POLITECNICO DI TORINO

SCUOLA INTERPOLITECNICA DI DOTTORATO

Research Doctorate Course in Electronics and Communications Engineering
XXVI cycle

Final Dissertation Submitted to the Faculty of Engineering

Design of pilot channel tracking loop systems for high sensitivity Galileo receivers

Nazelie Kassabian

Tutor
prof. Letizia Lo Presti

Co-ordinator of the Research Doctorate Course
prof. Ivo Montrosset

February 2014

PhD thesis reviewers:

1. Mark Petovello
Email: mark.petovello@ucalgary.ca
www.ucalgary.ca/petovello/

2. Yu Morton
Email: mortonyt@miamioh.edu
www.users.muohio.edu/mortonyt/

Members of examining board:

1. Prof. Renato De Mori
Email: renato.demori@univ-avignon.fr
http://www.cs.mcgill.ca/people/faculty/profile?uid=renato

2. Prof. Mario Gerla
Email: gerla@cs.ucla.edu
http://www.cs.ucla.edu/ gerla/

3. Prof.ssa Anna RICHELLI
Email: anna.richelli@ing.unibs.it
http://www.ing.unibs.it/ richelli/

Day of the defense: February 27, 2014

Abstract

Global Navigation Satellite Systems (GNSS) have been in the center stage of the recent technological upheaval that has been initiated by the rise of smartphones in the last decade. This is clearly reflected in the development of many applications based on GNSS technology as well as the emergence of multi-constellation GNSS with the launch of the first Galileo satellites at the end of the year 2011. GNSS does not only guarantee global positioning, navigation and timing services but also extends to applications in banking, agriculture, mapping, surveying, archaeology, seismology, commerce, ionosphere scintillation monitoring, remote sensing (soil moisture, ocean salinity, type of surface), wind speed monitoring, ocean surface monitoring, altimetry and many others. In the last decade, Location Based Services (LBS) have increased significant market demand where GNSS has been coupled with technologies based on terrestrial communication links in order to meet strict positioning accuracy requirements. In these conditions, relying on GNSS technology alone, raises a few challenges for signal synchronization even before positioning attempts and are mainly due to a considerable signal attenuation as it propagates through construction material and into indoor environments. Ionosphere scintillation induces a similar challenge where in addition to amplitude fading, the carrier phase and frequency suffer from indeterministic fluctuations.

This research activity is devoted to explore and design the elements constituting pilot channel scalar tracking loop systems, specifically tailored to Galileo signals. It is expected that running such systems with extended integration intervals offers robust synchronization of the incoming signal which is heavily affected by external indeterministic fluctuations. In some conditions, it is desired to follow these fluctuations as in ionosphere scintillation

monitoring while in other instances it is mainly desired to filter them out as noise to guarantee positioning capabilities. This is the objective of this research study which applies for both indoor environments and ionosphere scintillation affected signals. Towards this endeavor, a comprehensive theoretical study of the carrier and code tracking loops elements is undertaken, and particular attention is directed to the following aspects:

- carrier frequency and phase discriminators and the relative optimum integration time
- Galileo specific code discriminators and code tracking architecture especially tailored to Composite Binary Offset Carrier (CBOC) modulated signals
- optimum loop filters designed in the digital domain for different types of phase input signals
- local signal generation using a numerically controlled oscillator and loop filter estimates
- front-end filter bandlimiting effects on the tracking performance

This design is further tested with simulated Galileo signals with and without ionosphere scintillation as well as raw Galileo signals in an equatorial region during March 2013. Tracking performance comparison is carried out between the customized Galileo receiver developed in this research activity and an ionosphere scintillation dedicated professional GNSS receiver, the Septentrio PolaRxS PRO ®receiver.

To all the great people I met throughout my life

"We demand rigidly defined areas of doubt and uncertainty!"
Douglas Adams, The Hitchhiker's Guide to the Galaxy

Acknowledgements

I would like to acknowledge the many diverse people I had the honor and fortune to know and work with during my PhD research activity. You have added something special in this work. Some people were working behind the scenes and believed in me, even if I didn't know them personally. Thank you all:

- Politecnico di Torino, the Scuola Interpolitecnica di Dottorato (SIPD) and the Italian Ministry of Education which have constantly offered me their financial support through a generous scholarship and an opportunity to conduct research abroad as well as take part in many conferences and seminars nationally and internationally.
- My supervisor in Politecnico di Torino, Prof. Letizia Lo Presti who believed in my potential right from the start. Thank you for your kind support and sympathy. Your open attitude to discuss many aspects of this research study have been enlightening for my professional development.
- My supervisor in Miami University, Prof. Jade Morton for your infinite kindness, useful advices and availibility despite your busy schedule. Thank you for making my stay very productive and incredibly pleasant at Miami University. Your great attitude, immense sense of cooperation and warm hospitality is forever engrained in my mind.
- My parents without whom I would have never reached this point. Your unconditional love, endless support and infinite dedication have transmitted to me the way to be in touch with life's inmost secret. Thank you for letting me fly towards Earth's calling dream.

- My Italian family who has been a generous and endless support throughout my stay in Italy. Thank you for your hospitality and your efforts to make me feel at home. Learning Italian would have never been so easy without the stimulating and interesting conversations we had together dear Mariella.
- Mario, for your kindness, unending friendship, and constant encouragement towards life-building experiences. Thank you for opening up my horizons and bringing joy and spiritual wealth into my life through your constant suggestions for the most extraordinary music, books, stories, movies, art, photos, sites, gastronomy and even the most exquisite pastry (our beloved "cannolo torinese") that can be found in the world!
- All my friends, old and new, especially Antonio, Sebastiano, Nicoletta, Sabrina, Shadi, Aaron, Steve, Martina, Maurizio and Mario. I have had so much fun with you, learned from you and was inspired by you, thank you for your invaluable friendship.
- All the researchers in the Navsas group of Politecnico di Torino and ISMB, thank you for your collaboration, advices, pleasant conversations, lunches and laughs we shared. I also learned a lot from you.
- All the students I interacted with during my PhD study abroad period in Miami University. A special and warm thank you goes to my friend Steve whose integrity, cooperation and enlightening conversations I deeply valued. Your endless support, divine patience, warm company and the magic walks in and around Miami Campus hiking trails together with Aaron have been inspiring to me during some difficult moments.
- A very special thank you goes to you Martina Narineh, you have been an invisible supportive friend (and much more) across the Atlantic despite your busy schedule, constantly pushing me towards my next achievement through your simple and wise inputs. My personal growth rate has been quadrupled if not more ! I am forever grateful for your presence in my life.

Contents

List of Figures

List of Tables

Glossary

Roman Symbols

$\hat{f}[k]$	estimated NCO frequency at epoch k
$\hat{f}_0$	estimated frequency output from acquisition engine or FLL
A_1	zero delay or constant loop filter coefficient
A_2	single delay loop filter coefficient
A_3	double delay loop filter coefficient
A_D	Doppler rate
B_L	normalized noise equivalent bandwidth
B_N	noise equivalent bandwidth
c	speed of light
C/N_0	Carrier-to-noise ratio at RF front-end output
$C_E[n]$	early version of spreading code
$C_L[n]$	late version of spreading code
$C_P[n]$	prompt version of spreading code
$c_s[n]$	locally generated PRN code
$C_{2\phi_m}$	ratio of the NBD and NBP measurements at epoch m
D	inherent delay inside locked loop

$D(z)$	phase detector transfer function in z domain
$d[n]$	binary data sequence holding navigation message
d_s	early minus late code spacing in chips
d_{sc}	early minus late subcarrier spacing in chips
$e_f[k]$	estimated frequency by frequency discriminator
$e_p[k]$	estimated phase by phase discriminator
$e_u[k]$	estimated phase by unambiguous frequency aided discriminator
$F(z)$	loop filter transfer function in z domain
f_0	incoming signal true carrier frequency
f_c	spreading code chip rate
f_D	Doppler frequency
F_L	loop sampling frequency
F_S	signal sampling frequency
f_ϕ	phase detector function
F_{IF}	Intermediate Frequency
f_{sc}	subcarrier frequency
$G_1(f)$	PSD of BOC(1,1) signal
$G_6(f)$	PSD of BOC(6,1) signal
$G_s(f)$	PSD of MBOC(6,1,1/11) signal
h	receiver height over reflecting surface
$H(z)$	closed locked loop system transfer function
$H_e(z)$	closed locked loop system error transfer function

I	Ionosphere error in code and carrier phase measurements
$I[k]$	in-phase correlation value at epoch k
I_E	early in-phase point correlator
I_L	late in-phase point correlator
I_P	prompt in-phase point correlator
I_{CE}	early spreading code in-phase point correlator
I_{CL}	late spreading code in-phase point correlator
I_{CP}	prompt spreading code in-phase point correlator
I_{SE}	early subcarrier in-phase point correlator
I_{SL}	late subcarrier in-phase point correlator
I_{VE}	very early in-phase point correlator
I_{VL}	very late in-phase point correlator
K	number of integration intervals over which the CNR is estimated
K_B	Boltzmann constant
K_D	phase detector gain
L_c	correlation loss due to front-end bandlimiting
L_ϕ	two-sided PD linear tracking region
M	number of integration intervals over which the NBP and WBP are estimated
N	integer number of cycles between satellite and user receiver
$N(z)$	NCO transfer function in z domain
N_0	PSD value of AWGN on incoming signal
$n_A[n]$	instantaneous AWGN on incoming signal

N_i	PSD of input phase noise component
$n_I[k]$	instantaneous noise on in-phase correlator value
$n_Q[k]$	instantaneous noise on quadra-phase correlator value
$N_W(z)$	apparent additive noise term in the phase detector in z domain
$n_W[k]$	apparent additive noise term in the phase detector in time domain
$N_F(z)$	frequency NCO transfer function
$N_{PR}(z)$	phase and rate NCO transfer function
$N_{RO}(z)$	rate only NCO transfer function
N_{VEMLE}	normalization value of the code delay discriminator
NBD_m	Narrowband difference of in-phase and quadra-phase correlator outputs at time t_m
NBP_m	Narrowband power measurement at time t_m
NP_m	ratio of the NBP and WBP measurements
P	incoming signal total received power
$Q[k]$	quadra-phase correlation value at epoch k
Q_E	early quadra-phase point correlator
Q_L	late quadra-phase point correlator
Q_P	prompt quadra-phase point correlator
Q_{CE}	early spreading code quadra-phase point correlator
Q_{CL}	late spreading code quadra-phase point correlator
Q_{CP}	prompt spreading code quadra-phase point correlator
Q_{opt}	functional on which optimum loop filter is based upon
Q_{SE}	early subcarrier quadra-phase point correlator

Q_{SL}	late subcarrier quadra-phase point correlator
Q_{VE}	very early quadra-phase point correlator
Q_{VL}	very late quadra-phase point correlator
r	true range in meters between satellite and local user receiver
$R(.)$	correlation function between the incoming and the locally generated PRN signals
S	code phase discriminator function
$s[n]$	incoming received signal
S_4	Scintillation amplitude index
$s_c[n]$	subcarrier signal
$S_E[n]$	early version of subcarrier
S_I	Signal intensity
$S_L[n]$	late version of subcarrier
$S_P[n]$	prompt version of subcarrier
S_{4N_0}	Scintillation amplitude index due to ambient noise
SNR_C	post-correlation signal to noise ratio
T	Troposphere error in code and carrier phase measurements
T_c	spreading code period
T_g	Signal simulation time
T_I	pre-detection integration interval
T_L	loop sample interval
T_S	analog to digital sampling period
T_{eff}	Effective RF front-end temperature

T_{sc}	subcarrier subchip period
w_n	loop filter natural frequency
WBP_m	Wideband power measurement at time t_m
Z_E	complex early correlator
Z_L	complex late correlator
Z_P	complex prompt correlator
$G(z)$	open loop transfer function in z domain
$H_N(z)$	locked loop system noise transfer function
$H_{eN}(z)$	locked loop system error noise transfer function

Greek Symbols

α	generalized parameter that accounts for phase input and ξ
β	double-sided complex bandwidth
$\Delta\hat{\dot{\phi}}[k]$	estimated NCO phase rate correction
$\delta\omega$	true error in estimated discrete frequency
$\delta\omega_{max}$	maximum frequency range satisfying linearity
$\delta\phi$	true phase error between incoming and estimated phase
δt_s	satellite clock bias
δt_u	user receiver clock bias
ϵ_ϕ	multipath and receiver noise error in code phase measurements
ϵ_ρ	multipath and receiver noise error in code phase measurements
$\hat{\dot{\phi}}[k]$	estimated NCO phase rate
$\hat{\phi}(z)$	estimated NCO phase in z domain

$\hat{\phi}[n]$	estimated NCO phase in time domain
$\hat{\phi}_k$	estimated NCO total phase at kth epoch mid-interval
$\hat{\phi}_k[n]$	estimated NCO instantaneous total phase during epoch k
$\hat{\phi}_k^e$	estimated NCO total phase at end of kth interval
$\hat{\phi}_k^{re}$	estimated NCO total phase at end of kth interval updated with the most recent phase rate value
$\hat{\tau}_c$	estimated spreading code delay
$\hat{\tau}_{sc}$	estimated subcarrier delay
$\hat{\varphi}$	estimated NCO partial phase
λ	signal wavelength
ϕ	carrier phase range measurement
$\phi(z)$	true carrier phase in z domain
$\phi[n]$	true carrier phase in time domain
ψ	satellite elevation angle
ρ	code phase range measurement
σ_N^2	output phase noise variance due to input noise only
σ_ϕ	square root of the tracking jitter at the output of a PLL
σ_τ	square root of the tracking jitter at the output of a DLL/SLL
τ	spreading code delay
θ_e	receiver dynamic stress error
$\varphi[n]$	instantaneous incoming signal true partial phase
ξ	parameter to adjust tradeoff between noise power and transient energy
ζ	loop filter damping ratio

$\delta\phi_k$ true error in estimated phase at kth epoch mid-interval

μ_{NP} average ratio of the NBP and WBP measurements

ϕ_k true phase value at kth epoch mid-interval

$\sigma_{\Delta\phi}$ Scintillation phase index

Acronyms

A-GNSS Assisted-GNSS

A-GPS Assisted-GPS

ACQ Acquisition

ADC Analog to Digital Converter

ADR Accumulated Doppler Range

AltBOC Alternative Binary Offset Carrier

ARNS Aeronautical Radio Navigation Services

ASIC Application Specific Integrated Circuits

ASPeCT Autocorrelation side peak cancellation technique

ATAN Two-quadrant Arctangent

ATAN2 Four-quadrant Arctangent

AWGN Additive White Gaussian Noise

BCP Basic Cross Product

BJ Bump Jumping

BOC Binary Offset Carrier

BPSK Binary Phase Shift Keying

CAF Cross-Ambiguity Function

CBOC Composite Binary Offset Carrier

CDMA Code Division Multiple Access

CNR Carrier-to-Noise Ratio

COH Coherent

CP Cross Product

CU Continuous Update

DATAN Differential Two-quadrant Arctangent

DD Decision Directed

DDCP Decision Directed Cross Product

DDDP Decision Directed Dot Product

DDLL Digital Delay Locked Loop

DE Double Estimator

DFLL Digital Frequency Locked Loop

DFT Discrete Fourier Transform

DLL Delay Locked Loop

DOIT satellite receiver Dynamics, receiver Oscillator phase error, Ionosphere and Troposphere errors

DP Dot Product

DPLL Digital Phase Locked Loop

DSLL Digital Subcarrier Locked Loop

DU Discrete Update

EGNOS European Geostationary Navigation Overlay System

EML Early Minus Late

EMLE Early Minus Late Envelope

EMLP Early Minus Late Power

FDMA Frequency Division Multiple Access

FE Front-End

FLL Frequency Locked Loop

FOC Full Operational Capability

FPGA Field Programmable Gate Arrays

GAGAN GPS Aided Geo Augmented Navigation

GLONASS GLObalnaya NAvigatsionnaya Sputnikovaya Sistema or Global Navigation Satellite System

