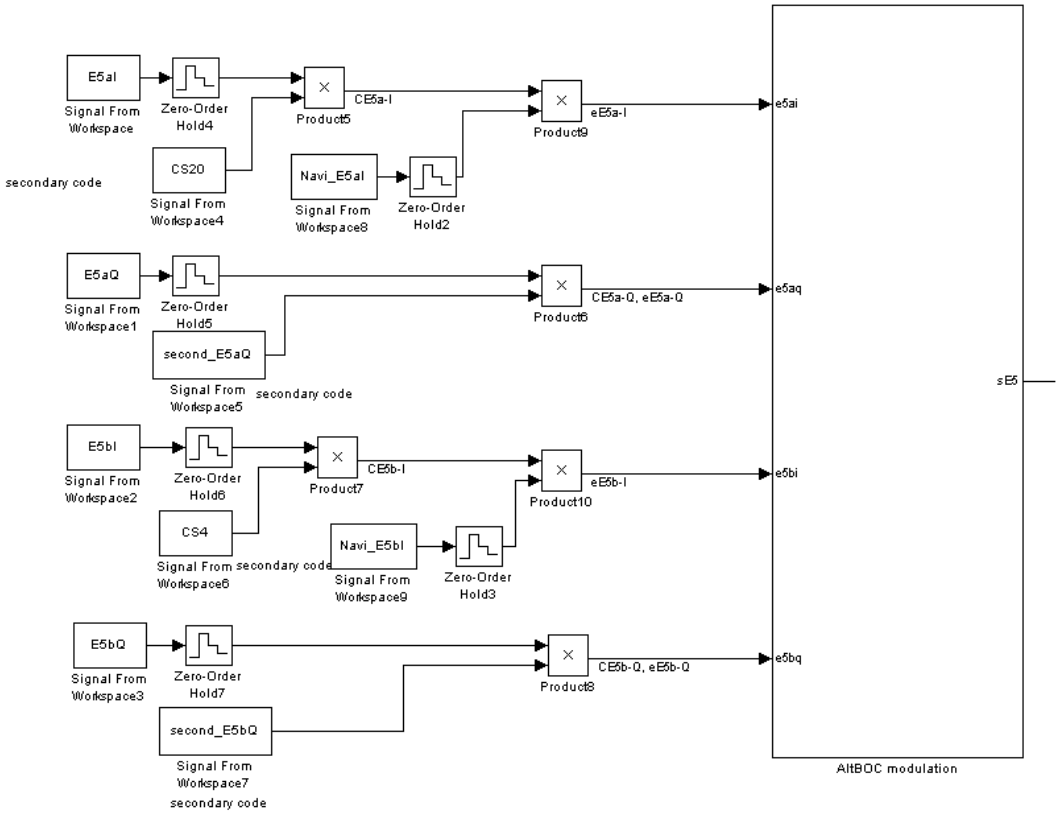

Figure 7.6: E5 signal transmitter block

E5 signal transmitter was developed at TUT at beginning of 2009. It generates E5 signal by using the AltBOC(15,10) 8-PSK modulation, as described in [9]. The snapshot of E5 signal transmitter is shown in Figure 7.6. The chosen sampling frequency is 52 MHz due to the higher chip rate (i.e.: 10.23 MHz) of the E5 signal [9].

8. Simulink model development

The original Simulink model only used SinBOC(1,1) tracking and nEML as code tracking discriminator. In order to fulfill the research requirements, some developments have been done.

8.1 NCO development

The NCO in the original Simulink model uses SinBOC(1,1) modulated reference code as shown in Figure 8.1. The ACFs of the received signal with the SinBOC(1,1) modulated E1B and E1C reference codes are almost identical, which are the typical SinBOC(1,1) ACF as presented in Figure 8.2.

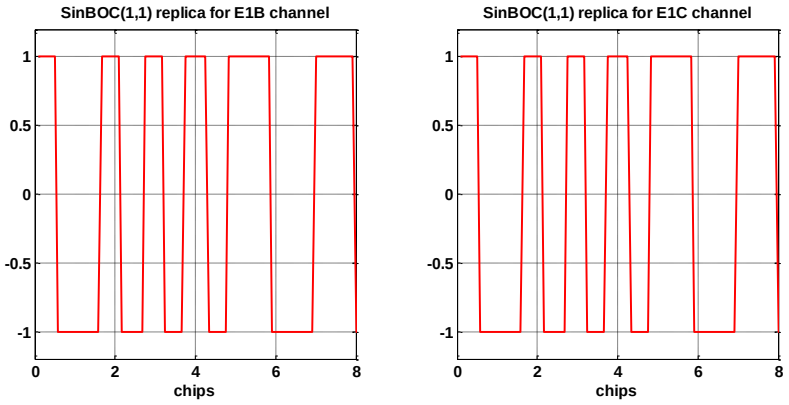

Figure 8.1: SinBOC(1,1) modulated reference codes for E1B and E1C channels

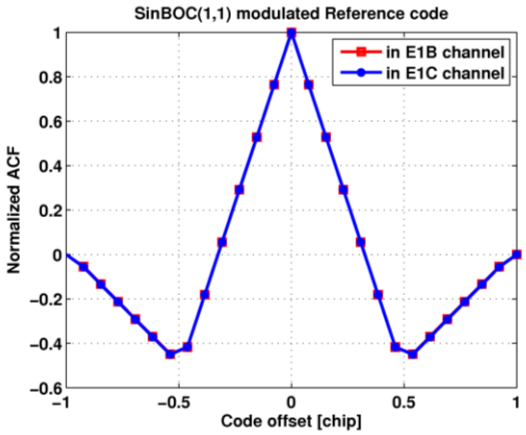

Figure 8.2: The normalized ACFs of E1B and E1C reference codes with SinBOC(1,1) modulation

The development in NCO block was done in such a way that SinBOC(6,1) sub-carrier wave was added. Then the weighted SinBOC(1,1) sub-carrier wave and weighted SinBOC(6,1) sub-carrier wave were summed or subtracted to generate CBOC(‘+’) waveform for E1B channel and CBOC(‘-’) waveform for E1C channel. Figure 8.3 shows the diagram of generating CBOC waveforms in NCO.

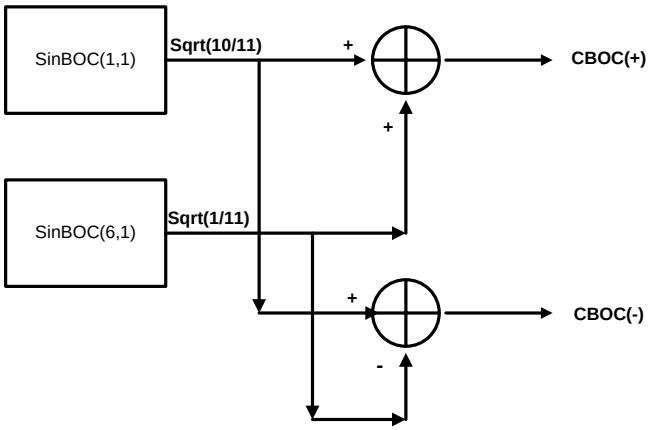

Figure 8.3: The block diagram of CBOC reference code generator in NCO

Figure 8.4 shows the CBOC(‘+’) modulated reference code for E1B channel and CBOC(‘-’) modulated reference code for E1C channel generated by developed NCO.

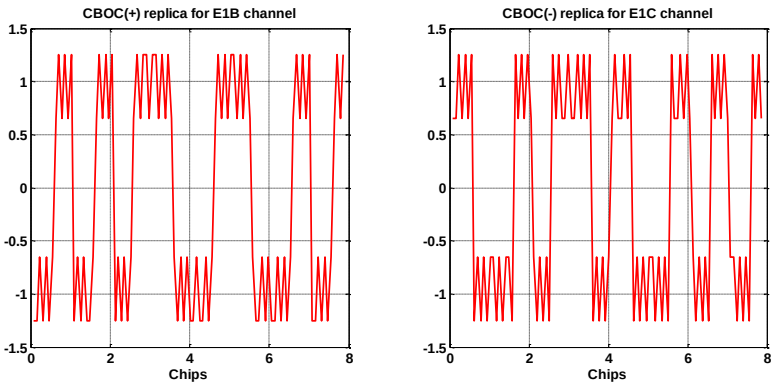

Figure 8.4: CBOC(+) and CBOC(-) modulated reference codes for E1B and E1C channels, respectively

The normalized ACFs of reference code used in E1B and E1C channel are shown in Figure 8.5. These two normalized ACFs look exactly the same as in the case of theory shown in Figure 3.6, which proves the implementation of generating CBOC modulated reference code in NCO is successful.