GNSS Global Navigation Satellite Systems

GPS Global Positioning System

ICD Interface Control Document

IF Intermediate Frequency

IIR Infinite Impulse Response

IRNSS Indian Regional Navigation Satellite System

LBS Location Based Services

LF Loop Filter

LHCP Left Hand Circularly Polarized

LNA Low Noise Amplifier

LPF Low Pass Filter

MAT Mean Acquisition Time

MAX Maximum

MAX/TC Maximum Threshold Crossing

MBOC Modified Binary Offset Carrier

MGD Multiple Gate Discriminator

MSBAS Multi-functional Satellite Based Augmentation System

NBD NarrowBand Difference

NBP NarrowBand Power

NCO Numerically Controlled Oscillator

NRZ Non Return to Zero

OS Open Service

P2P Peer to Peer

PD Phase Detector

PDF Probability Density Function

PDI Pre Detection Integration Interval

PGF Probability Generating Function

PI Proportional Integral

PLI Phase Lock Indicator

PLL Phase Locked Loop

PPP Precise Point Positioning

PRM Power Ratio Method

PRN Pseudo-Random Noise

PRS Public Regulated Service

PSD Power Spectral Density

PVT Position Velocity Time

QPSK Quadrature Phase Shift Keying

QZSS Quasi Zenith Satellite Systems

RAT Ratio

RF Radio Frequency

RG Reference Generator

RHCP Right Hand Circularly Polarized

RMS Root Mean Square

RNSS Radio Navigation Satellite Services

SBAS Satellite Based Augmentation Systems

SCC SubCarrier Cancellation

SDR Software Defined Radio

SI Signal Intensity

SIS Signal In Space

SLL Subcarrier Locked Loop

SNR Signal-to-Noise Ratio

SoL Safety-of-Life

SSB Single Side Band

SV Space Vehicle

TC Threshold Crossing

TOA Time Of Arrival

TTC Telemetry Tracking and Control

TTFF Time To First Fix

UFA Unambiguous Frequency Aided

UHF Ultra High Frequency

ULS Up-Link Station

USRP Universal Software Radio Peripheral

VEML Very Early Minus Late

VEMLE Very Early Minus Late Envelope

VSM Variance Summing Method

WAAS Wide Area Augmentation System

WBP WideBand Power

Chapter 1

Introduction

The discovery of navigation, the process of following a set of directions to travel from one point to another has been present since ancient times. According to Chinese storytelling, the compass was discovered and used in wars during foggy weather before recorded history [1]. Navigation entails positioning or localization which is the process of determing the position, velocity and orientation of an object [2].
In order to navigate, it is necessary to have a reference or a map besides positioning. Maps have been used through ancient times as well, starting from marking trees, or leaving traces, such as stone references and mountains. Later in time, pieces of parchment or clay tablets have been used to pass the spatial information of a region from one person to another. These are now called maps. The first recorded maps date from the Mesopotamian ancient civilizations 5000 years ago.
We have come a long way ever since, using satellites as a means for navigation but not only. In particular, satellite based navigation started in early 1970s with the Global Positioning System (GPS) program after three U.S. projects were explored including the U.S. Navy Navigation Satellite System also known as Transit, the U.S. Navy's Timation (TIMe navigATION) system and the U.S. Air Force project 621B [1]. Each of these systems experimented separately with the Doppler shift effect on a continuous wave signal, an atomic clock to predict satellite orbits and reduce ground control update rate and a Pseudo-Random Noise (PRN) signal to modulate the carrier frequency. In the same years, the Russian initiative was emerging and resulted in GLObalnaya NAvigatsionnaya Sputnikovaya Sistema (GLONASS). Although the early motivations and initiatives of such satellite based navigation were strictly coming from the military

side, there has been a smooth transition towards providing civilian services as well as Safety-of-Life (SoL) and commercial services. In fact, the recent trend of evermore decreasing chip size with an increase of processing power as captured by Moore's law, has made the Location Based Services (LBS) an attractive and commercially promising undertaking. As a consequence, indoor navigation has received considerable attention in these last years.

Moreover, remote sensing is another field where Global Navigation Satellite System (GNSS) has been rapidly gaining grounds in recent years. The latest trend is focused on the ionosphere scintillation impact not only on the final GNSS positioning performance, but on the ability of GNSS receivers to maintain synchronization with the incoming signals in order to solve the navigation equations. In this respect, it is also possible to assess ionosphere scintillation severity through scintillation indices that are basically measurements by the GNSS receiver as a product of its synchronization process. Ionosphere scintillation monitoring by multi-GNSS constellations is a very interesting subject indeed.

In both of these cases, the challenge is to maintain lock or synchronization with the incoming signal which is oftentimes buried in noise. The signal processing gain through correlation processes is crucial for this goal, and increasing the coherent integration interval over which the incoming signal is considered is the only solution for such a feat. Therefore, the constituent elements of GNSS scalar tracking loops which are a basic control instrument to maintain carrier phase and code synchronization, have to be carefully designed to yield the desired estimated response.

1.1 Background and Motivation

GNSS is a highly complex navigation satellite system infrastructure based on three segments: the space segment, the controlling ground segment and the user segment. GNSS main function is to provide global (all over planet Earth) positioning and timing capabilities through Time Of Arrival (TOA) measurements. GNSS does not only guarantee global positioning, navigation and time services but also extends to applications in banking, agriculture, mapping, surveying, archeology, seismology, commerce, ionosphere scintillation monitoring, remote sensing (soil moisture, ocean salinity, type of surface), wind speed monitoring, ocean surface monitoring, altimetry and many others.

In the last decade, LBS have met significant demand in the market where GNSS has been coupled with technologies based on terrestrial communication links in order to meet strict positioning accuracy requirements.

The reason behind this bundle of technology fusion is the shortcomings faced by the GNSS when operating indoors where the typical signal power falls from -130 dBm to -150 to -160 dBm which is approximately equivalent to a fall from 44 dB-Hz down to 24 and 14 dB-Hz [3]. In these conditions, a notable coherent gain through an increase of the coherent integration interval is necessary to perform accurate carrier and code synchronization also known as tracking. The objective of this kind of synchronization lies in following or tracking the behavior of the received signal (both in terms of PRN code and carrier). Tracking is performed by periodically correlating a local signal with the incoming one and generating a correction to apply on the locally generated signal. When the Signal-to-Noise Ratio (SNR) is lower than a certain threshold, it is highly probable that the correction signal due to the correlation process is not reliable. In this case, the receiver loses lock and a coarse tracking or acquisition of the signal is needed. Acquisition is a coarse estimation counterpart of tracking and precedes it. A false acquisition results in a penalty time that is incurred by carrier/code tracking loops which are unable to lock. The Mean Acquisition Time (MAT) is a receiver acquisition performance metric which takes into account this penalty time. This metric is an important parameter that defines the receiver performance given the fact that acquisition can form a bottleneck for the start of the rest of the signal processing blocks that depend on it. The challenges faced by real GNSS signals during the tracking stage are its amplitude fading (indoor/ionosphere scintillation conditions) as well as its carrier phase fluctuations.

In conclusion, the motivation of this research activity is the ever-increasing importance of using GNSS signals in harsh conditions. This importance stems from the growth of indoor LBS catalyzed by market need and its potential due to the launch of multiple international GNSS. Reliable GNSS services guaranteed by robust tracking can also pave the way to ionosphere scintillation monitoring from a network of stations distributed across the Earth with an emphasis around the equatorial and polar regions. Robust

GNSS tracking would also yield early identification of unexpected and abnormal activity of the Sun affecting the ionosphere. Guided by these motivations, this thesis will undertake the design and analysis of the components comprising GNSS digital scalar tracking loops with a focus on the Galileo Open Service (OS) pilot channels. This in turn aims to provide robust and reliable tracking of weak GNSS signals in unfavorable conditions characterized by low signal power or Carrier-to-Noise Ratio (CNR) and scintillation phenomena where both the amplitude and phase of the incoming signal experience indeterministic fluctuations.

1.2 Literature Review and Limitations of Previous Work

The limitations of previous work concentrate on the design of the tracking loop filters in analog domain and transformation to the digital domain instead of directly designing in the digital domain. Although very recently the trend has been shifting towards a design of tracking loops in the digital domain [4]-[5], the design of the phase detector has been ignored in these studies by assuming it a simple constant gain independent of the incoming CNR. In fact, it has been shown in [6] that the discriminator gain and noise propagation through the discriminator can have a radical impact on the overall performance of tracking loops especially in low CNR conditions. In addition, there is no closed form solution relating the noise equivalent bandwidth of the loop filter to the appropriate loop filter parameters. Nonetheless, there has not been any attempt to bridge this gap by approximating the curve that defines the relationship between the design specification and loop filter parameters by a linear piece-wise function.

In this thesis, Galileo receivers are the main focus and as such, Galileo specific code tracking loops and in particular Galileo specific code discriminators or phase detectors receive special attention. Galileo E1 OS signal features a multipeaked auto-correlation function given its Composite Binary Offset Carrier (CBOC) nature. In this setting, ambiguous tracking is a major threat encountered by Galileo E1 receivers inducing a position error of the order of 150 meters. Previous literature has presented several methods, each with its set of pros and cons, to tackle this issue. Some methods attempt to get rid of the complexity brought about by the subcarrier signal, losing out the accuracy that comes with it. The single side lobe or Single Side Band (SSB) technique [2]

and the SubCarrier Cancellation (SCC) technique [7] are listed in this category. On the other hand, other methods seek the enhanced code tracking accuracy offered by these subcarriers. These methods include Binary Offset Carrier (BOC) tracking with multiple gate discriminators Multiple Gate Discriminator (MGD) [8], the bump jumping algorithm, the Autocorrelation side peak cancellation technique (ASPeCT) algorithm [9], the Double Estimator (DE) as explained in [2] - [10] and the two-step Galileo CBOC tracking algorithm [11]. The bump jumping algorithm has the downside of taking too long to converge while the ASPeCT and dual estimator which has a third tracking loop dedicated to the subcarrier both yield a more complex loop implementation.

1.3 Research Objectives

The main objective of this thesis is to design GNSS specific scalar carrier/code tracking loops tailored for pilot channel use and for extended integration intervals. Increasing the integration interval to several tens of milliseconds is crucial in some applications such as urban canyons, indoor environments, as well as ionosphere scintillation scenarios characterized by a low signal-to-noise ratio and low phase fluctuations.

In this respect, various carrier frequency/phase discriminators are to be analyzed in-depth in order to select an appropriate integration interval to be used together with the optimum Phase Detector (PD) given a certain scenario and an input SNR. Moreover, novel Galileo code discriminators are to be designed to exploit the full potential offered by CBOC modulated signals without adding much complexity. In fact, considering that both SSB and SCC do not either fully or efficiently exploit the subcarrier presence, and that Bump Jumping (BJ), ASPeCT and DE are either not efficient or add too much complexity, the MGD is considered to be a good alternative. The focus will be directed in the design of a four taps Very Early Minus Late Envelope (VEMLE) discriminator that yields an unambiguous curve by using two pairs of early and late correlators, i.e. the usual early and late correlators and two additional very early and very late correlators. In addition, the DE is to be explored designing appropriate discriminator functions for the delay and subcarrier locked loops.

Another objective is to provide a methodology for loop filter design that minimizes

both the energy in the transient response of the loop error and the loop output thermal noise power or tracking jitter, taking into account various discriminator models as mentioned previously. In addition, the derivation of the loop filter expression and parameters are to be derived following an approximate linear function that relates the latter with the loop filter design metrics, such as the integration interval and loop filter noise equivalent bandwidth. The theoretical performance of such loop filters is to be assessed and compared with classical digital loop filters designed in the analog domain. The robustness of the design of such GNSS carrier/code tracking loops is in turn to be verified by Galileo simulated and real signals as well as ionosphere scintillation affected signals.

1.4 Thesis Outline

An introduction and history of navigation and satellite based navigation systems has been put forward in this chapter. The current scope of GNSS services targeted for civilian use and their respective challenges are presented, mainly indoor navigation and tracking of ionosphere scintillation affected GNSS signals. The motivation and background surrounding these challenges as well as previous work and its limitations are explored. Based on these limitations, the research objectives are formulated and carried out in the next chapters.

Chapter 2 offers a global view of the modern multi-constellation GNSS infrastructure, frequency plan and its diverse applications. A special attention is given to the European GNSS initiative known as Galileo, where its OS signals and properties are studied. The basic correlation operation in GNSS receivers is studied and synchronization performance metrics are presented. Ionosphere scintillation indices or scintillation estimation metrics are also defined.
The main contribution of this chapter is the implementation of a Galileo OS signal and ionosphere scintillation software simulation tool along with several design parameters to be specified to control the resulting output signal. Finally, a general overview of the typical signal processing blocks of a GNSS receiver is presented starting from the Front-End (FE) down to the tracking stage.

Chapter 3 introduces the basic concepts to be considered for robust tracking of GNSS and specifically Galileo OS signals. The design accounts for weak signals with a low CNR where long integration intervals is a must for successful tracking in terms of carrier frequency/phase as well as code delay. The main considerations are in terms of the phase detector (PD), the loop filter and the Numerically Controlled Oscillator (NCO), all of which are designed in digital domain.

In this respect, various coherent and noncoherent carrier frequency and phase discriminators are analyzed in-depth. A performance comparison is carried out for different SNR inputs to yield the optimum integration interval for a specific discriminator, taking into account the corresponding frequency/phase range that it is able to estimate reliably.

In addition, code phase discriminators specific to Galileo CBOC signals are explored. A MGD structure is considered to be a good alternative considering that other techniques in the literature do not either fully or efficiently exploit the subcarrier presence or add too much complexity. A two-steps normalized four taps VEMLE discriminator is presented that yields an unambiguous curve by using two pairs of early and late correlators, i.e. the usual early and late correlators and two additional very early and very late correlators.

Moving to the loop filter component of scalar tracking loops, a comprehensive set of loop filters that are considered in literature are collected. These classical loop filters are designed in the analog domain and then mapped to the digital domain. Due to the shortcomings of such filters when coupled with long integration intervals, a detailed derivation of the optimum loop filters designed in the digital domain is undertaken to minimize the phase output noise power and transient energy induced by different phase input signals (phase step, frequency step and frequency ramp). Moreover, a simple line approximation is performed to yield these optimum loop filters given a set of desired loop filter specifications and integration intervals.

The theoretical performance of such optimum loop filters is then assessed and compared with classical digital loop filters designed in the analog domain. The stability degree is tested through Bode plots and root locus plots where the loop gain is chosen to be the varying parameter. Gain and phase margins are also visited to analyze the system degree of stability for a range of frequencies. Finally, the performance of all the aforementioned loop filters is compared through simulation of the designed GNSS

scalar tracking loops designed all along this chapter.
The last analysis in this chapter concerns the receiver bandlimiting effects on the tracking performance in terms of both carrier and code phase thermal noise tracking jitter where the main focus is on the CBOC and Binary Phase Shift Keying (BPSK) modulated Galileo E1 and E5a/b OS signals.

Chapter 4 considers two experimental applications of the GNSS pilot channel tracking loop design presented in Chapter 3. It applies the acquired concepts on simulated Galileo OS signals affected by ionosphere scintillation amplitude and phase fluctuations. A performance assessment is carried out in terms of frequency and phase tracking error given that the signals are simulated and the correct frequency and phase values are known at each integration interval. Furthermore, Galileo real signals collected in the equatorial region of Ascension Islands are tested with the aforementioned design of Galileo scalar tracking loop. The second part of the chapter presents results obtained during a reflectometry experiment where carrier phase measurements are used in order to perform altimetry measurements of a flying aircraft on top of rivers in the Italian Piemonte region.

Finally, appendices A and B explore the MAT performance metric in standard and collaborative Peer to Peer (P2P) networks. An intuitive technique to derive the MAT is developed for the serial search case where a special algorithm that follows a specific search order is developed and analyzed. Different search strategies are considered in appendix B and a comparison of standard versus P2P acquisition engines is performed after choosing the best search strategy relative to each case.

Moreover, here is a list of publications that are the fruit of my PhD study:

- Technique for MAT analysis and performance assessment of P2P Acquisition Engines [12] published on IEEE.
- Mean acquisition time of GNSS peer-to-peer networks [13] published on IEEE.
- First Joint GPS/IOV-PFM Galileo PVT estimation using carrier phase measurements [14].

- A flight in the Piedmont region for water surface detection and altimetry experimentation [15] published on IEEE.
- Galileo Tracking Performance Under Ionosphere Scintillation [16] won the best paper award in the Fourth International Colloquium on Scientific and Fundamental Aspects of the Galileo Programme, session 3A: Ionosphere 2.
- Extended Integration Time for Galileo Tracking Robustness Under Ionosphere Scintillation [17] published on IEEE.
- Assessment of Galileo OS Signals and Tracking Algorithms during Equatorial Ionosphere Scintillation [18] to be submitted to the IEEE transactions on AES.

Chapter 2

GNSS, signals and receivers

This is an introductory chapter to present a global view of the modern GNSS, its infrastructure both in space and on ground as well as the spectrums used and diverse applications that stem from GNSS. In addition, the European GNSS initiative known as Galileo is the main focus of this work and thus Galileo OS signal structure is studied together with its spectrum and auto-correlation properties. The basic correlation operation that lies behind the successful synchronization of the GNSS signals is explained, together with standard metrics used to assess the resulting performance. Ionosphere scintillation indices or scintillation estimation metrics are also presented. A Galileo OS signal and ionosphere scintillation software simulation tool is implemented with several design parameters to be specified to control the resulting output signal. In addition, a general overview of the typical signal processing blocks of a GNSS receiver is presented starting from the FE down to the Position Velocity Time (PVT) computation block.

2.1 Satellite Navigation Systems

GNSS is a general term to indicate a global navigation satellite system that provides continuous positioning, navigation, timing and many other capabilities over the entire terrestrial globe. It encompasses three large segments that define any GNSS system, mainly the space, ground and user segments.

The space segment comprises a constellation of satellites orbiting the Earth at a certain altitude and orbital characteristics such as orbital inclination, orbital plane and nominal revolution period. These satellites are equipped with highly specialized instruments

including very stable atomic clocks that drive the signals transmission and which form the core of the satellite navigation concept. The main functions of the space segment are [19]:

- Constantly generating and transmitting code and carrier phase signals.
- Storing and broadcasting the navigation message uploaded by the control segment.