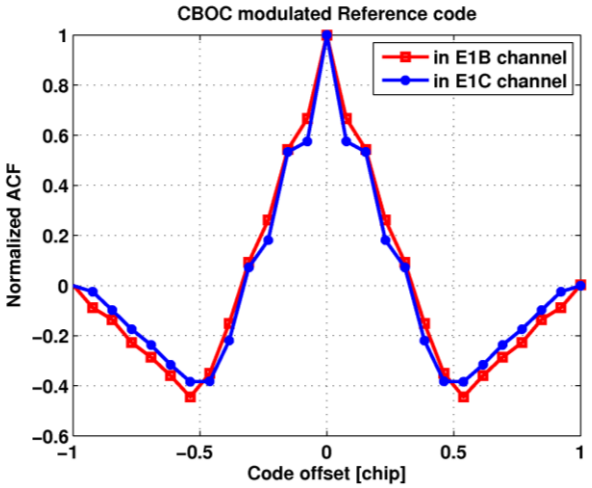

Figure 8.5: The normalized ACF of reference codes with CBOC modulation

Currently, there are two models in use. One is tracking with SinBOC(1,1) modulated reference code and another one is using CBOC modulated reference code.

8.2 Code tracking discriminator development

In the original Simulink model, only the normalized nEML was implemented as code tracking discriminator in both E1B and E1C channels. Currently, six code tracking discriminator algorithms are implemented, which are nEML, HRC, MGD, Dot Product, SBME and two-stage estimator. As the results got in Section 4.2, these discriminators are normalized by early-plus-late correlators and have the same functions as described in Section 4.2. The MGD structure and parameters are those found in the MGD optimization for infinite front-end bandwidth as described in Chapter 5. SBME first uses the same function as nEML and calculates the multipath error using the MP error estimation function as discussed in Section 4.1.5. The parameters used in MP function are summarized in Table 4.1. The two-stage estimator is implemented in such a way that nEML is running for about 0.1 second, afterwards the HRC is running for the rest of the simulation time. The CNR estimator is implemented in a separate module outside the code tracking discriminator. The estimated CNR value is stored in the memory so that it can be loaded when the nEML finishes running for 0.1 seconds. If the estimated CNR is higher than 33 dB-Hz, the HRC will be used as the discriminator in the rest of the simulation; otherwise the simulation will be continued with nEML as the discriminator. The snapshot of CNR estimator module in E1B channel of Simulink model is shown in Figure 8.6. It uses the same inputs as code tracking discriminator. The same structure is used in E1C channel.

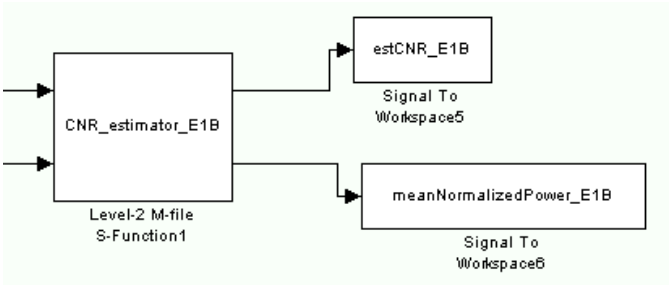

Figure 8.6: The snapshot of CNR estimator block in E1B channel

8.3 Switching architecture implementation

The basic idea of implementing switching architecture is to test if the receiver can receive E1 and E5 signals periodically. The goal is to test whether the receiver is still working and if they are working, how the receiver performance is affected by the value of switching period. From the software receiver point of view, an equivalent method is to switch the signal in transmitter, which transmits E1 signal for T seconds and then E5 signal for another T second and so on.

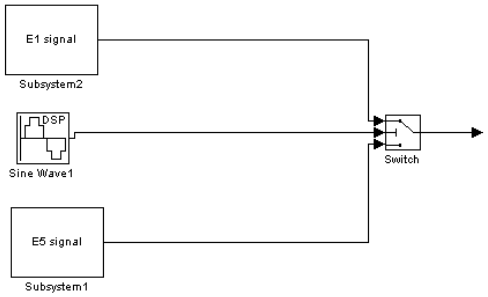

Figure 8.7: Switching architecture in transmitter block

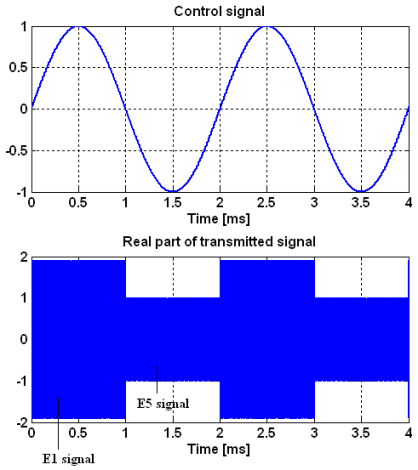

Figure 8.8: Switching architecture for 4 ms with T=1ms

The switching architecture of the Simulink model is shown in Figure 8.7. E1 and E5 signals transmission are controlled by a Sine wave block, which is used to decide the duration of T. The period of Sine wave is 2T. When the control signal is bigger than 1, E1 signal is transmitted, otherwise, E5 signal is transmitted. An example of the transmitted signal is shown in Figure 8.8. The switching period is 1 ms, E1 and E5 signals are switched exactly according to the control signal.

9. Simulink-based simulation results

In this chapter, the Simulink-based simulation results are presented. The chapter starts by presenting the comparison of different code tracking algorithms. Then it provides a comparative analysis of data/pilot tracking of E1 OS signal. The results presented in Section 9.3 show the impact of code tracking loop filter bandwidth on the tracking performance. The comparison of MBOC tracking and SinBOC(1,1) tracking is shown in Section 9.4. In the end, the performance of signal tracking with switching architecture is analyzed.

The nEML, HRC, MGD and Dot Product discriminator functions used in simulation are summarized in Table 9.1 . The SBME uses the same discriminator function as nEML as mentioned in Table 9.1 and the multipath error is calculated using Equation (17) with the parameters mentioned in Table 4.1. The two-stage estimator uses the same function as nEML and HRC as mentioned in Table 9.1.

Table 9.1: Summary of code tracking discriminator functions

Algorithm	Function used in simulations
nEML	$D = \frac{\lvert I_E\rvert + \lvert Q_E\rvert - \lvert I_L\rvert - \lvert Q_L\rvert}{\lvert I_E\rvert + \lvert Q_E\rvert + \lvert I_L\rvert + \lvert Q_L\rvert}$
HRC	$D = \frac{a_1(\lvert I_E\rvert + \lvert Q_E\rvert - \lvert I_L\rvert - \lvert Q_L\rvert) + a_2(\lvert I_{VE}\rvert + \lvert Q_{VE}\rvert - \lvert I_{VL}\rvert - \lvert Q_{VL}\rvert)}{\lvert I_E\rvert + \lvert Q_E\rvert + \lvert I_L\rvert + \lvert Q_L\rvert}, a_1 = 1, a_2 = -0.5$
MGD	$D = \frac{a_1(\lvert I_E\rvert + \lvert Q_E\rvert - \lvert I_L\rvert - \lvert Q_L\rvert) + a_2(\lvert I_{VE}\rvert + \lvert Q_{VE}\rvert - \lvert I_{VL}\rvert - \lvert Q_{VL}\rvert) + a_3(\lvert I_{VVE}\rvert + \lvert Q_{VVE}\rvert - \lvert I_{VVL}\rvert - \lvert Q_{VVL}\rvert)}{\lvert I_E\rvert + \lvert Q_E\rvert + \lvert I_L\rvert + \lvert Q_L\rvert}, a_1 = 1, a_2 = -0.6, a_3 = 0.1$
Dot Product discriminator	$D = \frac{(I_E - I_L)I_P + (Q_E - Q_L)Q_P}{I_E^2 + Q_E^2 + I_L^2 + Q_L^2}$

The simulations are carried out with developed Simulink model at TUT. The channel profile is a static channel. The signals are in-phase added together with a pre-defined path gain and a path delay. The sampling frequency is 13 MHz. All the simulations were done under infinite front-end bandwidth. The performance criteria are based on Root Mean Square Errors (RMSE) and code tracking error variance. The RMSE values are expressed in meters and they are computed as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (\hat{\tau}_{LOS}^{(i)} - \tau_{LOS}^{(i)})^2}{N}} c \tag{33}$$

where c is the speed of light, $\hat{\tau}_{LOS}^{(i)}$ is the estimated LOS delay in seconds, $\tau_{LOS}^{(i)}$ is the true LOS delay in seconds and N is the number of points used to compute the statistics.

Another evaluation criterion is the tracking error variance. If the code tracking error is $A = (\hat{\tau}_{LOS}^{(i)} - \tau_{LOS}^{(i)})c$ in meter, the tracking error variance (m^2) can then be obtained by:

$$Variance = E(A^2) - E(A)^2 \tag{34}$$

RMSE and tracking error variance are usually used together to evaluate the performance of algorithms in various environment.

9.1 Performance analysis of code tracking algorithms

In this section, we present the simulation results and the performance comparison among the delay trackers: nEML, HRC, MGD, SBME, Dot Product discriminator and two-stage estimator. The target is to test the performance of algorithms in different channel profiles.

Two channel profiles are used in the simulations. One is two paths with 1 sample (sampling frequency is 13 MHz) (around 0.08 chips) path separation, and another one is with bigger path separation, which has 3 samples (sampling frequency is 13 MHz) (around 0.2 chips).

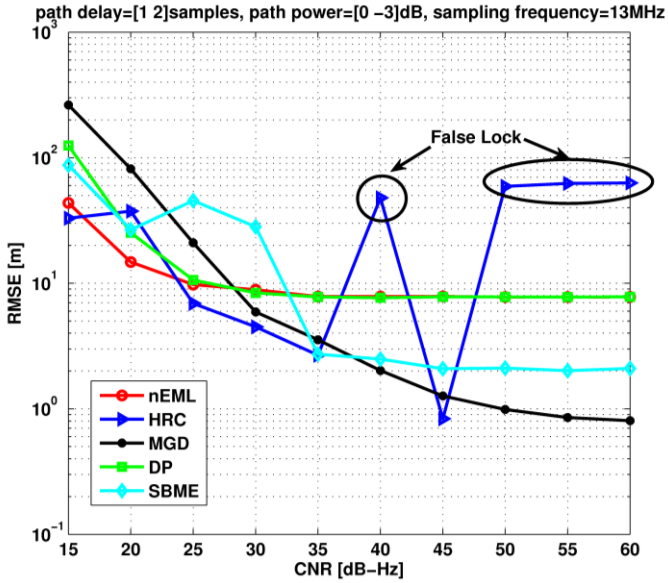

Figure 9.1: RMSE vs. CNR of nEML, HRC, MGD, Dot Product and SBME with 1 and 2 samples path delay

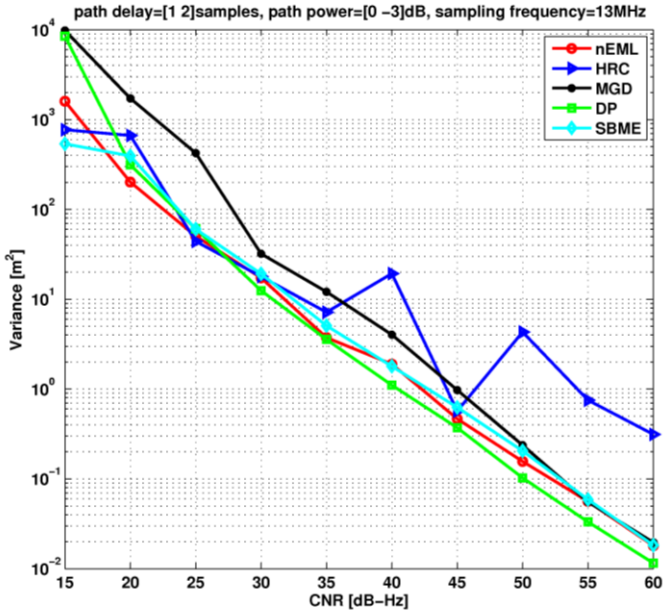

Figure 9.2: Variance vs. CNR of nEML, HRC, MGD, Dot Product and SBME with 1 and 2 samples path delay

Figure 9.1 and Figure 9.2 show the comparison between five code tracking algorithms with short-delay multipath in terms of RMSE and variance. It can be found in Figure 9.1 that the nEML and Dot Product have the same performance when the CNR=35 dB-Hz and onwards. HRC is very unstable even when the CNR is higher than 40 dB-Hz. It is because of the fact that HRC has more false lock points than that in nEML. The discriminator actually locked to the false point instead of the locking to the main peak. Comparing the performance of SBME with nEML, the RMSE of SBME is about 6 meters less than that of nEML when CNR is from 35 dB-Hz onwards. The MGD gives the best performance in this case. However, it has more than 200 meters mean error at noisy condition because the additional two correlators bring in more noise.

Figure 9.2 describes the tracking error variance of five algorithms. At the points of loss of lock in HRC, the corresponding variance is very high. MGD has the biggest variance compared with other four algorithms. This is due to the fact of that MGD has three correlator pairs so that more noise is brought in. The variance of SBME is similar to the variance of nEML, because they use the same algorithm in the code tracking discriminator. Here, the Dot Product gives the best performance due to its slightly steeper S-curve than nEML.

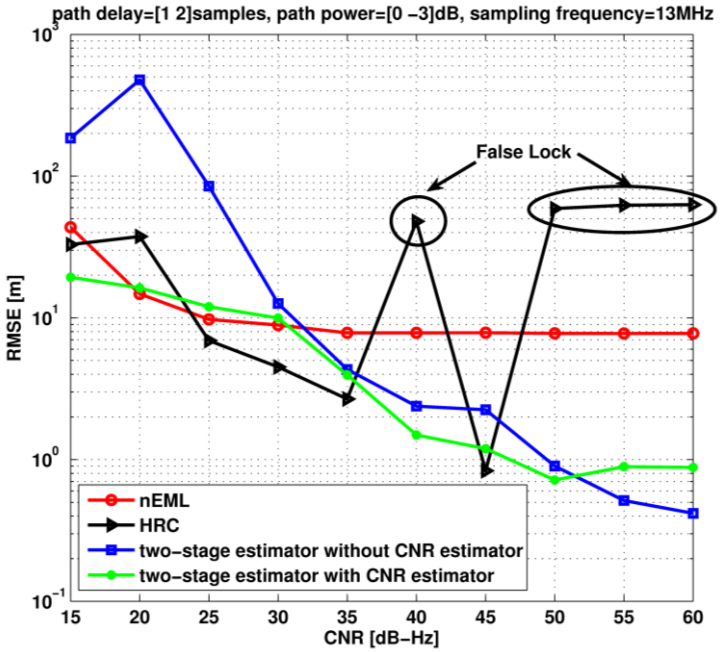

Figure 9.3: RMSE vs. CNR of nEML, HRC, two-stage estimator with/without CNR estimator with 1 and 2 samples path delay