The space segment is undergoing extensive modernization and in particular the two current legacy systems. The U.S. GPS system is in constant modernization, and the Russian GLONASS system is restoring its system to a full 24-satellite constellation with plans to transmit Code Division Multiple Access (CDMA) instead of Frequency Division Multiple Access (FDMA) signals and considering a new frequency band (L5/G3). On the other hand, the two new GNSS systems represented by the European Galileo and the Chinese Compass systems have started launching their satellites in recent years. It is estimated that in 2020 when all four of these global navigation systems reach their full deployment, a total of 113 satellites will be present in space [20].

In addition to global navigation satellite systems, there are regional navigation satellite systems providing regional coverage and designed for general use, such as the Indian Regional Navigation Satellite System (IRNSS), the Japanese Quasi Zenith Satellite Systems (QZSS), and the Chinese Beidou-1 which will evolve from the regional system into the global system Beidou-2 or COMPASS. Moreover, aviation-specific regional systems represented by Satellite Based Augmentation Systems (SBAS) systems such as the European European Geostationary Navigation Overlay System (EGNOS), the U.S. Wide Area Augmentation System (WAAS), the Japanese Multi-functional Satellite Based Augmentation System (MSBAS), and the Indian GPS Aided Geo Augmented Navigation (GAGAN) offer extra satellites that offer better signal accuracy, availability and integrity for civilian aviations users.

The ground segment is the brain of the GNSS system. It has the following functions [19]:

- Control and maintain the status and orbital configuration of the satellite constellation.

- Predict the ephemeris or satellite position information as well as satellite clock evolution.
- Keep track of the GNSS time scale.
- Update the navigation message to be uploaded to the satellites.

Each of the aforementioned four GNSS systems has a specific ground segment structure, but let us mention that the Galileo Full Operational Capability (FOC) phase will involve two redundant ground control centers, five Telemetry Tracking and Control (TTC) stations, nine Mission Up-Link Stations (ULS) and a world-wide network of Galileo sensor stations. Further details on the Galileo ground segment as well as the structure of each of the remaining three GNSS ground segments can be found in [19].

In conclusion, the user segment is comprised of GNSS receivers which process the transmitted signals, compute pseudoranges which are estimates of the distance between the receiver antenna and the satellite. The next step is to solve the navigation equations and compute their relative position and very accurate time to finally provide navigation services by the use of appropriate maps. The basic elements of a GNSS receiver include a dedicated single or multiple frequency antenna, a radio front-end, a baseband signal processing module, and a human/machine interface. Further details are found in Section 2.4.

2.1.1 GNSS concept

GNSS satellites transmit code modulated carrier phase signals. Systems based on CDMA transmission use the same carrier frequency but use a different PRN code sequence to enable the user receiver to distinguish between different satellites signals. Through various signal synchronization modules including acquisition, tracking and data demodulation, it is possible to compute the time it takes for the satellite transmitted signal to reach the user receiver. This time interval is converted to a distance range by assuming a constant signal travel speed equal to the speed of light. In a three-dimensional world, it is possible to use three such measurements to construct three spheres and deduce two intersecting points. However, one of the intersecting points is close to the Earth's surface while the other is not and the user receiver position can easily be deduced. Moreover, the time that is measured on board of satellites

is not synchronized to the time that is measured by the user receiver, and thus a fourth measurement obtained by a fourth satellite is necessary to deduce the user position estimate with an acceptable error. This is possible thanks to the knowledge and tracking of the satellites position in space.

2.1.2 Frequency plans

As previously mentioned, the GNSS space segment is going through an extensive modernization process. The GPS system which currently occupies the three frequency bands denoted by L5, L2 and L1 bands as shown in Figure 2.1, launched the new Block II-R modernized satellites as early as 26 September 2005. These satellites transmit the new military signal and the more robust civil signal L2C. By 28 May 2010, one of the GPS Block II-F modernized satellites was launched which was designed to transmit the third civil signal on the SoL L5 frequency band. The GPS Block III satellites on the other hand are designed to transmit the fourth civil signal, the L1C signal on the L1 frequency band.

The GLONASS system which currently occupies the three frequency bands L5/G3, G2 and G1 bands as shown in Figure 2.1, launched the second generation GLONASS-M (or Uragan-M) satellites as early as 2001. These satellites started transmission of a second civil signal on the G2 band. By 26 February 2011, one of the GLONASS-K (or Uragan-K) third generation satellites were launched which was designed to transmit CDMA signals for civilian applications on the new L5/G3 band as well as the G2 and G1 bands which also transmit using the FDMA technique.

Figure 2.2 shows that the Galileo system occupies the three frequency bands denoted by E5, E6 and E1 bands with different modulation types and covering different services. The Beidou system also occupies the three frequency bands denoted by B2, B3 and B1 bands as shown in Figure 2.1.

2.2 Galileo OS signal structure

The Galileo OS signals are open source free of charge signals designed to provide competitive position and timing performance for the user community. The Galileo OS

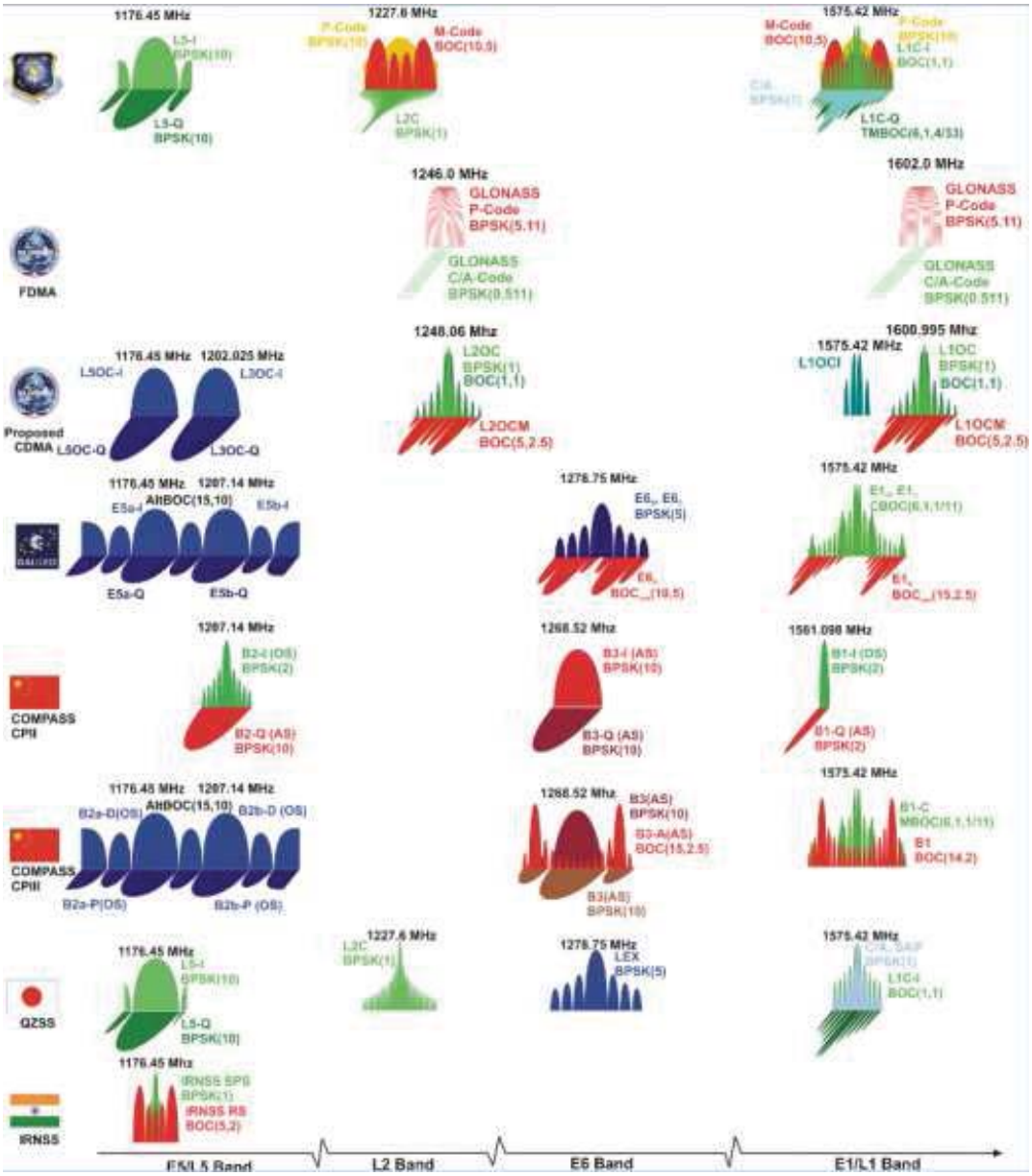


Figure 2.1: GNSS frequency plan and modulations for the GPS, GLONASS, Galileo and Beidou (Compass) systems - showing the legacy and proposed version of GLONASS as well as Compass versions II and III (Source: Stefan Wallner)

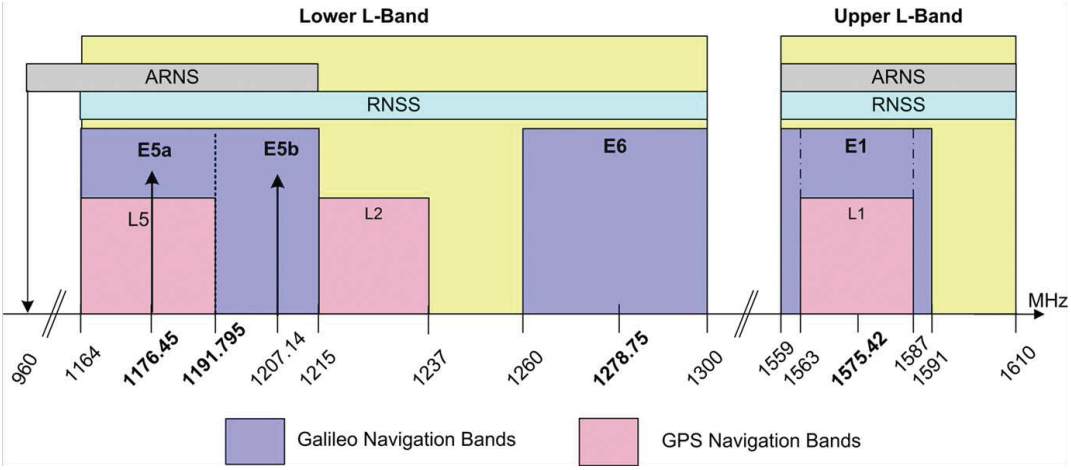


Figure 2.2: GPS and Galileo frequency plan - with respect to Aeronautical Radio Navigation Services (ARNS) and Radio Navigation Satellite Services (RNSS) frequency bands.

mainly consists of three frequency bands, E1 and E5a/b as described in the Galileo OS Interface Control Document (ICD) [21]. The transmitted power on both of these frequency bands is divided onto two channels, the data and pilot channels. Due to plans to extend the integration time to more than one code period, only one of these channels is considered in this thesis, the pilot channel.

2.2.1 Galileo E1 OS signal

The Galileo E1 frequency band is found in the upper L-Band in the allocated spectrum for RNSS as well as the ARNS [21]. It is centered around 1575.42 MHz, the same center frequency as that of the GPS L1 frequency band. The signal modulation scheme for Galileo E1 is shown in Figure 2.3 where $s_{E1}(t)$ is the Galileo E1 OS band-pass signal as transmitted from present and future Galileo satellites. It is modulated by a spreading code $C_{E1-B}(t)$ and a navigation data message $D_{E1-B}(t)$ in the data channel and a different spreading code $C_{E1-C}(t)$ which includes the secondary code in the pilot channel, as well as a weighted sum of two subcarrier signals $\alpha sc_{E1-BOC(1,1)}(t) \pm \beta sc_{E1-BOC(6,1)}(t)$ (higher weight $\alpha = 10/11$ to the lower rate subcarrier and lower weight $\beta = 1/11$ to the higher rate subcarrier).

The combination of these two subcarriers make up the CBOC modulation CBOC(6,1,1/11)

in the time domain. In fact, the data and pilot channel subcarriers are equal to the weighted sum and difference between the narrowband and the wideband signals BOC(1,1) and BOC(6,1) respectively [21]. As a consequence, a certain orthogonality exists to separate the data channel from the pilot channel. The subcarrier and spreading code are perfectly time aligned so that a rising edge in the spreading code corresponds to a possible rising edge in the subcarrier signal, and in any case to a different subchip. This is the cornerstone of tracking the delay of the spreading code and subcarrier signal product considering it a single unique variable for both of the signals. This concept is further discussed in detail in Section 3.4.4.1.

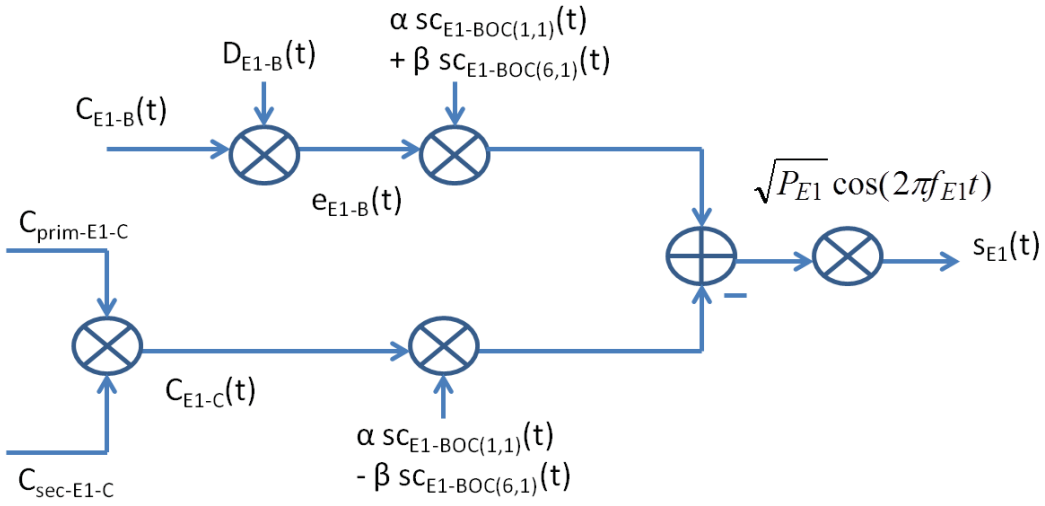


Figure 2.3: Galileo E1 OS modulation scheme - on both data and pilot channels down to the pass-band signal as generated on board Galileo satellites.

In the frequency domain, it is known as the modified binary offset carrier MBOC(6,1,1/11) and its power spectral density Power Spectral Density (PSD) is defined as:

$$G_s(f) = \frac{10}{11}G_1(f) + \frac{1}{11}G_6(f) \tag{2.1}$$

where $G_1(f)$ and $G_6(f)$ are the PSD of the binary offset carrier BOC(1,1) and BOC(6,1) signals. This is the total PSD of the Galileo E1 in-phase channel. In fact, the E1 signal allocates data/pilot channels on the same in-phase I channel while the quadra-phase Q channel is reserved for the Public Regulated Service (PRS) which has a wider frequency

bandwidth as shown in Figure 2.1. The Galileo OS E1 reference bandwidth is specified to be $\beta = 24.552$ MHz as compared to the GPS L1, L5 and Galileo E5a/b reference bandwidths of 20.46 MHz. However, correlation losses due to bandlimiting is discussed in Section 3.6 where it is shown that a two-sided bandwidth of 14.322 MHz is enough to guarantee the improved accuracy brought about by the design of the Galileo E1 ranging code including its data and pilot channel subcarriers.

The Galileo E1 differs from the GPS L1 signal modulation scheme in that it has an additional subcarrier defined in the time domain as CBOC modulation, more specifically CBOC(6,1,1/11). Figure 2.4 shows 5 sampled chips of Galileo E1 signal's pure spreading code relative to PRN 1 in the top, and then the product of that spreading code with the pilot channel subcarrier using a sampling frequency of 16.3676 MHz.

As a consequence of the CBOC subcarrier, and the BOC subcarrier in general, the code auto-correlation function $R(\tau)$ exhibits two side peaks around zero code phase as seen in Figure 2.5. Moreover, it is shown that the Galileo E1 pilot channel exhibits a sharper peak than the corresponding data channel. This is due to the orthogonal pilot channel subcarrier that is in anti-phase with respect to the data channel.
The GPS L1 signal on the other hand, is a CDMA signal as well but is characterized by a triangular shape normalized auto-correlation function as seen in Figure 2.6(b). Figure 2.6 shows the sampled and non-sampled versions of a portion of the GPS CDMA spreading signal relative to PRN 1. In Figure 2.6(a), the first 50 chips are plotted and the relative auto-correlation values for the range of lags between ± 50 chips from the true code delay.
The Galileo E1 OS signal ranging codes are built of primary and secondary codes by using a tiered code construction [21]. The primary code chip rate is 1.023 Mcps, similar to the GPS L1 C/A and GPS L1C signals. The primary code length is 4 ms and its length is 4092 chips. The primary codes of the data/pilot channels are long pseudorandom noise PRN optimized memory codes and are published in the Galileo OS ICD [21]. The secondary code on the pilot channel is unique for all satellites. The data channel is deprived of a secondary code but instead is multiplied by the navigation data, while the pilot channel is deprived of the navigation data and multiplied by a secondary code of length 25 chips and duration 100 ms.

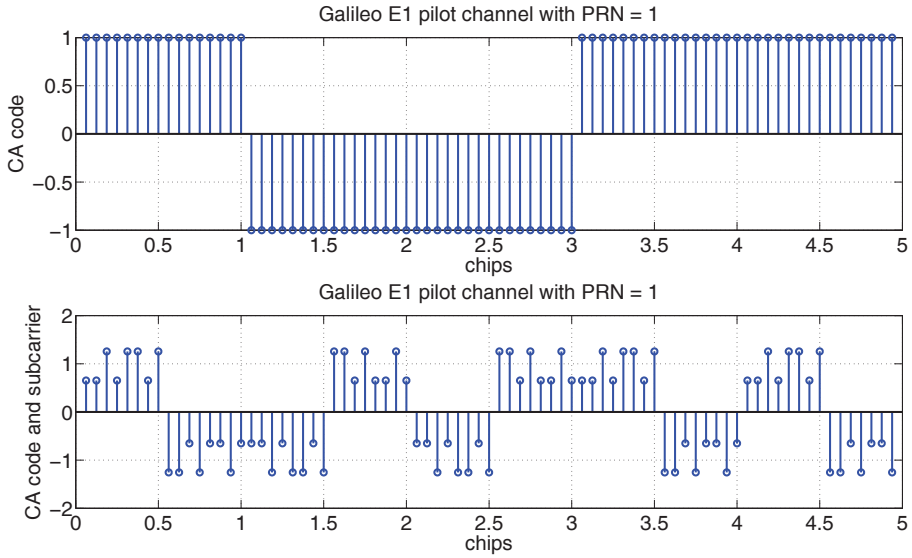


Figure 2.4: Galileo E1 spreading sequence and subcarrier - the CBOC(6,1,1/11) modulation for PRN 1 using a sampling frequency of 16.3676 MHz.

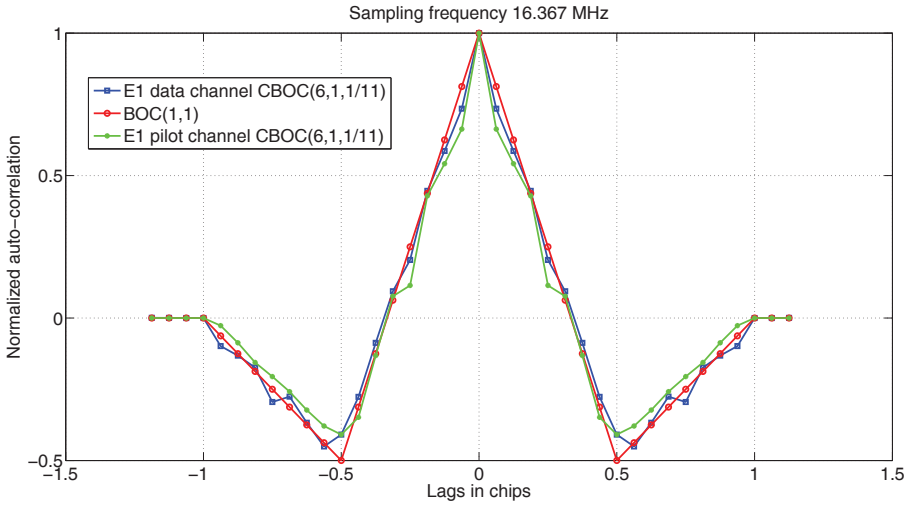
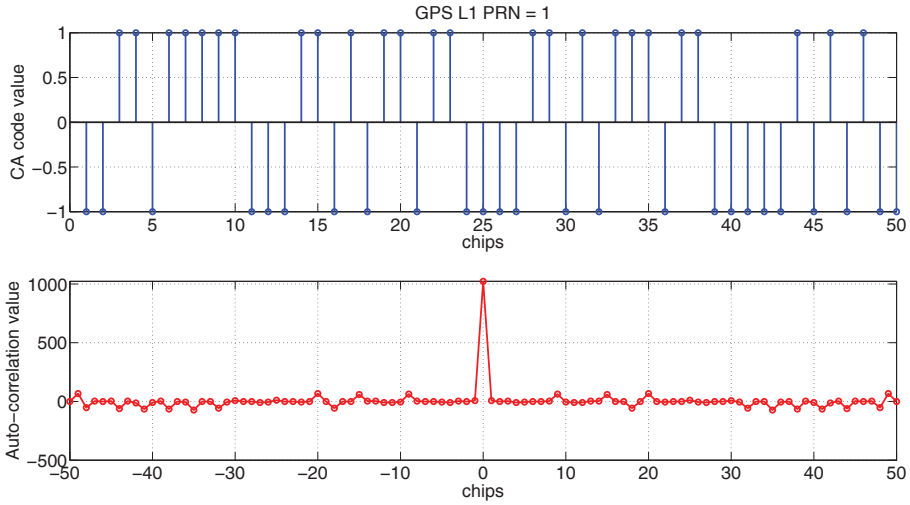
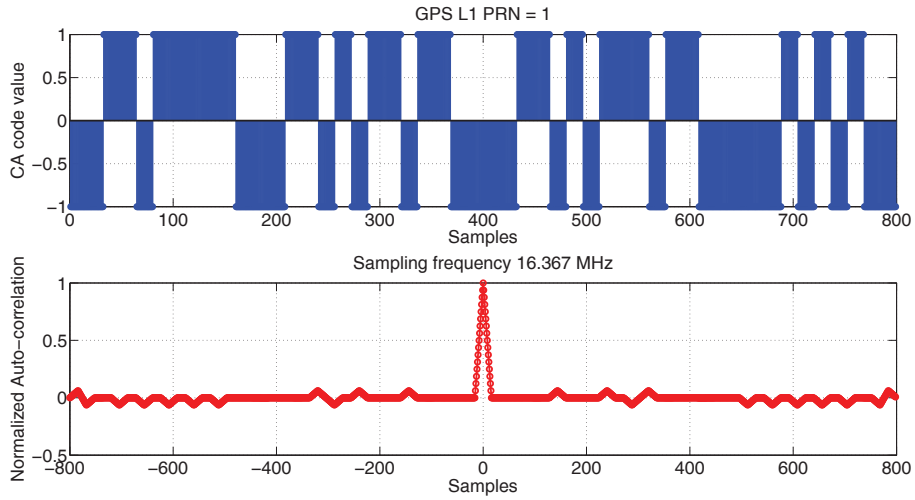


Figure 2.5: Galileo E1 total code auto-correlation function - of PRN 1 showing a few chips around the true code delay and using a sampling frequency of 16.3676 MHz.



(a) GPS L1 CA code and auto-correlation function for the first chips



(b) GPS L1 CA code and auto-correlation function for the first samples

Figure 2.6: GPS L1 CA code and auto-correlation function - GPS L1 CA code for PRN 1 and auto-correlation function in terms of chips and samples

Due to the overlaying secondary code on the primary code, the auto-correlation function of the Galileo E1 pilot channel signal considered over several code periods, presents several side peaks at the boundaries of the secondary code chips as shown in Figure 2.7. The data channel on the other hand shows a periodic nature due to absence of secondary code chips.

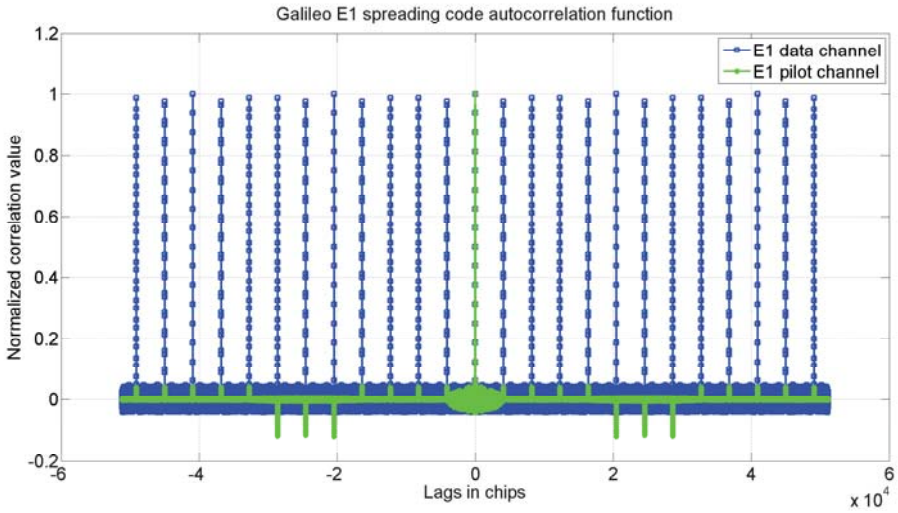


Figure 2.7: Galileo E1 tiered code auto-correlation function - on both data and pilot channels with a sampling rate of 16.3676 MHz.

2.2.2 Galileo E5 OS signal

The Galileo E5a frequency band is found in the lower L-Band and is centered around 1176.45 MHz, the same center frequency as that of the GPS L5 frequency band as can be seen in Figure 2.1. Similarly, it can be seen that the Galileo E5b frequency band is also found in the lower L-Band and is centered around 1207.14 MHz. Both frequency bands have a bandwidth of 20.46 MHz and are situated in the allocated spectrum for RNSS as well as the ARNS used by civil aviation users [21].

If the entire E5 wideband signal is considered as in an Alternative Binary Offset Carrier AltBOC(15,10) modulation, then the existence of subcarriers is imperative. However, the E5a/b signals, when considered as separate frequency bands, are Quadrature Phase

Shift Keying (QPSK) signals, or even BPSK(10) signals when one of the data/pilot orthogonal channels is taken into account. The value 10 is present in BPSK(10) and AltBOC(15,10) because the E5a/b chipping rate is 10.23 MHz. The signal modulation scheme of Galileo E5a signal is shown in Figure 2.8 where $s_{E5a}(t)$ is the Galileo E5a OS band-pass signal as transmitted from present and future Galileo satellites.

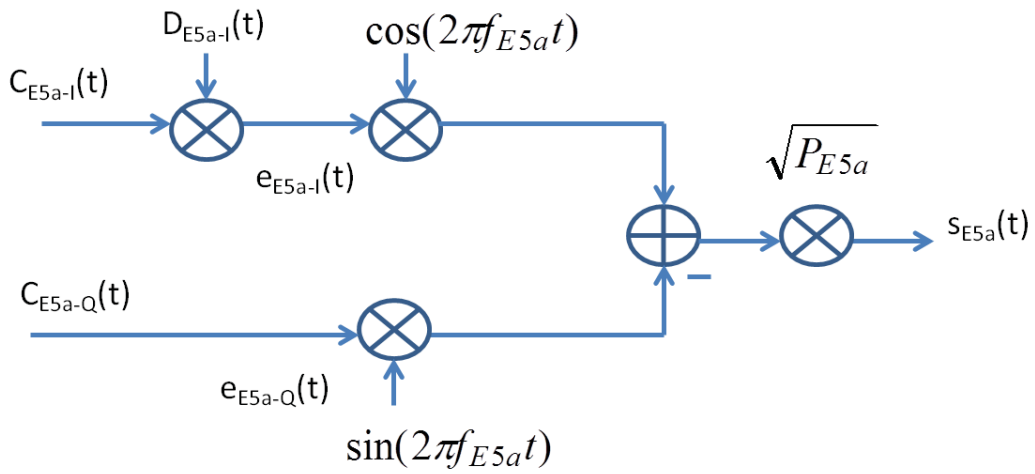


Figure 2.8: Galileo E5a OS modulation scheme - on both data and pilot channels down to the pass-band signal as generated on board Galileo satellites.

The E5a signal is modulated by a spreading code $C_{E5a-I}(t)$ and a navigation data message $D_{E5a-I}(t)$ in the data channel and a different spreading code $C_{E5a-Q}(t)$ in the pilot channel. Both data and pilot channel spreading codes include the secondary code and each of the channels is multiplied by cosine and sine of the center frequency of the E5a band. Due to a chipping rate of 10.23 MHz, the minimum sampling frequency that satisfies the Nyquist sampling theorem at baseband is 20.46 MHz. Moreover, unlike the Galileo E1 OS channels, the E5a/b data/pilot channels are characterized by orthogonality.

Both of these channels primary PRN codes can be generated using linear feedback shift registers and last 1ms while the secondary codes are published in hexadecimal format in [21]. The data channel offers a unique secondary code for all satellites (20

ms for E5a data channel and 4 ms for E5b data channel) while a different secondary code of 100 ms is used for every satellite on the pilot channel.

The auto-correlation function of a single code period of the E5a/b Galileo signals shown in Figure 2.9 is the typical BPSK auto-correlation function (similar to that of GPS L1 C/A code) where a single peak is present at zero phase lag. Due to the overlaying secondary code on the primary code, the auto-correlation function of the Galileo E5a/E5b pilot channel signal considered on several code periods, presents several side peaks at the boundaries of the secondary code chips as shown in Figure 2.10. The Galileo E5a data channel on the other hand shows a periodic nature due to the lower number of secondary code chips with respect to the pilot channel (20 instead of 100) such that the unity peak is repeated 5 times at different code delay values.

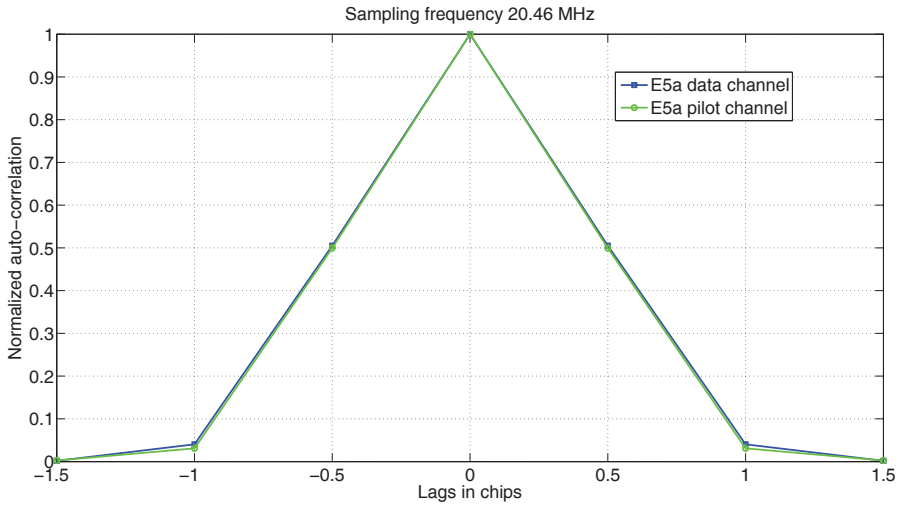


Figure 2.9: Galileo E5a total code auto-correlation function - relative to PRN 1 showing a few chips around the true code delay and using a sampling frequency of 20.46 MHz.

2.3 Simulation/estimation of Galileo signals/CNR and ionosphere scintillation signals/indices

Simulation of Galileo signals follows the same basic scheme used to simulate GNSS signals with some peculiarities mainly the primary/secondary codes specific to Galileo

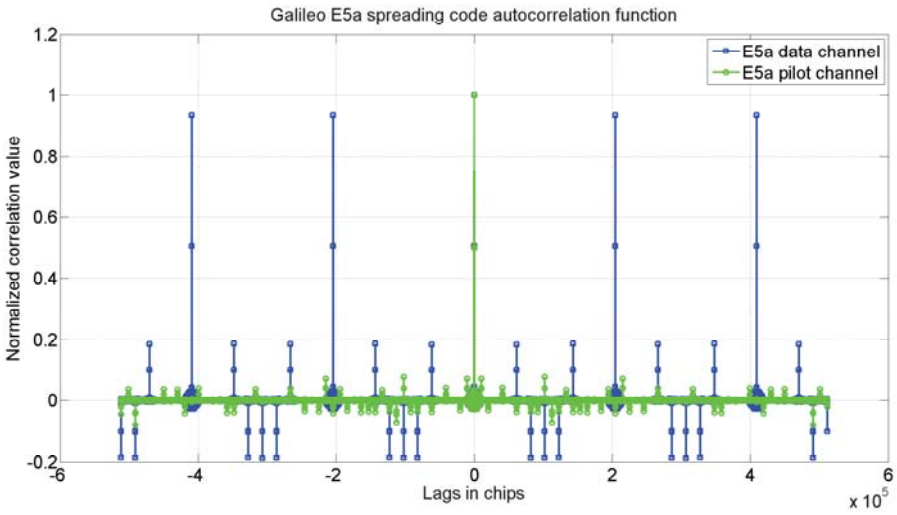


Figure 2.10: Galileo E5a tiered code auto-correlation function - on both data and pilot channels with a sampling rate of 20.46 MHz.

with the corresponding chip rate and code period as well as the characteristic modulation scheme, including the subcarriers, the data/pilot channels and the navigation data bits specified in the Galileo ICD [21]. It is widely known that it is possible to estimate the CNR of the signal at the output of a GNSS receiver antenna, by considering the in-phase and quadra-phase correlator outputs using two techniques Power Ratio Method (PRM) and Variance Summing Method (VSM) widely used in the literature. Ionosphere scintillation signals are simulated in terms of multiplicative amplitude time histories and additive phase time histories using the Cornell model [22]. The scintillation severity is assessed by two scintillation indices S_4 and $\sigma_{\Delta\phi}$ which make use of the in-phase and quadra-phase correlator outputs as well as incoming carrier frequency and phase estimates.