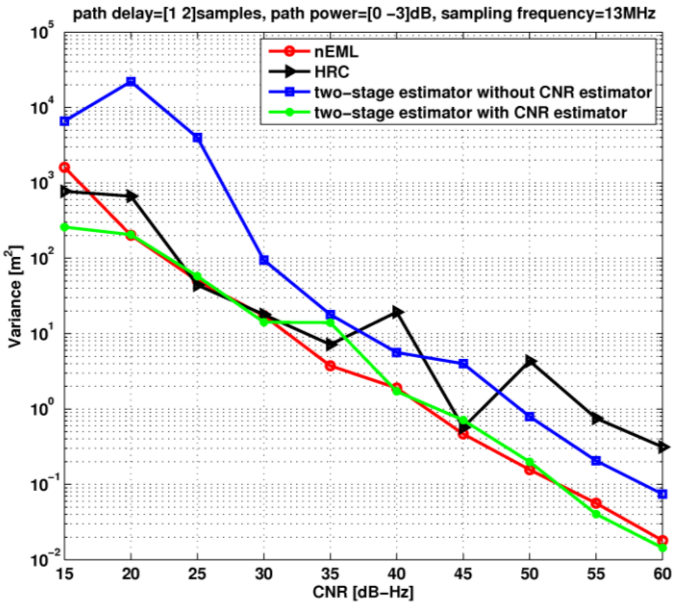

Figure 9.4: Variance vs. CNR of nEML, HRC, two-stage estimator with/without CNR estimator with 1 and 2 samples path delay

In Figure 9.3 and Figure 9.4, the RMSEs and variances versus CNR of the two-stage estimator are shown. Compared with classic HRC, the two-stage implementation solves the problem of false lock. The implementation of CNR estimator uses only the nEML at low CNRs, which avoids the bad performance of HRC in noisy condition. The breakpoint of the curve of two-stage estimator is between 30 and 35 dB-Hz CNR, it follows the CNR threshold, which is configured at 33 dB-Hz in the algorithm.

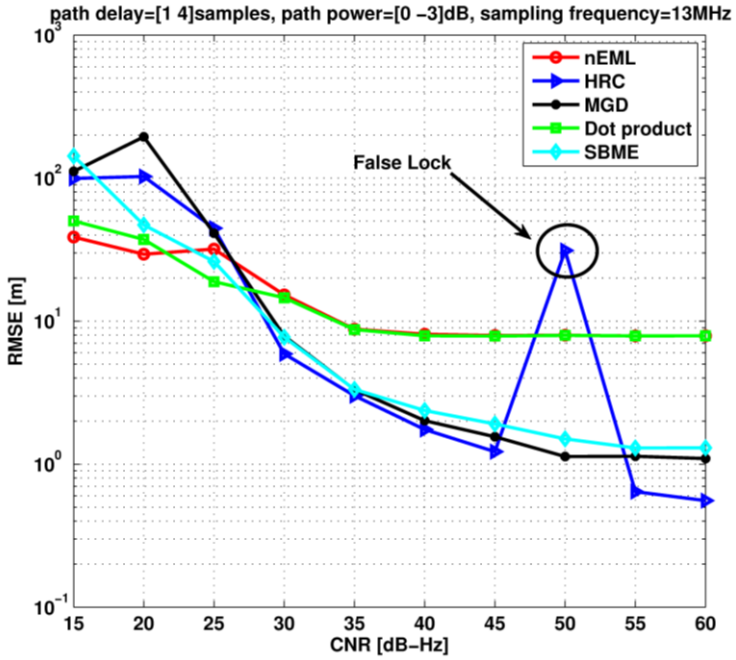

Figure 9.5: RMSE vs. CNR of nEML, HRC, MGD, Dot Product and SBME with 1 and 4 samples path delay

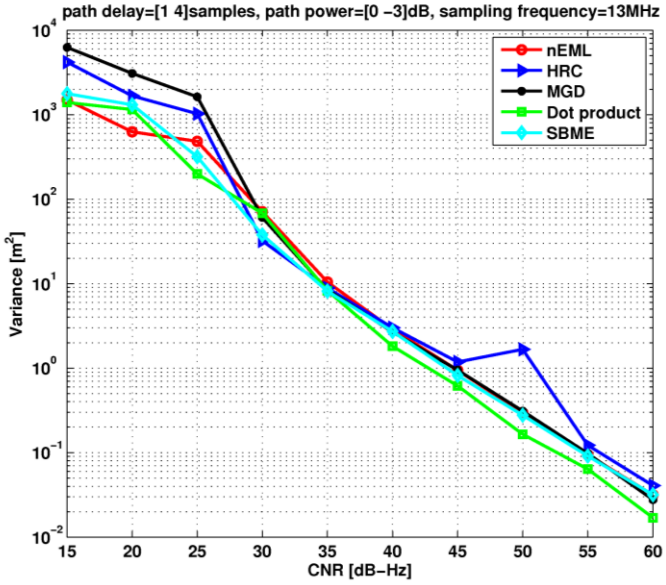

Figure 9.6: Variance vs. CNR of nEML, HRC, MGD, Dot Product and SBME with 1 and 4 samples path delay

Figure 9.5 and Figure 9.6 show the corresponding RMSE and variance results versus CNR curves for five algorithms when the path separation is 3 samples. We can see that there is only one peak in HRC curve. It is because the bigger path separation reduces the possibility of HRC locking to a false point. Again, nEML and Dot Product have similar performance. The SBME has better performance than nEML in this case too. With increasing CNR, the difference between nEML and SBME is bigger. In this case, the

MGD does not show better performance than the other algorithms and it has the worst performance in very noisy environment. Again, with this channel profile, variance of Dot Product has the best performance. Excepting the point of loss of lock in HRC, HRC has similar variance performance as MGD, SBME and nEML from CNR=35 dB-Hz onwards.

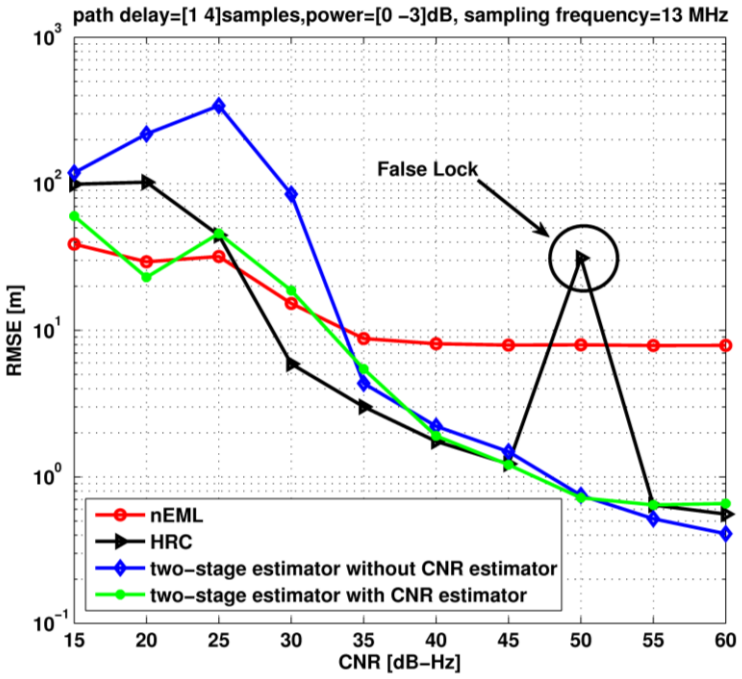

Figure 9.7: RMSE vs. CNR of nEML, HRC, two-stage estimator with/without CNR estimator with 1 and 4 samples path delay

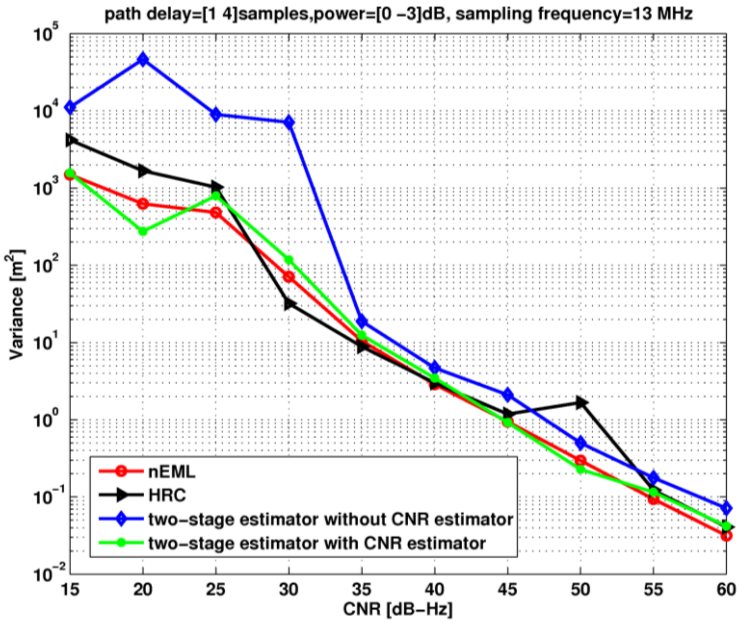

Figure 9.8: Variance vs. CNR of nEML, HRC, two-stage estimator with/without CNR estimator with 1 and 4 samples path delay

The RMSEs and tracking error variances versus CNR of two-stage estimator with 1 sample and 4 samples path delay are shown in Figure 9.7 and Figure 9.8. Similar to Figure 9.3 and Figure 9.4, the two-stage estimator avoids the false lock and follows the performance of nEML when CNR is lower than 35 dB-Hz and it has a similar performance as HRC for CNR higher than 35 dB-Hz.