2.3.1 GNSS signal definition

The received or Intermediate Frequency (IF) signal from a single Space Vehicle (SV) or satellite at the input of a GNSS receiver is denoted as $s[n]$ and is generically expressed as:

$$s[n] = \sqrt{P}c_s[n-\tau]e^{j(2\pi f_0 n T_S + \varphi[n])} + n_A[n] \tag{2.2}$$

where P is the total received signal power, $c_s[n-\tau]$ is the spreading code delayed by τ expressed in samples (and which includes a synchronized subcarrier signal $s_c[n-\tau]$ in the case of BOC modulated signals, i.e. Galileo), f_0 is the incoming signal true carrier frequency including the front-end intermediate frequency as well as the Doppler frequency, T_S is the analog to digital sampling period, $\phi[n]$ is the instantaneous carrier partial phase and $n_A[n]$ is an additive Gaussian complex noise term which represents thermal noise and is assumed to be white with a single sided PSD of N_0 Watts/Hz. It is worth mentioning that the binary data sequence holding the navigation message $d[n]$ is present on the data channel or the in-phase part of the signal but has been omitted in this representation for simplicity.

2.3.2 GNSS signal recovery: correlations

In a typical GNSS receiver, the received signal is correlated with a locally generated signal with appropriate code, carrier frequency and carrier phase delay $\hat{\tau}$, $\hat{\omega}$ and $\hat{\phi}$ respectively. As mentioned earlier, the resulting in-phase and quadra-phase correlations are used to measure the SNR as well as to estimate the carrier Phase Lock Indicator (PLI) and CNR, which is a code delay lock indicator. It is worth noting that the SNR is a GNSS receiver estimate by definition and mainly depends on the integration time while the CNR is intrinsic to the input signal power and the Radio Frequency (RF) front-end thermal noise. The in-phase and quadra-phase correlations at each integration interval or epoch k yield [6]:

$$I[k] = \sqrt{P}d[k]R(\tau[k])\text{sinc}\left(\frac{\delta\omega[k]T_I}{2}\right)\cos(\delta\phi[k]) + n_I[k] \tag{2.3}$$

$$Q[k] = \sqrt{P}d[k]R(\tau[k])\text{sinc}\left(\frac{\delta\omega[k]T_I}{2}\right)\sin(\delta\phi[k]) + n_Q[k] \tag{2.4}$$

where $R(.)$ is the correlation function between the incoming and the locally generated PRN signals, $\delta\phi = \phi - \hat{\phi}$ and $\delta\omega = \omega - \hat{\omega}$ are respectively the estimated phase and frequency error at a given epoch k and $n_I[k]$ and $n_Q[k]$ are the instantaneous noise on in-phase and quadra-phase correlations.

It is stated in [23] that the CNR estimates are considered in code phase lock detectors while PLI are used by carrier phase lock detectors. Moreover, [23] states that PLI are normalized estimates of the cosine of twice the carrier phase error and hence the closer

the error is to zero, the closer the PLI is to unity. In fact, it is proven in [24] that phase lock requires that input and output signals or voltages be in quadrature, that is, manually shifted by $\pi/2$ inducing a zero carrier phase error estimate.

2.3.3 Carrier phase and code delay lock indicators

A common metric used in the GNSS literature to express the incoming signal power with respect to noise present in the measurement or processing GNSS receiver is the CNR. It gives an idea of the difficulty and likelihood of successfully using the received signal with acceptable positioning accuracy. Unlike the signal to noise ratio which uses GNSS receiver correlator outputs, discussed shortly after, the CNR is not a measurement by the receiver but rather an expected value given the receiver hardware architecture as well as the antenna incident signal power. The CNR refers to the output of a GNSS receiver's RF front-end and is defined as:

$$\frac{C}{N_0} = 10\log_{10}P - 10\log_{10}N_0 \tag{2.5}$$

where P, as previously mentioned, is the power of the received signal or signal strength and N_0 expressed in W/Hz depends on the effective RF front-end temperature which includes the antenna temperature and the RF front-end temperature and consequently noise figure in the following manner [3]:

$$N_0 = K_B T_{eff} \tag{2.6}$$

Although the CNR is strictly dependent on the receiver and signal characteristics, it is possible to obtain CNR estimates by using correlator outputs to measure the signal power in narrow and wide bandwidths. In fact, there are at least two methods to estimate the CNR [25], the PRM and the VSM both of which are widely used in literature. The PRM uses wideband and narrowband power measurements while the VSM method only makes use of raw power measurements at a rate corresponding to the integration intervals. NarrowBand power (NBP) measurements are obtained by integrating the raw correlator output measurements and squaring at the next step, while the WideBand Power (WBP) measurements consist in integrating the squared values of the correlator outputs. The expressions of NBP_m and WBP_m at time $t_m = m.M.T_I$

with $m = 1, 2, ...$ where M is the number of integration intervals over which they are estimated [23]:

$$NBP_m = \left(\sum_{k=1}^{M} I[k]\right)^2 + \left(\sum_{k=1}^{M} Q[k]\right)^2 \tag{2.7}$$

$$WBP_m = \sum_{k=1}^{M} \left(I^2[k] + Q^2[k]\right) \tag{2.8}$$

It should be noted that the NBP measurements rely on the integration of raw correlator outputs and as such have to be stripped of any modulation including the navigation data bits and secondary code chips. In this thesis, the pilot channel is considered so that data bits are of no concern and the secondary code chips present at each primary code period are wiped off after computing the secondary code delay in the acquisition engine. Integration intervals T_I have to be chosen to be a multiple of the primary code period, i.e. 4 ms and 1 ms for Galileo E1 and E5a/b signals respectively.

The CNR estimate is derived using the ratio of these power measurements NP_m and its corresponding average μ_{NP} in the following manner:

$$\left|\frac{\hat{C}}{N_0}\right|_{PRM} = 10\log_{10}\left(\frac{1}{T_I}\frac{\mu_{NP} - 1}{M - \mu_{NP}}\right) \tag{2.9}$$

where

$$\mu_{NP} = \frac{1}{h}\sum_{m=1}^{h} NP_m \tag{2.10}$$

is the average noise power over $h = K/M$ values, K being a multiple of M representing the number of integration intervals over which the CNR is estimated and the estimated noise power is :

$$NP_m = \frac{NBP_m}{WBP_m} \tag{2.11}$$

The VSM method on the other hand only uses the mean $\bar{P}$ and variance σ_P of the raw power estimates $P_x[k] = I^2[k] + Q^2[k]$ without any accumulation and the CNR estimate is:

$$\left|\frac{\hat{C}}{N_0}\right|_{VSM} = 10\log_{10}\left(\frac{1}{2T_I}\frac{\mu_P}{\mu_N}\right) \tag{2.12}$$

where

$$\mu_P = \sqrt{\bar{P}^2 - \sigma_P} \tag{2.13}$$

and

$$\mu_N = \frac{1}{2}(\bar{P} - \mu_P) \tag{2.14}$$

As previously mentioned the CNR estimate is a code delay lock indicator used inside a code lock detector. Carrier lock detectors on the other hand use PLI which estimates the cosine of twice the carrier phase error by using two variables, the NBP as previously defined and NarrowBand Difference NBD measurements defined as the narrowband power difference of in-phase and quadra-phase correlator outputs at time t_m:

$$NBD_m = \left(\sum_{k=1}^{M} I[k]\right)^2 - \left(\sum_{k=1}^{M} Q[k]\right)^2 \tag{2.15}$$

The carrier PLI is then given at time t_m by the ratio of these two variables:

$$C_{2\phi_m} = \frac{NBD_m}{NBP_m} \tag{2.16}$$

2.3.4 Galileo OS signal simulation

Galileo E1 and E5a/b OS signal (data and pilot channels) simulation routine is developed and written in Matlab language. Taking into consideration the generic expression of the received signal at the input of a GNSS receiver in Equation 2.2, a set of parameters can be specified to generate the desired simulated signals. These parameters include:

- the frequency band to be generated specified by band,
- the intermediate frequency F_{IF},
- the sampling frequency F_S,
- the Doppler frequency f_D,
- the Doppler rate A_D,
- the PRN or SV number,
- the spreading code phase or code delay τ,
- the CNR C/N_0,
- the signal bandwidth,

- the data type (bit4, int, double),
- the time duration over which the signal is to be simulated T_g.

There is also the option to exclude any of the code, carrier and noise signals giving the opportunity to test and analyze the mechanisms at the heart of carrier phase and code delay tracking. Given the aforementioned two-sided reference bandwidth values of 24.552 and 20.46 MHz of the Galileo E1 and E5a/b OS signals, the intermediate frequency should be carefully chosen so as to be greater than the single sided reference bandwidth values 12.276 MHz and 10.23 MHz respectively so as to avoid aliasing with the negative side of the spectrum. The sampling frequency should then be chosen so as to satisfy the Nyquist sampling theorem (twice the bandwidth). The methodology followed in generating the simulated signals is a structured approach where modular functions perform unique duties as shown in Figure 2.11:

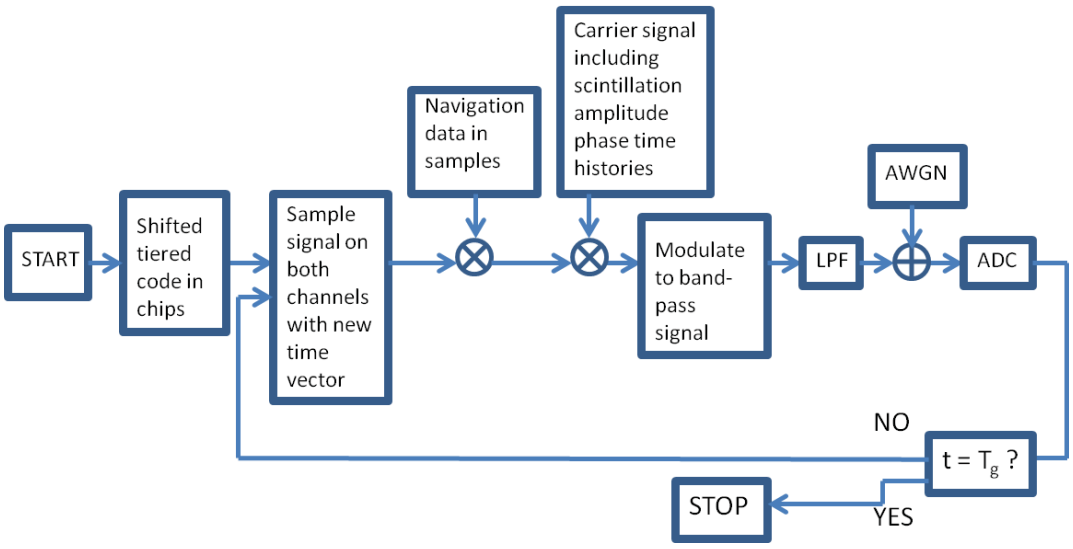


Figure 2.11: Galileo signal simulation scheme including scintillation signal - where the sampling of the tiered code is performed periodically to take into account the Doppler effect, the carrier signal takes into account the scintillation amplitude and phase time histories, goes into a Low Pass Filter (LPF), and an Analog to Digital Converter (ADC) after adding Additive White Gaussian Noise (AWGN).

the first block further shown in detail in Figure 2.12 generates the baseband code in terms of chips over the period of a secondary code which is 100 ms for both E1 and E5a/b pilot channels; the data channel is generated by repeating the primary code

until the same length is reached. Basically, this step involves the multiplication of the primary and secondary codes following a tiered code structure. A Non Return to Zero (NRZ) signal is generated by mapping the logical binary values 0 and 1 to +1 and -1 respectively. The next step for the E1 OS signal is to multiply each chip by the 12 sub-chips of the corresponding CBOC subcarrier. The E5a/b signals skip this step because there is no subcarrier. After that step, it is necessary to shift the code by the complement of the desired code phase, and then to sample with the desired F_S taking into account the desired Doppler frequency and rate by introducing a new time vector. The reason for introducing the code phase shift in terms of chips rather than samples and hence the process of shifting before sampling, is because with a nonzero Doppler frequency, the effective code phase in samples changes at every block of signal simulation, i.e. 100 ms in our case. Finally the data message is generated using the random function and with the specified data rate in the Galileo ICD [21]. Moreover, the front-end is simulated by multiplying the real and imaginary part of the baseband signal by the cosine and sine of the appropriate carrier frequency (including the Doppler frequency and rate) as in Equation 2.2, setting the frequency as:

$$f_0 = F_{IF} + f_D + A_D n T_S \tag{2.17}$$

The last steps involve passing the resulting signal through a Low Pass Filter LPF, adding an AWGN and applying an ADC to limit the signal resolution to a few bits. This procedure is repeated until the time variable reaches the signal simulation period T_g.

In the following, the intermediate frequency is set to 0 Hz and the sampling frequency is chosen to be 25 Msamples/s. A sample noiseless Galileo E1 OS signal is generated setting the input parameters to some arbitrary values, i.e. f_D = 2 kHz, A_D = -0.2 Hz/s, a code delay of 20560 chips (5 secondary chips and 100 primary chips), and a time duration T_g = 2 seconds. Since there is no noise in this signal, it is possible to verify the shape of the MBOC(6,1,1/11) signal spectrum as shown in Figure 2.13. A sample noiseless Galileo E5a signal with similar parameters is generated to verify the E5a PSD in Figure 2.14.

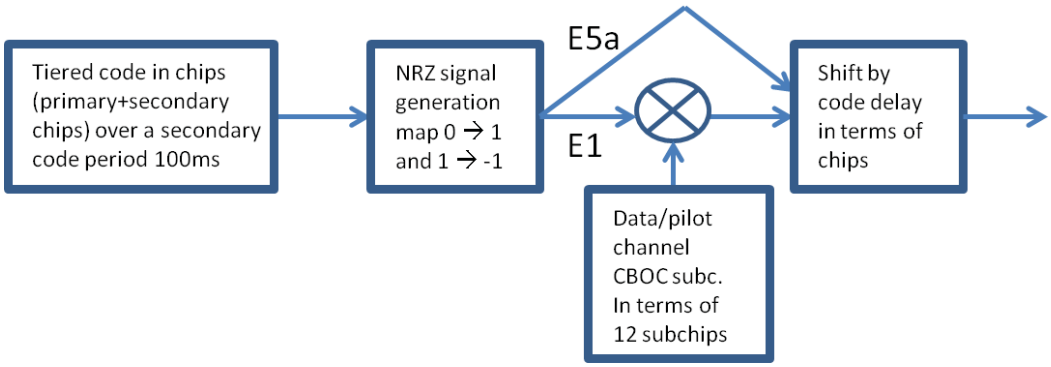


Figure 2.12: Galileo tiered code signal generation - mapping it to a NRZ signal, multiplying by any subcarrier, shifting by any code delay and finally sampling it with a new time vector that includes the Doppler frequency and rate effect.

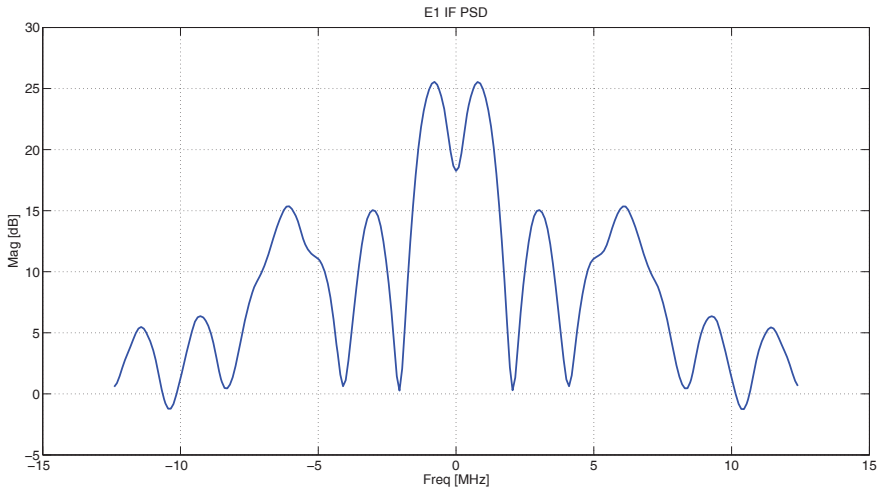


Figure 2.13: Noiseless Galileo E1 OS PSD - over the ICD specified reference bandwidth of 24.552 MHz.

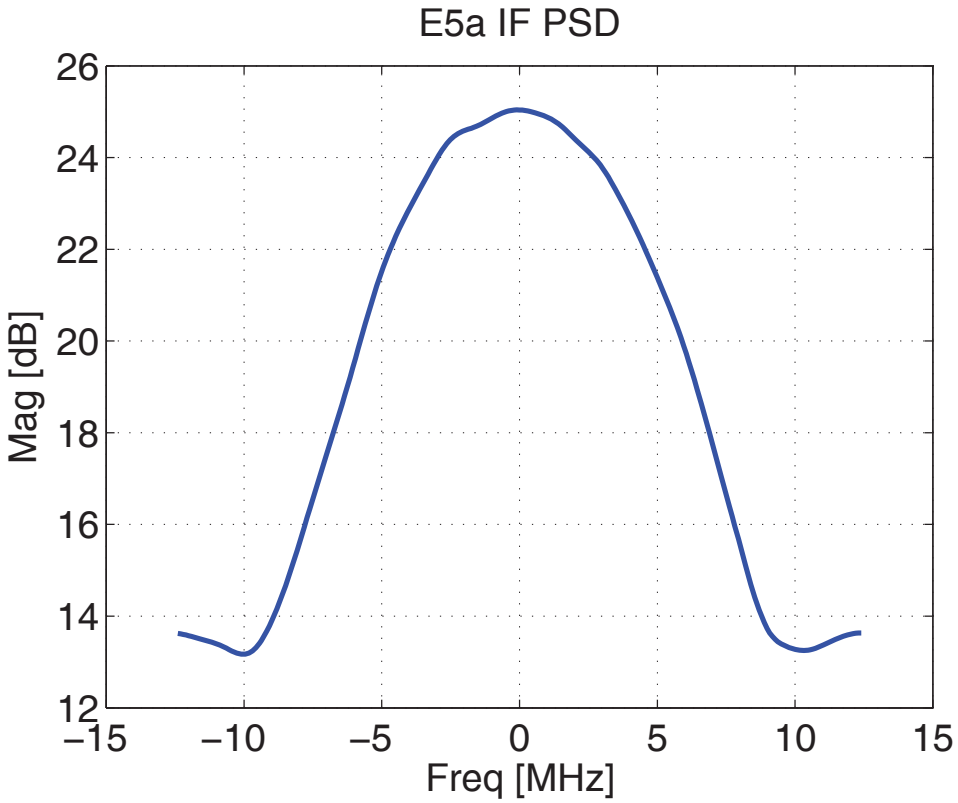


Figure 2.14: Noiseless Galileo E5a OS PSD - over the ICD specified reference bandwidth of 20.46 MHz.

2.3.5 Ionosphere scintillation simulation

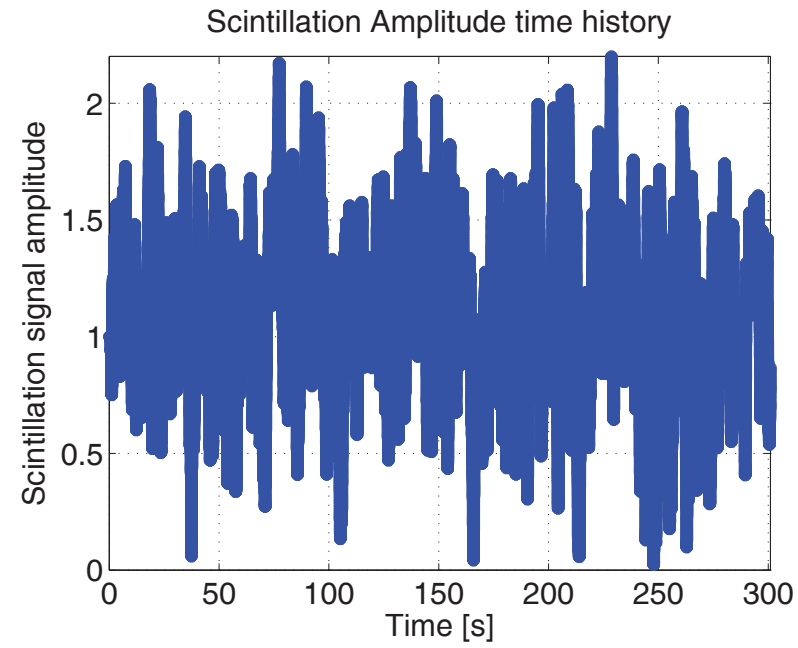
The ionosphere scintillation is simulated in addition to the Galileo OS signals, taking into account the impact over the amplitude and phase of the original Galileo signal. The ionosphere scintillation model described in [22] is used to simulate ionosphere scintillation where two inputs, the standard scintillation amplitude index S_4 and the channel decorrelation time τ_0, determine the amplitude and phase disturbance over the ultra high frequency Ultra High Frequency (UHF) L-band signals. These two parameter inputs indicate the severity and speed of the scintillation in terms of amplitude and phase fluctuations. A possible improvement to this model would be to control the speed of the fluctuations in the amplitude and phase time history separately. The model in [22] is used in this work and is deemed realistic as it draws on an extensive set of equatorial scintillation data to approximate the signal propagation channel, namely by estimating both the scintillation amplitude distribution as well as the spectrum of the complex scintillation signal. A relatively medium scintillation characterized by a large S_4 index value of 0.6 and a low τ_0 value of 0.8 seconds is simulated where the corresponding amplitude and phase of the scintillation signal are shown in Figure 2.15.

2.3.6 Ionosphere scintillation indices estimation

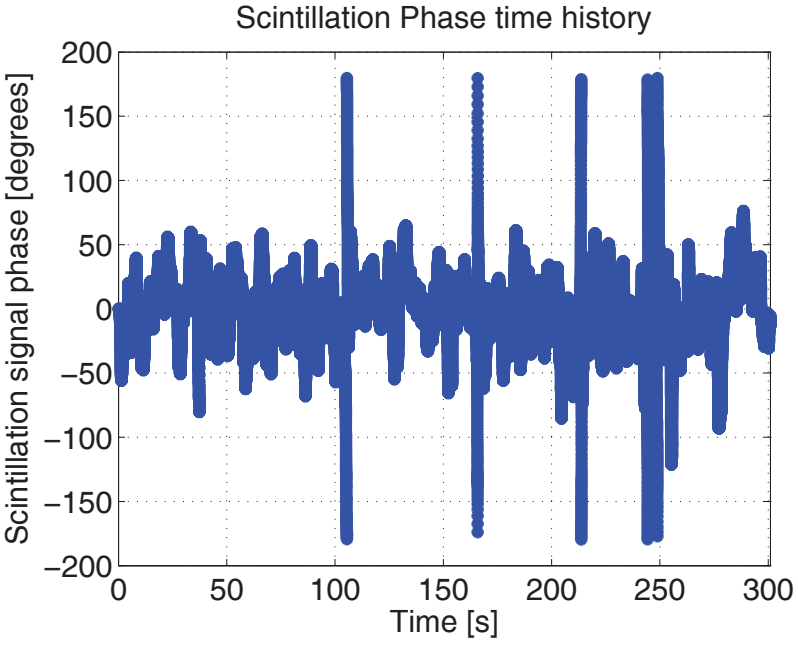
As was previously shown, the I and Q prompt correlator measurements are crucial in the estimation of the CNR of the received signal. Similarly the S_4 index estimation is based on the signal intensity (SI) which is equal to the difference of narrowband and wideband power estimates [26] as expressed in Section 2.3.3.

$$S_4 = \sqrt{\frac{< {S_I}^2 > - < S_I >^2}{< S_I >^2}} \tag{2.18}$$

where $< . >$ is the expected value. In fact, S_4 is a normalized standard deviation of signal intensity and is normally computed on 60 second intervals. Typically narrowband and wideband power values are computed every 20 ms using multiple I and Q correlator values following Equations 2.7 and 2.8, i.e. M=5 for Galileo E1 OS and 20 for Galileo E5a/b signals. This is due to the different accumulation intervals used on the two frequency bands respectively which is determined by their code periods, i.e. 4 ms for Galileo E1 OS and 1 ms for Galileo E5a/b. However, this does not prevent a fair



(a) Scintillation amplitude



(b) Scintillation phase

Figure 2.15: Simulated scintillation signal amplitude and phase - using an $S_4 = 0.6$ and a $\tau_0 = 0.8$ to be multiplied and added respectively to the raw Galileo simulated signal.

comparison between the two signals as it has been proved in [27] that using the normalization in the S_4 computation, removes any dependence of S_4 on the accumulation interval T_I. Finally, the need for detrending arises due to the dependence of the signal intensity measurements on the satellite receiver Dynamics, the receiver Oscillator phase error, the Ionosphere and Troposphere errors, collectively known as the (DOIT) terms. For that end, a 6th order Butterworth low pass filter is used with a cutoff frequency of 0.05 Hz instead of the traditional 0.1 Hz [27]. The detrended signal intensity is obtained by dividing the raw signal intensity by the low pass filter output adjusted by the filter group delay:

$$S_I = \frac{NBP - WBP}{(NBP - WBP)_{LPF}} \tag{2.19}$$

Moreover, the S_4 due to ambient noise has to be removed from the squared value of the raw S_4 index:

$$S_{4N_0} = \frac{100}{\hat{C}/N_0}\left(1 + \frac{500}{19\hat{C}/N_0}\right) \tag{2.20}$$

where $\hat{C}/N_0$ estimates are derived using either the PRM or the VSM expressions as desribed in the previous subsections.

On the other hand, the traditional phase scintillation index $\sigma_{\Delta\phi}$ is the standard deviation of the Accumulated Doppler Range (ADR) which is the accumulation of the estimated phase by the Phase Locked Loop (PLL) during the kth integration interval and the carrier discriminator estimated phase error $e_p[k]$:

$$\sigma_{\Delta\phi} = std(\sum_k 2\pi\hat{f}[k]T_I + e_p[k]) \tag{2.21}$$

However, before computing the standard deviation, the carrier phase measurements as the signal intensity measurements, should be detrended as well to ward off DOIT trends. The detrending is applied at post-processing through a 6th order Butterworth high pass filter with a cutoff frequency of 0.05 Hz [27]. Consequently, the standard deviation of the detrended carrier phase measurements is computed every 30 seconds.

2.4 Receiver blocks

GNSS receivers architecture has gone through significant change over the last few decades, especially with the advance of Field Programmable Gate Arrays (FPGA) technology and subsequently with the speed of software processing. In fact, convential GNSS receivers which were once built according to Application Specific Integrated Circuits (ASIC) and FPGA technology relying mostly on hard-wired platforms, are more often pushing towards a Software Defined Radio (SDR) based technology. This is quite helpful for signal processing engineers nowadays, as it releases the full potential of post-processing in order to analyze a range of acquisition/tracking/demodulation and position techniques and algorithms tailored for specific applications and signal structures [2].

Throughout this dissertation, GNSS SDR technology is considered where the software capability is moved as close as possible to the antenna. Nonetheless, the FE separates the software defined baseband signal processing blocks from the antenna where the Signal In Space (SIS) is captured. Indeed, the front-end is a hardware module which comes after the antenna, and includes any Low Noise Amplifier (LNA), mixers/filters designed to down-convert the RF signal into an IF signal, ADC and baseband conversion. Figure 2.16 presents the IF baseband signal processing scheme which can be fully implemented in software. It encompasses several key stages of a GNSS receiver, including acquisition, tracking, bit synchronization, data demodulation and PVT estimation.

A bundle of signals as a result of the transmission of several satellites are summed up at the input of the receiver. One such signal denoted $s[n]$ is the result of a single satellite contribution. The acquisition engine is responsible of delivering rough estimates of the incoming signal carrier frequency $\hat{f}_0$ and spreading/ranging code phase $\hat{\tau}_0$ relative to a single satellite. This can be achieved thanks to the CDMA nature of most of GNSS signals. The tracking engine further refines these estimates, and prepares for the bit synchronization and data demodulation stages in order to set up the parameters for solving the position/velocity/time of the user receiver. Figure 2.17 further shows the two tracking loops PLL and Delay Locked Loop (DLL) that work in parallel to estimate the carrier frequency/phase and ranging code delay.

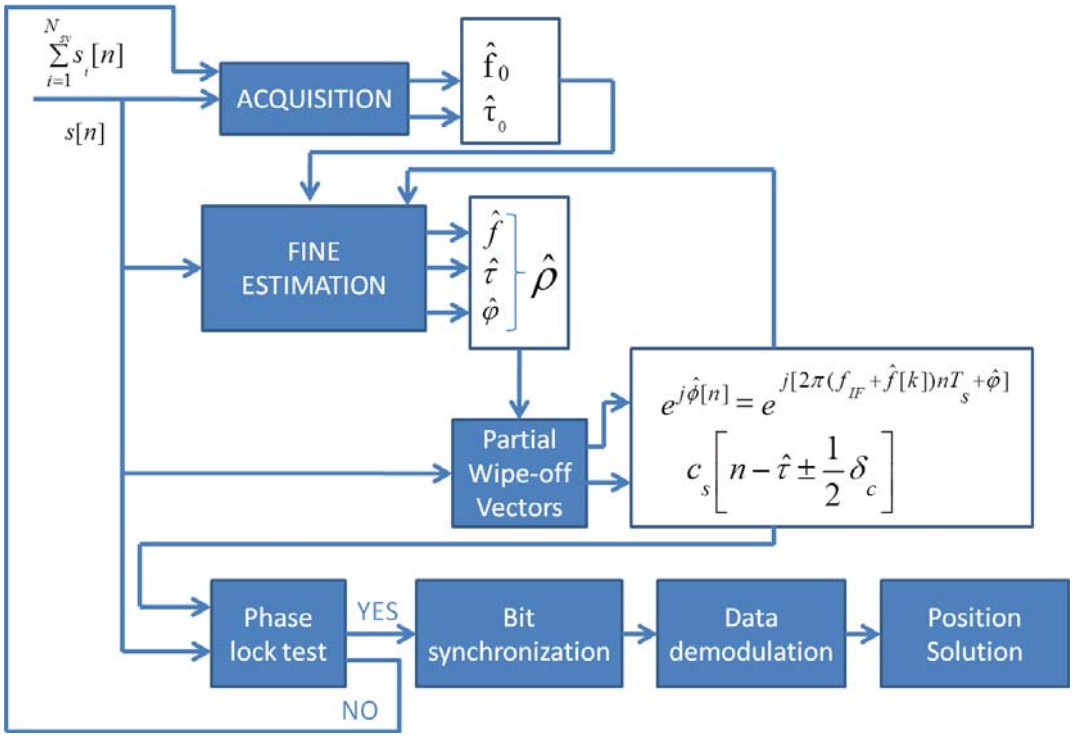


Figure 2.16: A general scheme depicting GNSS receiver blocks - starting from the input digital signal down to the position solution estimation.

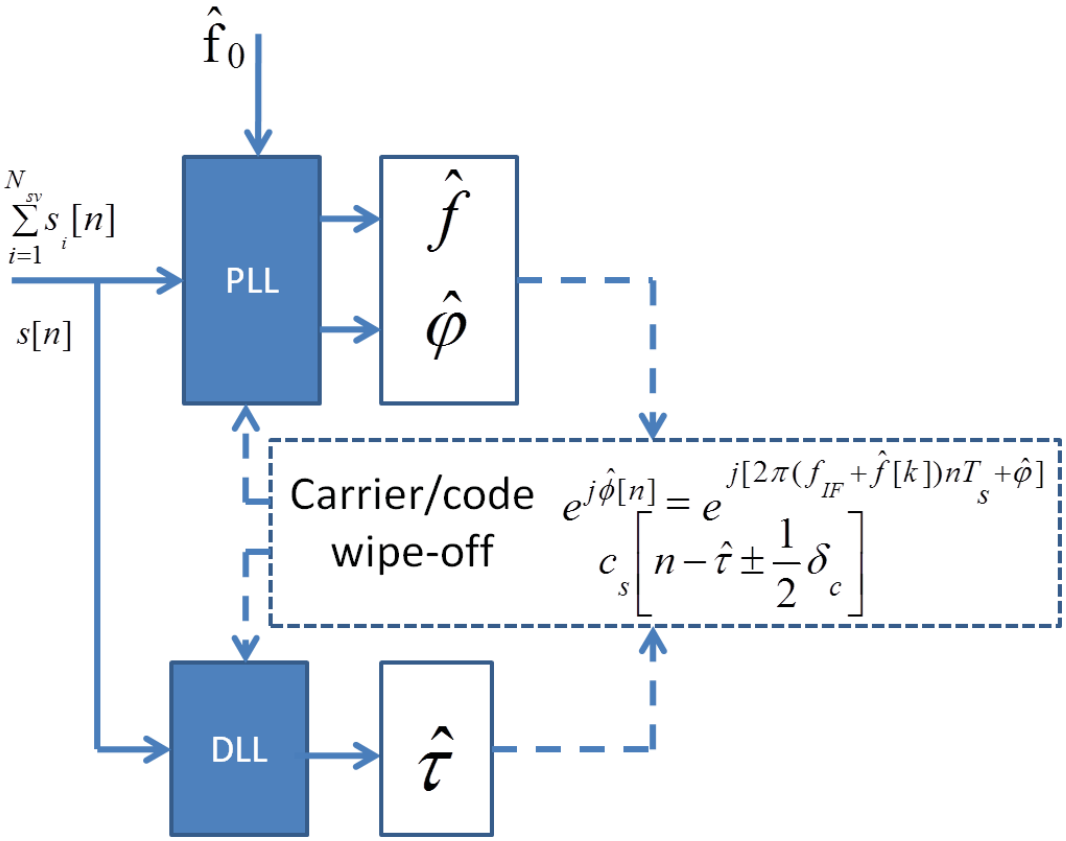


Figure 2.17: A general view of the fine estimation process - also known as tracking.