9.2 Data/Pilot tracking comparison

For GNSS receiver design, the use of two signal channels per satellite will allow several choices to track signals. Therefore, it is interesting to know which tracking theme is better. The code tracking discriminator algorithm is the nEML. The channel profile is a two-path static channel with 1 and 3 samples delay (sampling frequency is 13 MHz) and the first path has unit amplitude and second path amplitude is 0.7. The E-L spacing is 0.08 chips. SinBOC(1,1) reference code and infinite front-end bandwidth are used in the simulations. The RMSEs and tracking error variances of data-only, pilot-only and combined data/pilot tracking are shown in Figure 9.9 and Figure 9.10.

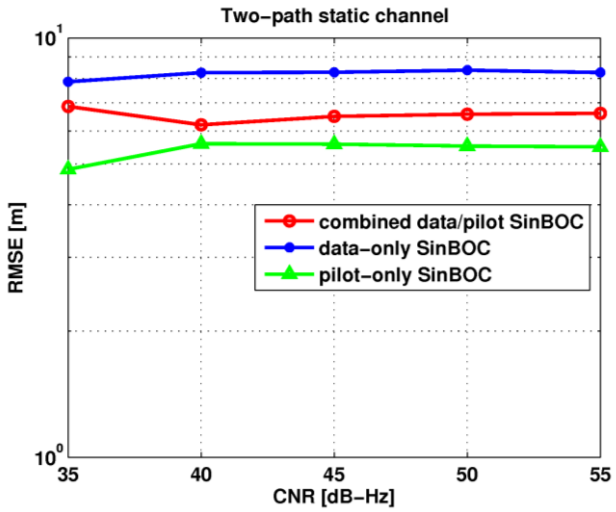

Figure 9.9: The RMSE of simulation results of data-only, pilot-only and combined data/pilot tracking

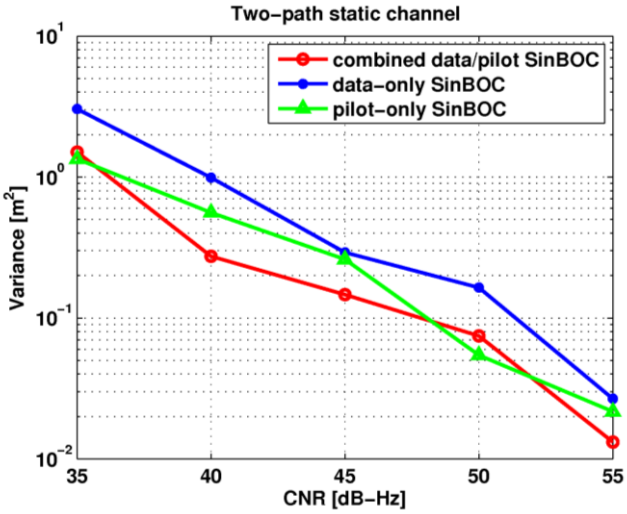

Figure 9.10: The tracking error variance of simulation results of data-only, pilot-only and combined data/pilot tracking

From Figure 9.9, it can be noticed that the tracking with pilot-only channel provides better performance as compared to that with data channel. This is because the modulation used in pilot channel is CBOC(-), which has narrower peak in correlation shape than the CBOC(+) (used in data channel). From the tracking error variance point of view, as shown in Figure 9.10, the combined data/pilot tracking has the best tracking error variance. For example, at 0.1 m^2 code tracking variance, the combined data/pilot tracking is about 5 dB better in terms of CNR than data-only, however, at this tracking error variance, pilot-only tracking has almost the same error variance as the combined data/pilot.

The RMSE curves are almost flat at different CNR, because the variance is very small compared with the multipath error, which means that the multipath is the main source of error in the tracking.

9.3 Impact of code tracking loop bandwidth

The configuration of the code tracking loop bandwidth is a tradeoff between the convergence time and noise level. In order to find the optimum loop bandwidth, the following simulations have been carried out. Two channel profiles are used: single path with 0.1 chips delay and 0 dB power and two paths with [0.1 0.15] chips delay and [0 -3] dB gain. The CNR is chosen at 35 dB-Hz. Three code tracking algorithms, nEML, HRC and MGD with parameters [1 -0.6 0.1] are used.

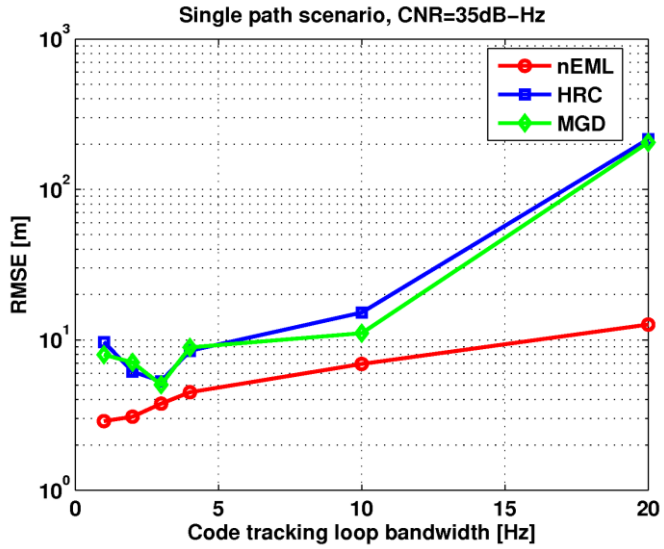

Figure 9.11: RMSE vs. code tracking loop bandwidth with nEML, HRC and MGD algorithms in single path scenario

Figure 9.11 presents the RMSE of three code tracking algorithms, nEML, HRC and MGD with different code tracking loop bandwidths in single path scenario. The tracking error of nEML is proportional to the loop bandwidth. HRC and MGD have similar tracking performance, but the tracking error is not completely proportional to the loop bandwidth. It has the trend of decreasing between 1 and 3 Hz loop bandwidth. Afterwards when the loop bandwidth is increasing, there is a significant degradation on the tracking performance. It can also be noticed that, compared to HRC and MGD, the nEML shows the best performance for all the loop bandwidths.

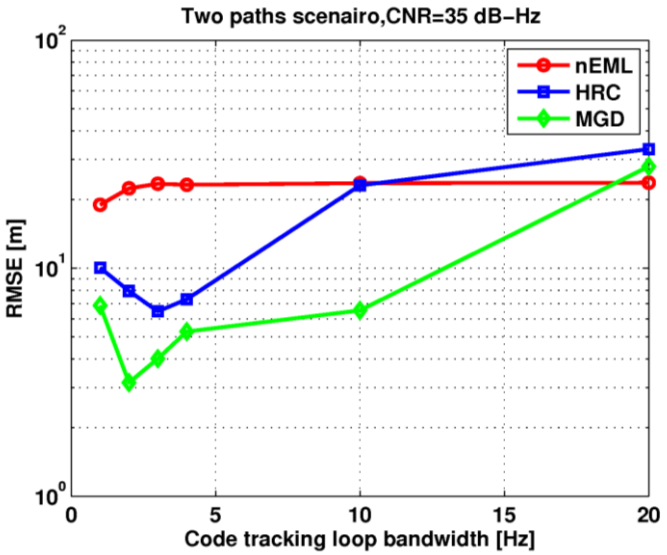

Figure 9.12: RMSE vs. code tracking loop bandwidth with nEML, HRC and MGD algorithms in two-path scenario

Figure 9.12 shows the root mean square of tracking error of nEML, HRC and MGD algorithms in two paths environment. It is found that, nEML has stronger noise resistance capability than the other two considered algorithms. The nEML performance remains constant when the loop bandwidth is wider than 4 Hz. However, with narrower loop bandwidth, when the noise level is lower, HRC and MGD show their capability of multipath mitigation. The reason for performance degradation with larger filter bandwidth is because of the fact that both HRC and MGD are very sensitive to noise.

9.4 Comparison of SinBOC(1,1) tracking and MBOC tracking

In this section, three scenarios are used for testing the performance of delay trackers under different environments. The configurations of three scenarios are shown in Table 9.2.

Table 9.2: Simulation scenarios and multipath definition

Scenario	Channel type	Relative path delay(samples)	Relative path gain (dB)
Scenario 1	Single path	1	0
Scenario 2	Two-path static channel	[1 3]	[0 -3]
Scenario 3	Four-path static channel	[1 3 5 7]	[0 -2 -4 -6]

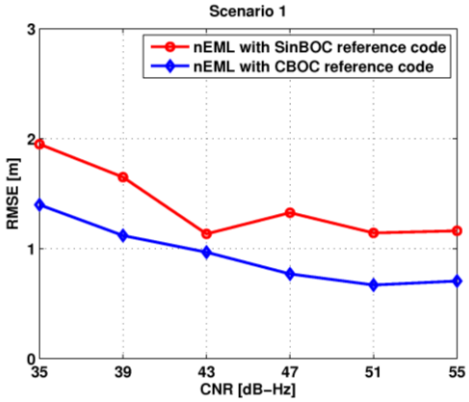

Figure 9.13: Comparison between SinBOC(1,1) tracking and MBOC tracking with nEML in scenario 1

In Figure 9.13, it shows the RMSE of nEML with SinBOC(1,1) tracking and CBOC tracking in single path scenario. The tracking with CBOC reference code is about half meter better in terms of RMSE than that with SinBOC(1,1) reference code.

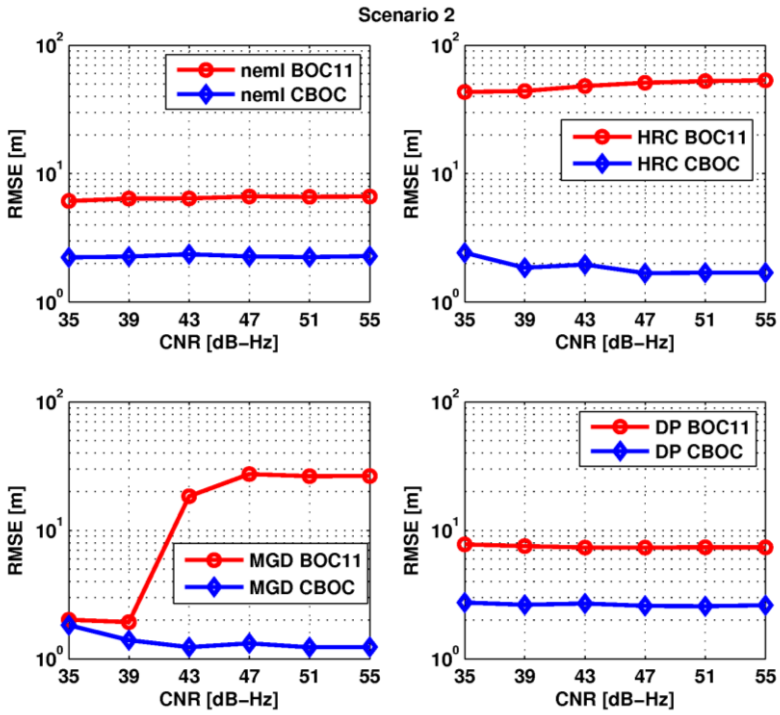

Figure 9.14: Comparison between SinBOC tracking and CBOC tracking with nEML. HRC, MGD and Dot Product in scenario 2

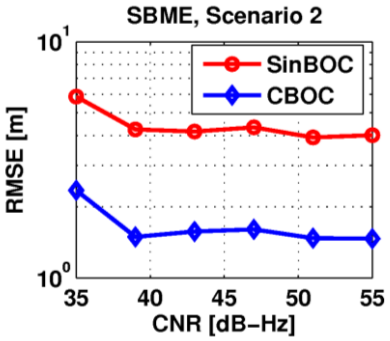

Figure 9.15: Comparison between SinBOC tracking and CBOC tracking with SBME in scenario 2

The tracking performance of nEML, HRC, MGD, Dot Product and SBME with both SinBOC(1,1) reference code and CBOC reference code in terms of RMSE are shown in Figure 9.14 and Figure 9.15. It can be noticed that the tracking with CBOC reference code brings big benefit than that with SinBOC(1,1) reference code in two-path scenario. The Figure 9.16 gives a good explanation of why CBOC can provide such a big improvement. The correlation function shape between the CBOC modulated transmitted signal and CBOC modulated reference code is much narrower than that with SinBOC(1,1) modulated reference code. The presence of Line-Of-Sight (LOS) signal (first peak in blue dash curve) and Non Line-Of-Sight (NLOS) signal (second peak in blue dash curve) is distinguished.

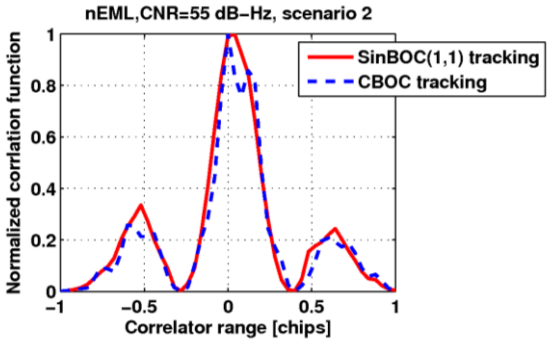

Figure 9.16: Comparison of correlation function between tracking with SinBOC(1,1) tracking and CBOC tracking with nEML in scenario 2

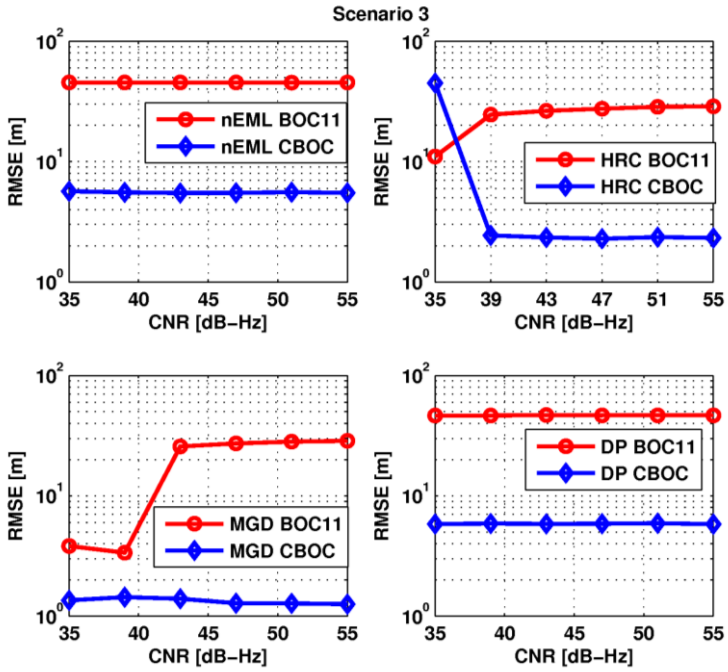

Figure 9.17: Comparison between SinBOC(1,1) tracking and CBOC tracking with nEML, HRC, MGD and Dot Product in scenario 3

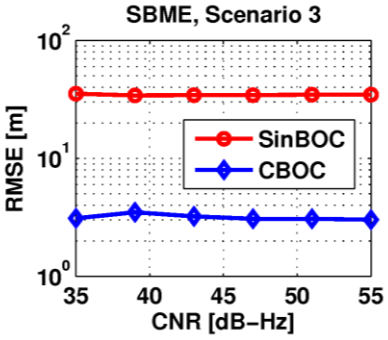

Figure 9.18: Comparison between SinBOC(1,1) tracking and CBOC tracking with SBME in scenario 3

The simulation results shown in Figure 9.17 and Figure 9.18 obtained with scenario 3 also show better tracking performance with CBOC reference code. Similarly, when at least two channel paths exist, the discriminator can still allocate the LOS path signal (the first small peak in blue dash curve) as shown in Figure 9.19. However, there is a risk of locking to false peak when the NLOS signal is strong, because the peak of LOS signal in the correlation function can be covered by the NLOS signal.