2.4.1 Acquisition

The acquisition block is the first block in a GNSS receiver signal processing chain as shown in Figure 2.16. It performs a rough estimation of the received signals parameters, mainly the spreading code delay and Doppler frequency. It then passes it on to a more refined estimation block, the tracking block in view of demodulation and positioning capabilities. As such, the acquisition engine is an important block to analyze as it can represent a bottleneck for the start of the rest of the blocks in the overall GNSS receiver. In fact, the Mean Acquisition Time (MAT) is a performance metric very often used in the literature to indicate together with the Time To First Fix (TTFF) the performance of a GNSS receiver. It is often relative to a single satellite signal rather than a minimum number of signals needed for positioning capibilities, and is studied as such in the following.

Almost all literature on the computation of the MAT is either focused on the standard serial search Threshold Crossing (TC) criterion [28], [29], a Maximum Threshold Crossing (MAX/TC) criterion [30] or a hybrid MAX/TC criterion [31], with no particular interest in the potential aidings received by a network of peers. Appendix A focuses on the serial TC criterion in light of the P2P context and an analysis of the MAT is performed under weak and strong signal conditions. Appendix B on the other hand, considers all search strategies and conducts a performance comparison between standard and P2P acquisition engines in light of the corresponding most appropriate search strategy. Moreover, a verification procedure is included together with a CNR aiding as a P2P network aiding information.

2.4.1.1 Galileo acquisition

Due to the Galileo signal structure which includes a secondary code, special care must be taken if signal acquisition is to be performed through coherent integration on several code periods. In fact, each code period is multiplied by a secondary code chip for both E1 and E5a Galileo OS signals except for the E1 data channel. The secondary code ensures good cross-correlation properties by extending the length of ranging/spreading codes. During coherent integration over several code periods, the presence of the secondary code modulation results in a destruction of the energy peak built in one code

period by other code periods. Therefore, it is essential to identify the delay or phase of the secondary code in addition to the primary code, in order to wipe it off when performing coherent integration over extended code periods.

It is worth noting that the coherent integration is of special use when dealing with weak signal conditions. In fact, it increases the post-correlation signal to noise ratio without a concurrent increase of the noise floor as is evident in case of noncoherent accumulations. Several techniques exist to solve for the secondary code phase, including the exhaustive approach by testing all possible secondary code phase positions after having solved for the primary code phase. Another technique considers the characteristic length approach [32] which identifies the minimum length of a sequence needed to solve for the chip position or phase. The minimum characteristic length is then evaluated for both Galileo E1 and E5 OS signals with 7 being the characteristic length of the 25 chips long secondary code present on the E1 pilot channel. On the other hand, the Galileo E5a/b pilot channels exhibit a different secondary code for each PRN and thus also yield a different characteristic length ranging from 9 to 18 but always less than 20 as shown in [32]. This approach is tailored to save memory and decrease power requirements as well as to deliver faster response by minimizing the number of operations needed to solve for the secondary code phase.

In conclusion, in order to perform coherent integration over several code periods (over 1 and 4 ms for Galileo E5a/b and E1 signals respectively), it is necessary to identify the secondary code phase. Consequently, a minimum integration time $T_I \geq 7 \cdot 4$ ms equivalent to 28 ms is to be used while performing acquisition of Galileo E1 pilot channel signals. Conversely, an integration time $T_I \geq 20 \cdot 1$ ms or 20 ms during pilot channel acquisition, guarantees identification of the secondary code phase for all PRNs on the Galileo E5a/b frequency band.

2.4.2 Tracking

As mentioned in the previous section, tracking is the fine counterpart of acquisition. It uses the output raw parameters estimated by the acquisition engine, and performs a fine estimation of the same parameters, i.e. the spreading code delay and Doppler

frequency. This is crucial to synchronize the incoming signal to the locally generated reference signal and perform navigation message demodulation and position/velocity/time computation in a reliable way.
The tracking stage paradox has always been center to a compromise between dynamic stress tolerance, and accuracy provided by the tracking loops. To accommodate for dynamic stress tolerance, the Pre Detection Integration Interval (PDI) denoted by T_I should be small, and a Frequency Locked Loop (FLL) discriminator should be used with a wide loop filter bandwidth. On the contrary, to increase accuracy and sensitivity, larger PDI should be used, possibly with a PLL discriminator and a narrow loop filter bandwidth. It has been shown by Ward [33] that the best solution for this dilemma, is to use FLL in the beginning and then FLL assisted PLL carrier tracking mode with a transition criteria involving a carrier-to-noise ratio meter and a phase lock detector.

A head to head comparison between FLLs and PLLs yields no particular winner but some characteristics are worth mentioning.

- An FLL discriminator makes use of current and previous phase values to estimate absolute phase difference between two epochs while a PLL discriminator makes use of current phase value only to estimate absolute phase in a single epoch. Consequently, in case the navigation message duration is 20 ms, the maximum PDI for an FLL is 10 ms compared to 20 ms for a PLL, in order not to straddle with data transition bits. However, the navigation message is absent in pilot channels, and therefore the maximum PDI can be increased irrespectively of the use of an FLL or a PLL without taking care of data bit transitions.

- The FLL is more robust compared to a PLL in terms of noise rejection. It has been shown in [34] that FLLs are able to track signals lower than 17 dB-Hz in static conditions, the limit at which PLLs loose lock. In dynamic conditions, FLLs track signals as low as 15 dB-Hz compared to 23 dB-Hz for PLLs in unaided conditions. With Doppler aiding, that figure goes down as low as 10 dB-Hz compared to 19 dB-Hz for PLLs, that is of course after accommodating the loop equivalent bandwidth. All these threshold values are derived in a statistical sense.

- The PLL is more accurate in the sense that it provides up to 2 orders of magnitude smaller delta range measurement errors than the FLL. Unlike the FLL, the PLL also provides a precise carrier phase solution.
- The FLL is less sensitive to colored noise (satellite clock, receiver clock phase jitter which is high with small B_N values, vibration, propagation delay due to ionosphere).
- The FLL's minimum achievable noise bandwidth B_N is smaller when dynamic stress effect is mitigated by Doppler aiding while it requires a noncoherent code tracking loop to maintain code synchronization.
- The recovery of navigation data bits inside the FLL is complicated if not impossible.

Chapter 3 presents the basic blocks inside a tracking control loop and analyzes the design of each of these blocks with respect to a pilot channel tracking scheme in view of processing Galileo OS E1 and E5a/b signals.

Chapter 3

GNSS pilot channel tracking

This chapter introduces the basic concepts to be considered for robust tracking of GNSS and specifically Galileo OS signals. The design accounts for weak signals with a low CNR where long integration time is a must for a successful tracking of these signals in terms of carrier frequency/phase as well as code delay. The main considerations are in terms of the phase detector (PD), the loop filter and the NCO, all of which are designed in digital domain. An analysis of the bandlimiting effects on the tracking performance in terms of thermal noise tracking jitter is presented where the main focus is on the CBOC and BPSK modulated Galileo E1 and E5a/b OS signals.

3.1 FLL/PLL/DLL structure

A standard frequency/phase/delay locked loop is a feedback control loop designed to track either carrier frequency and phase or code delay using a control signal which is an estimate of the tracking or estimation error. Its task is to maintain coherence between the input signal frequency/phase and the estimated output frequency/phase using phase comparison [24]. It consists of a pair of in-phase and quadra-phase correlators which process the product of the incoming signal and the output of the Reference Generator (RG), a PD or discriminator, a loop filter and a NCO. In this sense, a digital PLL (DPLL) is a proportional integral (PI) control loop which tracks the carrier phase of the input signal such that it maintains a zero steady state phase error. A digital frequency locked loop (DFLL) and a delay locked loop (DDLL) are the frequency and spreading code phase counterpart. GNSS carrier tracking loops typically use a second/third order

DPLL while the code tracking loop can be either a first order carrier aided DDLL or a second order unaided DDLL. A generic scheme of a GNSS FLL/PLL digital tracking loop is shown in Figure 3.1 and features the aforementioned three basic blocks:

- RG that is the NCO
- PD or discriminator
- Loop filter

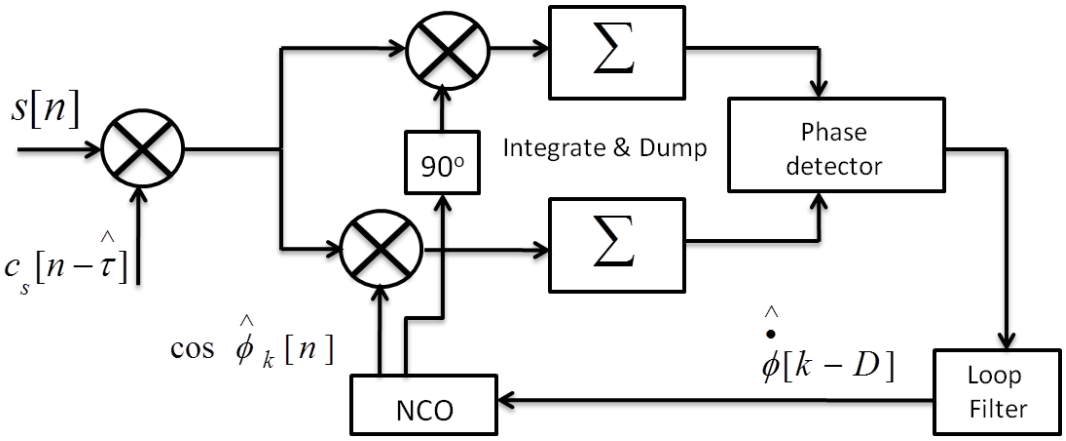


Figure 3.1: General scheme of a discrete phase tracking loop - is applicable for both FLL and PLL.

The incoming received signal $s[n]$ is first correlated with a locally generated PRN code $c_s[n-\hat{\tau}]$ that tries to mimic the behavior of the incoming PRN code (with an estimated code delay and code frequency which can vary depending on the Doppler frequency). These two parameters, code delay and code frequency are the by-products of the DLL. Subsequently, the integrators correlate the resulting signal from the product $s[n]c_s[n-\hat{\tau}]$ with the in-phase and quadra-phase of the carrier signal, and the result is summed over the total samples in an integration interval T_I. The resulting $I[k]$ and $Q[k]$ correlation values are then fed into the PD to estimate the phase error. The loop filter estimates the phase rate $\hat{\dot{\phi}}[k]$ at which the NCO has to function in order to generate a phase $\hat{\phi}_k[n]$ which minimizes the phase error between the received and local signals.

It should be noted that the same scheme shown in Figure 3.1 can represent a DLL

where the NCO outputs are replaced by early, prompt and late local PRN code replicas. The locally generated code $c_s[n]$ should also be replaced by the in-phase and quadrature phase carrier signal with the correct phase and frequency returned by the FLL/PLL. The DLL would then output an estimated code delay and code frequency. Figure 3.2 shows a global general scheme of a FLL/PLL/DLL mutually working to minimize the phase error and achieve lock.

It is more efficient and convenient to analyze the digital locked loop presented in 3.1 in its linear form in the frequency domain as can be seen in Figure 3.3. This linear version is applicable when the phase error between the reference and output signal is relatively low, that is when the loop is already locked.

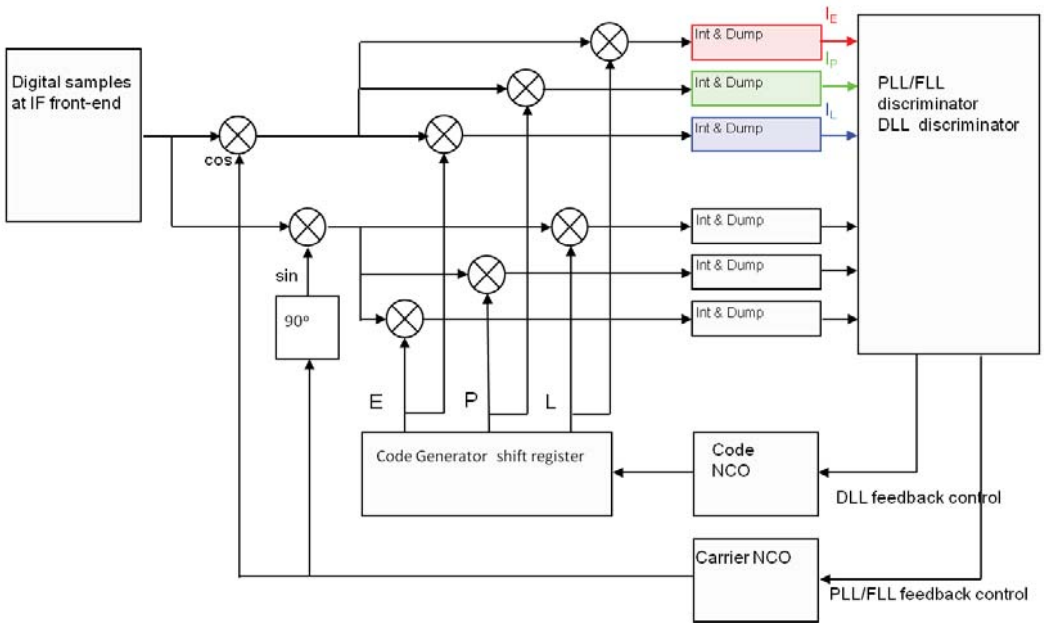


Figure 3.2: Global scheme of an FLL/PLL/DLL - Different discriminators and different NCOs are used.

The discriminator is expressed in terms of its z-transform $D(z)$ in ideal conditions, a multiplicative gain K_D and an additive noise term $N_W(z)$, both of which are function of the reference or incoming signal SNR [6]. The loop filter and NCO are expressed in the z-domain as $F(z)$ and $N(z)$. The next block z^{-D} expresses any delay in the loop

coming from the integration process to evaluate the phase error, where D indicates the number of delay units in terms of loop sample interval T_L. If the non-unity PD multiplicative gain K_D is compensated by some estimation mechanism like CNR estimation, and assuming a minimum delay of $D = 1$, the open loop transfer function can be expressed as:

$$G(z) = D(z)F(z)N(z) \tag{3.1}$$

which is used to determine the system transfer function $H(z)$ [35] and [24] as well as the system noise transfer function $H_N(z)$ by taking into account the noise effect in the expression of the estimated phase:

$$\begin{aligned} \hat{\phi}(z) &= (\phi(z) - \hat{\phi}(z))G(z) + N_W(z)G(z)/D(z) \\ \hat{\phi}(z)(1 + G(z)) &= \phi(z)G(z) + N_W(z)G(z)/D(z) \\ \hat{\phi}(z) &= H(z)\phi(z) + H_N(z)N_W(z) \end{aligned} \tag{3.2}$$

where $H(z)$ is the system transfer function describing in the z-domain the estimated output phase given the input phase:

$$H(z) = \frac{\hat{\phi}(z)}{\phi(z)} = \frac{G(z)}{1 + G(z)} \tag{3.3}$$

and $H_N(z)$ is the system noise transfer function yielding the contribution of the input additive noise term $N_W(z)$ to the estimated output phase:

$$H_N(z) = \frac{\hat{\phi}(z)}{N_W(z)} = \frac{G(z)/D(z)}{1 + G(z)} \tag{3.4}$$

noting that

$$H(z) = D(z)H_N(z) \tag{3.5}$$

Similarly, the system error function $H_e(z)$ and the system error noise transfer function $H_{eN}(z)$ can be derived substituting Equation 3.2 into the definition of the estimation error:

$$\begin{aligned} \phi(z) - \hat{\phi}(z) &= \phi(z) - H(z)\phi(z) - H_N(z)N_W(z) \\ \phi(z) - \hat{\phi}(z) &= \phi(z)(1 - H(z)) - H_N(z)N_W(z) \\ \phi(z) - \hat{\phi}(z) &= H_e(z)\phi(z) + H_{eN}(z)N_W(z) \end{aligned} \tag{3.6}$$

where $H_e(z)$ is the system error transfer function describing in the z-domain the estimated output phase error given the input phase:

$$H_e(z) = \frac{\phi(z) - \hat{\phi}(z)}{\phi(z)} = 1 - H(z) = \frac{1}{1 + G(z)} \tag{3.7}$$

and $H_{eN}(z)$ is the system error noise transfer function yielding the contribution of the input additive noise term $N_W(z)$ to the estimated output phase error:

$$H_{eN}(z) = \frac{\phi(z) - \hat{\phi}(z)}{N_W(z)} = -H_N(z) = -\frac{G(z)/D(z)}{1 + G(z)} \tag{3.8}$$

The linearity condition of the system is assumed to be satisfied, that is the system is assumed to be already in a locked condition. In the following sections, the transfer function of each individual block is defined and derived to satisfy a number of conditions.