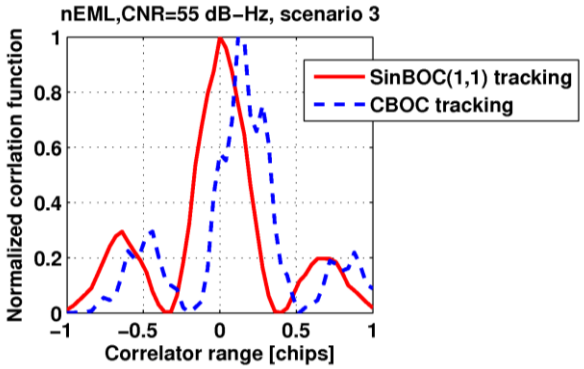

Figure 9.19: Comparison of correlation function between tracking with SinBOC(1,1) tracking and CBOC tracking with nEML in scenario 3

9.5 Impact of switching architecture on tracking performance

In this section, the impact of switching architecture on tracking performance is shown. In order to avoid the impact from other error sources, for example, the multipath and heavy noise, the simulations have been done in a single path scenario and in relatively good conditions, which means that the CNR is set at higher than 30 dB-Hz. The nEML is used in the simulations. The simulation results are shown in terms of RMSE and tracking error variance as can be seen in Figure 9.20 and Figure 9.21.

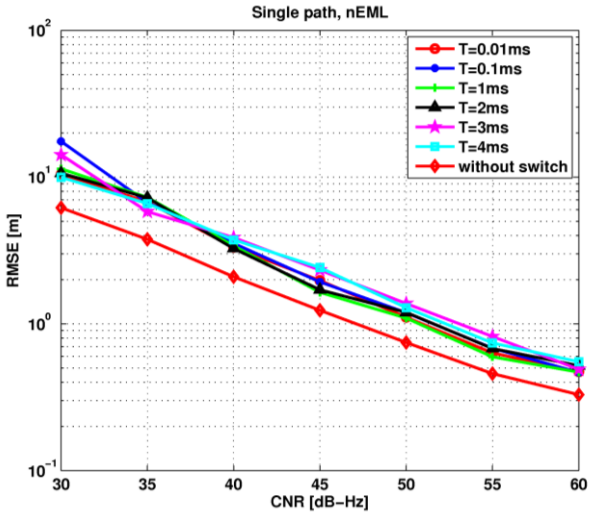

Figure 9.20: Comparison between different switching time and without switching in terms of RMSE

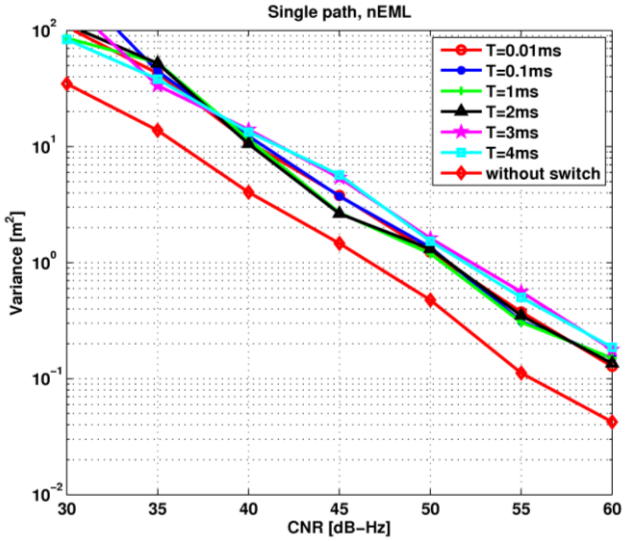

Figure 9.21: Comparison between different switching time and without switching in terms of code tracking error variance

The simulation results in Figure 9.20 and Figure 9.21 indicate that we lost at least 5 dB CNR no matter how long the switching period is. Among the results with different switching time, the switching for each 1 ms gives the best results. The switching at 3 ms and 4 ms show the worst performance. This is because the Galileo E1 receiver does the integration every 4 ms, if the switching is done at 3 ms or 4 ms, every other integration point has only little signal or even only noise, which degrades the tracking performance. In general, the RMSE for 1 ms switching time is slightly smaller than the other switching time slots.

10. Conclusions and future works

This chapter summarizes the main contributions of this thesis work and draw conclusions on the findings obtained from the research results. Besides, the future continuation of this research work is discussed.

10.1 Conclusions

In this thesis, we analyzed some representative code tracking algorithms for CBOC modulated Galileo OS signal. The optimum MGD parameters for MBOC modulated signal and the best normalization factor for nEML, HRC and MGD were found. All these algorithms were tested from the tracking point of view for CBOC modulated Galileo OS signal and their performance were evaluated in a Simulink model developed at TUT in terms of RMSE and tracking error variance. In addition, this thesis work enhanced the development of the original Simulink model, which can be used directly in the future work.

Regarding the results related to the normalization factor, it was shown that the normalization by the sum of early and late correlator gave the best results, especially in low CNR condition.

Concerning the results related to the MGD parameter optimization for MBOC modulated signals, as presented in Chapter 5, under infinite front-end bandwidth assumption, the MGD with optimum coefficients is better than HRC and nEML. Under limited front-end bandwidth assumption, for low bandwidths, MGD is slightly better than nEML, but the gap is not significant. For high bandwidths, MGD and HRC outperform the nEML while having a very similar performance.

When we assessed the performance of studied code tracking algorithms in terms of RMSE and tracking error variance, we noticed that the two-stage estimator out-performed all the other considered code tracking algorithms. The two-stage estimator combines the noise resistant property of nEML and the significant multipath mitigation performance of HRC in good conditions. The use of nEML on the first stage decreases the possibility of locking to a false point compared with the case when only HRC was used.

The presence of the pilot component helps a receiver to have variable choices to track the data or the pilot component, or both. The results in Section 9.2 showed that the tracking pilot-only channel has the smallest tracking error, but the combined data/pilot tracking has the best performance from the tracking error variance point of view.

However, if only one channel is to be used, the tracking pilot-only channel is the best choice.

The impact of code tracking loop bandwidth on the tracking performance is significant as the results shown in Section 9.3. Considering both scenarios in the simulations, the nEML code tracking error is proportional to the loop bandwidth. Therefore, the loop bandwidth can be set to a relatively small value (e.g., 1 Hz) when nEML is used. When HRC or MGD is used, the loop bandwidth can be set at 3 or 4 Hz.

Considering the results of tracking with MBOC modulated reference code shown in Section 9.4, we drew the following conclusions. The tracking with reference CBOC showed better tracking performance as compared with the reference SinBOC(1,1) receiver, with high front-end bandwidth.

The simulation results about the use of the switching architecture showed that the receiver can receive E1 and E5 signals periodically but we lose about 5 dB CNR compared with the results without switching. Therefore, it indicates the fact that the switching architecture degrades the tracking performance, and also at the same time increases the complexity of the receiver design.

10.2 Future research works

The two-stage estimator used in this thesis is a combination of nEML and HRC. It has the properties of nEML and HRC, which makes it outperform the other code tracking algorithms. The work can be continued by analyzing the possibility to combine other delay tracking algorithms in order to exploit the benefits of different algorithms. The CNR estimator used with the two-stage estimator is one of the CNR estimation algorithms used in literature. The use of other CNR estimator, such as moment based CNR estimator, is also worthy to test in the continuation.

Although the switching architecture degrades the code tracking performance and increases the receiver design complexity, it shows the possibility of dual frequency receiver design. It may require new tracking algorithms to compensate the losses.

The signal used in this thesis is the Galileo E1 (E1B and E1C) signal. The study regarding other Galileo signals, such as E5 signals can be a topic of further investigation.

Finally, although the Simulink model was designed to meet the realistic condition, it would be beneficial to test the delay tracking algorithms with real or measured data.

Bibliography

[1] G.W.Hein, J.A.Avila-Rodriguez, S.Wallner, A.R.Pratt, J.Owen, J. L. Issler, J.W., "MBOC: The new optimized spreading modulation recommended for Galileo L1 OS and GPS L1C," in *Position, Location, And Navigation Symposium, 2006 IEEE/ION*, pp.883– 892, April 2006.

[2] Avila-Rodriguez, J.A. Hein, G.W., Wallner, S., Issler, J.L., Ries, L., Lestarquit, L., De Latour, A., Godet, J., Bastide, F., Pratt, T., Owen, J., "The MBOC Modulation-A Final Touch for the Galileo Frequency and Signal Plan," in *http://www.insidegnss.com/node/174*, Sep/Oct 2007.

[3] A.V.Dierendonck, P.Fenton, and T.Ford, "Theory and performance of narrow correlator spacing in a GPS receiver," *Journal of the Institute of navigation*, pp. 265–283, vol. 39, Fall 1992.

[4] D.Skournetou and E.S.Lohan, "Non-coherent multiple correlator delay structures and their tracking performance for Galileo signals," in *in Proc. of International Conference on ITS Telecommunications (ITST)*, May/Jun 2007, Switzerland.

[5] G.A. McGraw, Rockwell Collins, M.S.Braasch, "GNSS Multipath Mitigation Using Gated and High Resolution Correlator Concepts," *ION NTM 1999*, pp. 333-342, Jan 1999.

[6] P.Enge, "GPS Modernization: Capabilities Of the New Civil Signals," in *In Invited Paper for the Australian International Aerospace Congress*, Brisbane, Aug. 2003.

[7] E.D.Kaplan and C.J.Hegarty., *Understanding GPS: Principles and Applications*, 2nd ed.: Artch House, 2006.

[8] P.Misra and P.Enge, *Global Positioning System - Signals, Measurements and Performance*.: Ganga-Jamuna Press,Massachusetts, 2006.

[9] "Galileo Open Service Signal In Space Interface Control Document,OS SIS ICD (2008), Draft 1," http://www.gsa.europa.eu/go/galileo/os-sis-icd.

[10] European Space Agency, "Galileo-The European Programme for Global Navigation Services," http://www.hellas-sat.net/files/file/EU_galileo_brochure.pdf, 2003.

[11] G.W.Hein, J.Godet, J.L.Issler, J.C.Martin, P.Erhard, R.L.Rodridues,T.Pratt, "Status of Galileo frequency and signal design," in *In Proc. of ION GPS*, pp.266-277, Sept. 2002.

[12] "L1 band part of Galileo signal in Space ICD (SIS ICD)," , www.galileoic.org/la/files/Galileo%20OS%20SIS%20ICD%20230506.pdf.

[13] B. Eissfeller, G. Ameres, V. Kropp, and D. Sanroma, "Performance of GPS, GLONASS and Galileo," *Eissfeller et al.*, p. 185–199, 2007.

[14] M.F.Samad, "Effects of MBOC Modulation on GNSS Acquisition Stage," Department of communication engineering, Tampere University of Technology, Finland, Master's thesis August 2006.

[15] K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, *A Software-Defined GPS and Galileo Receiver: A single-Frequency Approch*. Birhkhäuser Boston, 2007.

[16] J.W.Betz, "The Offset Carrier Modulation for GPS modernization.," in *Proc.of ION Technical Meeting*, pp.639-648, Jun. 1999.

[17] E.S.Lohan, A.Lakhzouri, M.Renfors, "Binary-offset-carrier modulation techniques with applications in satellite navigation systems," *Wireless Communications and Mobile Computing, Volume 7, Issue 6 (p 767-779), 2006*.

[18] [Online]. http://electromaniacs.com/content/view/201/9/

[19] Hein G, Irsigler M, rodriguez JA, Pany T., "Performance of Galileo L1 signal candiates," in *In CDROM Proceeding of European Navigation Conference GNSS,May 2004*.

[20] E.S.Lohan and M.Renfors, "Correlation Properties of Multiplexed-BOC (MBOC) Modulation for Future GNSS Signal," in *European Wireless Conference, Paris, France,April 2007*.

[21] J.A.Avila-Rodriguez, S.Wallner, G.Hein, E Rebeyrol, O.Julien, C. Macabiau,L. Ries, A. DeLatour, L. Lestarquit, and J. Issler., "CBOC-An Implementation of MBOC," *First CNES Workshop on Galileo Signals and Signal Processing, Toulouse,France, October 2006*.

[22] M.Fantino, M.Gianluca, P.mulassano, M.Pini, "Performance Analysis of MBOC, AltBOC, and BOC Modulation in Term of Multipath Effects on the Carrier Tracking Loop within GNSS receiver," *Aerospace and Electronic System Magazine, IEEE*, Volum 12, Issue 11, Page 4-10, Nov.2008.

[23] J.M.Sleewaegen, W.De Wilde, M.Hollreiser, "Galileo AltBOC Receiver," in *May 17, 2004, ENC-GNSS 2004, Rotterdam*.

[24] M.Zhodzicshsky, V.Veitsel, S.S.Novatisky, and F.Kamgar L.Garin, "Signal correlation technique for a receiver of a spread spectrum signal including a pseudo-random noise code that reduces errors when a multipath signal is present," US Patent 6,272,18.

[25] D. de Castro, J. Diez, and A. Fernandez, "A New Unambiguous Low-Complexity BOC Tracking Technique," in *in Proc. of ION GNSS*, 2006.

[26] H.Hurskainen, E.S.Lohan, X.Hu, J.Raasakka, and J.Nurmi, "Multiple gate delay tracking structures for GNSS signals and their evaluation with simulink, systemC, and VHDL," in *International Journal of Navigation and Observation*, vol. 2008, Article ID 785695, 17 pages, 2008.

[27] X.Hu and E.S.Lohan, "GRANADA validation of optimized Multiple Gate Delay structures for Galileo SinBOC(1,1) signal tracking," in *ITST Proceedings, (Sophia Antipolis,France)*, Jun. 2007.

[28] J.M.Sleewaegen, and F.Boon, "Mitigating short-delay multipath: a promising new technique," , Proceedings of ION GPS-2001, Sept. 2001.

[29] M.Z.H.Bhuiyan, E-S.Lohan, and M.Renfors, "A Slope-Based Multipath Estimation Techniques for Mitigating Short-Delay Multipath in GNSS Receivers," , accepted in Special Session on "Circuits, Systems and Algorithms for Next Generation GNSS" at ISCAS 2010, Paris, France.

[30] B.W.Parkinson, J J.Spiker Jr, *Global Positioning system: Theory and Application Volume 1*.

[31] J.W.Betz and K.R.Kolodzieski, "Extended Theory of Early-Late Code Tracking for a Bandlimited GPS Receiver," in *to be published in Navigation: Journal of the Institute of Navigation*, Fall 2000.

[32] "IRGAL Final Workshop," , http://www.irgal.com, Sep 2008.

[33] M. Restle G.Heinrichs, "A Novel Galileo/GPS Navigation Software Receiver," in *20th International Technical Meeting of the Satellite Division of The Institute of Navigation, ION GNSS 2007,* Fort Worth, Texas USA, September 2007.

[34] C.O'Driscoll, M.Petovello, and G.Lachapelle, "PLAN Group's Software-Based Receiver: Current Status, Ongoing Work and Ultra-Tight GNSS/INS Integration," in *ION Alberta Section Meeting, http://www.ion-alberta.org/meetings,* Oct. 2007.

[35] T.Pany, D.Sanroma, J.Avila-Rodrigues, B.Eissfeller, F.Forster, ""High precision signal processing in a PC software receiver"," *Aerospace and Electronic Systems Magazine, IEEE,* vol. 21, no. 8, pp. 20-25, Aug. 2006.

[36] I.Joo, J.E.Lee, S.Lee, J.H.Kim ,D.W.Lim, S.J.Lee, "S/W based IF Signal Simulator Prototyping for L1 C/A, L2C, and E1(B&C)," in *IEEE ICICS 2007.*