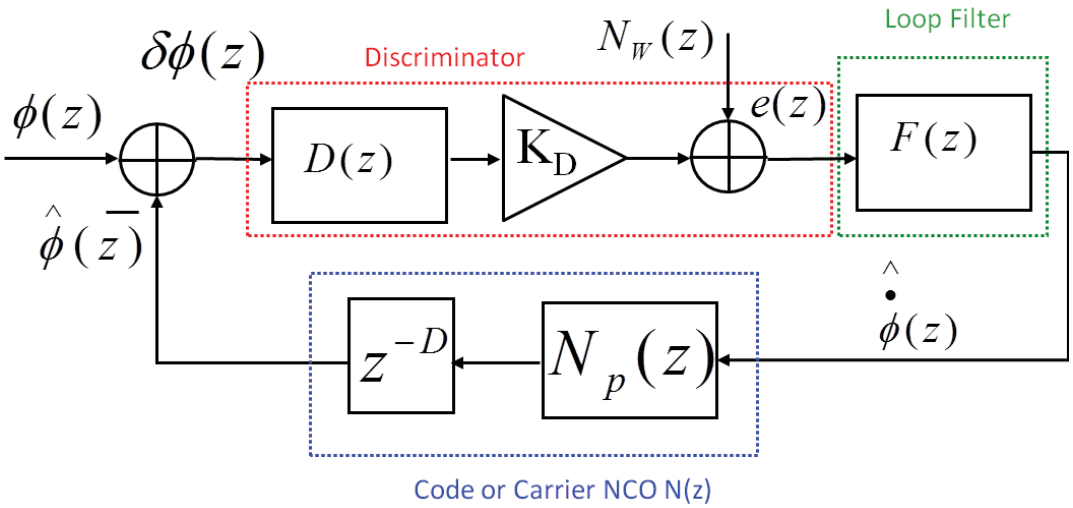


Figure 3.3: Linearized digital phase locked loop - in terms of transfer functions in z-domain and variable gain K_D and noise term $N_W(z)$ depending on input signal-to-noise ratio.

3.2 CU and DU loops

It is informative noting that the PLL/FLL scheme shown in Figure 3.1 works on two different sampling rates. The signal sampling rate $F_S = 1/T_S$ and the loop sampling rate $F_L = 1/T_L$. At this point, it is important to draw distinctions between generic

phase tracking loops and PRN spread-spectrum systems specific tracking loops. In part, it is this distinction that raises two kinds of update loops, Continuous Update (CU) and Discrete Update (DU) loops. CU loops work at the signal sampling frequency rate, that is the frequency the raw analog signal is sampled to obtain a digital signal. CU loops are highly theoretical for GNSS applications and require a high degree of complexity. DU loops on the other hand, work at the loop sampling rate F_L which is a relatively low multiple of the signal sampling rate (kHz instead of MHz).

The difference stems from the fact that PRN spread-spectrum systems hence GNSS tracking loops (unlike generic loops which have a direct way of measuring the estimation error) have to generate the estimation error by correlating the received signal with a locally generated one over an integration interval T_I. The correlation or integration and dump process is initiated every T_L seconds, which is the loop update interval or the loop sampling interval T_L.

The locally generated signal in turn, needs the parameters estimated by these tracking loops. However, the total phase estimated at a rate of $F_L < F_S$ by either a FLL/PLL/DLL PD is a sort of averaged phase at half the integration time $T_I/2$ instead of the signal sampling interval T_S. Therefore, GNSS tracking loops and more specifically the digital NCO has to estimate two variables, the phase and phase rate or frequency of the incoming signal. This code/carrier frequency is the rate of change in a specific integration interval that is instrumental in generating a phase for each sample within the correlation or integration interval (more on this in Section 3.3). The locally generated signal is thus constructed using the code/carrier phase and phase rate for each sample in the integration interval T_I. Correlating this local signal with the received one, results in turn in a new phase estimation error which is how the loop keeps working at subsequent epochs (separated by T_L). It should be noted that the loop update interval is usually set to be equal to the integration interval T_I because the reaction of the loop is a consequence of the phase estimated at half the integration interval, as well as to simplify the operation of the digital locked loop.

It is true that the distinction between CU and DU loops is evident when it comes to the operation mode of the loops, however, it has long been a tradition to study

DU loops based on CU loops in terms of analog domain transfer functions, using the s-transform. More specifically, although these DU loops are very frequently used in digital tracking loops, it is not necessarily true that they are designed so that the loop filter parameters take account of the inherent distinction between CU and DU loops. In fact, a CU approximation is performed where the normalized loop noise equivalent bandwidth $B_L = B_N T_I \rightarrow 0$, and the CU approximation yields loop roots which coincide with the original design requirements, i.e. the damping factor and the noise equivalent bandwidth B_N. However, as the normalized bandwidth $B_L = B_N T_I$ exceeds a certain value, the loop starts diverging from the required design yielding a higher noise equivalent bandwidth [36]. This is the case because of the shortcomings of the transformation used to jump from analog to digital domain. It is shown in [5] that using a bilinear transformation, a one-to-one frequency mapping from analog to digital is guaranteed as long as the product of the discrete frequency error by the loop sample interval is much less than 1, i.e. satisfiying $\delta\omega T_L << 1$ which explains the condition on the normalized loop bandwidth. A DU formulation of the problem needs to be elaborated as shown in [36]. The only caveat in [36] (also found in [37]) is that the loop filter is assumed to have a specific structure mainly composed of pure integrators $1/s$ in series such that the loop filter transfer function can only have poles at zero in continuous time and depending on the transformation method (bilinear, backward or forward Euler integration), poles at unity in discrete time. This can considerably limit the maximum integration time the loop can sustain for a specific loop bandwidth as shown in [5]. The best option would be to design the loop filter with no such constraints. One possible criteria to design the loop is based on minimizing the transient energy and the thermal noise tracking jitter as shown in [4, 5, 38].

3.3 NCO modeling

The incoming signal has a sampling rate of $F_S = 1/T_S$ and its discrete time domain is defined in terms of the variable $n = 1, 2, \cdots, T_I F_S$. The PLL/FLL scheme output has a loop sampling rate of $1/T_L$ which is usually equivalent to $1/T_I$. Its discrete time domain is defined in terms of the variable $k = 0, 1, 2, ...$ which corresponds to $n = T_L/2 + kT_L$. Therefore the NCO phase rate $\hat{\dot{\phi}}[k]$ (rad/s) is defined and constant during each epoch k while the total phase $\hat{\phi}_k[n]$ (rad) has to be generated for every sampling instant n during epoch k. The relationship between the phase rate and the generated total phase at each sampling interval T_S characterizes different NCO types, from the most often used, phase-continuity preserving, rate only NCO to the less famous phase and rate NCO which does not guarantee phase-continuity.

Linear analysis of the digital phase locked loop assuming a constant frequency during each integration interval, entails the use of a single phase value for every loop sampling interval or epoch k in order to estimate the phase error. This single phase value $\hat{\phi}_k$ is taken at mid-integration interval, as it is an intuitive mean or average of the generated phase in the integration interval. For very low loop update rates, which usually correspond to long integration intervals, this assumption does not hold true because the phase rate or frequency of the signal is more likely to change and not remain constant. This is why the average generated phase does not necessarily coincide with the phase at mid-integration interval. For the sake of linear analysis, it is assumed, unless otherwise noted, that a phase register holding the phase $\hat{\phi}_k$ at mid-integration interval is constantly updated at the loop sampling rate $1/T_L$.

3.3.1 Phase rate only NCO

The phase rate only NCO feedbacks the phase rate value $\hat{\dot{\phi}}[k+1]$ estimated by the loop filter at time $k+1$ to estimate the total average phase at the $k+1$th mid-interval $\hat{\phi}_{k+1}$ such that:

$$\hat{\phi}_{k+1} = \hat{\phi}_k^e + \frac{T_I}{2}\hat{\dot{\phi}}[k+1] \tag{3.9}$$

where $\hat{\phi}_k^e$ is the total phase at the end of the interval k given by:

$$\hat{\phi}_k^e = \hat{\phi}_k[T_I F_S] = \hat{\phi}_k + \frac{T_I}{2}\hat{\dot{\phi}}[k] \tag{3.10}$$

Initially, the average estimated phase at mid-integration interval for $k = 0$ is set using the frequency $\hat{f}_0$ estimated by the acquisition engine or alternatively the FLL as:

$$\hat{\phi}_0 = \hat{\dot{\phi}}_0 \frac{T_I}{2} = 2\pi \hat{f}_0 \frac{T_I}{2} \tag{3.11}$$

Moreover, there is an inherent delay, D units of T_L which is the loop update interval, because the estimated phase rate at the current epoch can only be applied in the next epoch. Incorporating this delay in the phase rates, the resulting average phase at epoch $k+1$ becomes:

$$\hat{\phi}_{k+1} = \hat{\phi}_k + \frac{T_I}{2}\left(\hat{\dot{\phi}}[k-D] + \hat{\dot{\phi}}[k+1-D]\right) \tag{3.12}$$

Taking the Z-transform of Equation 3.9, the corresponding rate only NCO transfer function can be obtained:

$$N_{RO}(z) = \frac{T_I(z^{1-D} + z^{-D})}{2(z-1)} = \frac{T_I(z+1)}{2z^D(z-1)} \tag{3.13}$$

On the other hand, the instantaneous phase $\hat{\phi}_{k+1}[n]$ generated for each sampling interval T_S during the $(k+1)$th epoch, can be expressed as:

$$\hat{\phi}_{k+1}[n] = \hat{\phi}_k^e + \hat{\dot{\phi}}[k-D]nT_S \tag{3.14}$$

Looking into the phase values generated using Equation 3.14 at the end and beginning of two successive epochs $\hat{\phi}_k[T_I] = \hat{\phi}_k^e$ and $\hat{\phi}_{k+1}[1] = \hat{\phi}_k^e + \hat{\dot{\phi}}[k-D]T_S$, it can be seen that the estimated phase value at the end of epoch k almost coincides with the estimated phase value at the beginning of epoch $k+1$ which explains why the rate only NCO maintains phase continuity.

3.3.2 Phase and rate NCO

The phase and rate NCO feedbacks the phase and phase rate value estimated by the loop filter at time k and $k+1$ respectively to estimate the total phase at mid-interval $k+1$ such that:

$$\hat{\phi}_{k+1} = \hat{\phi}_k^{re} + \frac{T_I}{2}\hat{\dot{\phi}}[k+1] \tag{3.15}$$

where

$$\hat{\phi}_k^{re} = \hat{\phi}_k + \frac{T_I}{2}\hat{\dot{\phi}}[k+1] \tag{3.16}$$

which means that the most recent estimated phase rate is being used to correct the phase estimated at the end of previous epoch k. This is why it can be expected that

the phase and rate NCO yields a better phase tracking performance. Considering the inherent delay D between generating the phase rate and the corresponding phase at the subsequent epoch, the resulting average phase at epoch $k+1$ is:

$$\hat{\phi}_{k+1} = \hat{\phi}_k + T_I \hat{\dot{\phi}}[k+1-D] \tag{3.17}$$

and the corresponding NCO transfer function can be expressed as:

$$N_{PR}(z) = \frac{T_I z^{1-D}}{z-1} \tag{3.18}$$

On the other hand, the instantaneous phase $\hat{\phi}_{k+1}[n]$ generated for each sampling interval T_S at the $k+1$th epoch, can be expressed as:

$$\hat{\phi}_{k+1}[n] = \hat{\phi}_k^{re} + \hat{\dot{\phi}}[k+1-D] n T_S \tag{3.19}$$

Comparing Equations 3.19 and 3.14 it becomes evident why the phase and rate feedback NCO should outperform the rate only feedback NCO. Moreover, it can be shown that phase continuity is not preserved by the phase and rate NCO using Equation 3.19 to compute the phase at the beginning and end of two successive epochs. For example, assuming that the computation delay $D = 1$, it can be shown that:

$$\hat{\phi}_{k+1}[1] \approx \hat{\phi}_k + \frac{T_I}{2} \hat{\dot{\phi}}[k] \tag{3.20}$$

and

$$\hat{\phi}_k[T_I] \approx \hat{\phi}_{k-1} + \frac{3T_I}{2} \hat{\dot{\phi}}[k-1] = \hat{\phi}_k + \frac{T_I}{2} \hat{\dot{\phi}}[k-1] \tag{3.21}$$

It can be easily seen that these two phase values do not guarantee phase continuity unless the phase rate remains constant over successive epochs which means that the loop is already in steady-state.

3.3.3 Analog version of phase and rate NCO

The analog version of phase and rate NCO does not use a phase register to accumulate the phases at mid-intervals. Instead, it uses a phase rate or frequency register $\hat{f}$ which gets updated every T_L. Taking the aforementioned inherent delay D due to phase rate correction computation, the estimated frequency or phase rate during each epoch k is expressed as:

$$\hat{f}[k] = \hat{f}_0 + \frac{\Delta \hat{\dot{\phi}}[k-D]}{2\pi} \tag{3.22}$$

where $\Delta\hat{\dot{\phi}}$ is the phase rate correction estimated by the loop filter at each epoch k as shown in Figure 3.4. Moreover, a partial phase less than 2π is usually generated based on the remaining partial phase from the previous interval:

$$\hat{\varphi}[k] = \mathrm{mod}(2\pi\hat{f}[k]T_I + \hat{\varphi}[k-1], 2\pi) \tag{3.23}$$

where mod is the modulo operator.

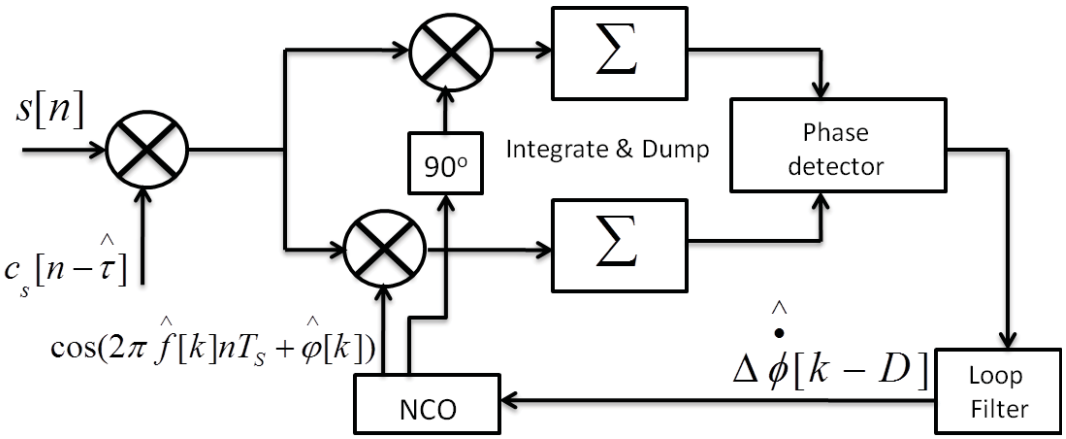


Figure 3.4: General scheme of a carrier tracking loop using an analog version NCO - The NCO updates a frequency register instead of a phase register.

The impact of this frequency and partial phase registers update in terms of estimated average total phase $\hat{\phi}_k$ at mid-intervals can be assessed by noticing that it is sufficient that a partial phase (modulo 2π) be taken into account given that:

- cosine and sine functions are periodic with period 2π
- the frequency or phase rate is already set from the loop filter output as in Equation 3.22

The total phase at mid-integration interval can be expressed as:

$$\hat{\phi}_k = 2\pi\hat{f}[k]\frac{T_I}{2} + \hat{\varphi}[k] \tag{3.24}$$

and substituting Equation 3.22 and 3.23 into 3.24, the total phase at mid integration interval can be derived at epoch k and $k-1$ assuming a delay $D = 1$:

$$\begin{aligned}
\hat{\phi}_k &= 2\pi\hat{f}_0\frac{T_I}{2} + \Delta\hat{\dot{\phi}}[k-1]\frac{T_I}{2} + \mathrm{mod}(2\pi\hat{f}_0T_I + \Delta\hat{\dot{\phi}}[k-1]T_I + \hat{\varphi}[k-1], 2\pi) \\
\hat{\phi}_{k-1} &= 2\pi\hat{f}_0\frac{T_I}{2} + \Delta\hat{\dot{\phi}}[k-2]\frac{T_I}{2} + \mathrm{mod}(2\pi\hat{f}_0T_I + \Delta\hat{\dot{\phi}}[k-2]T_I + \hat{\varphi}[k-2], 2\pi)
\end{aligned}$$

It can be noticed that the partial phase at epoch $k-1$ can be expressed in terms of $\hat{\varphi}[k-2]$:

$$\hat{\varphi}[k-1] = \text{mod}(2\pi \hat{f}_0 T_I + \Delta\hat{\dot{\phi}}[k-2]T_I + \hat{\varphi}[k-2], 2\pi) \tag{3.25}$$

Moreover, since the modulo operation is distributive and $\text{mod}(\text{mod}(.)) = \text{mod}(.)$, it can be derived that:

$$\hat{\phi}_k = \hat{\phi}_{k-1} + \text{mod}(2\pi \hat{f}_0 T_I + \Delta\hat{\dot{\phi}}[k-1]T_I, 2\pi) - \Delta\hat{\dot{\phi}}[k-2]\frac{T_I}{2} + \Delta\hat{\dot{\phi}}[k-1]\frac{T_I}{2} \tag{3.26}$$

Assuming that the phase rate correction is almost equal between two consecutive epochs and taking the modulo 2π operation, it can be concluded that:

$$\text{mod}(\hat{\phi}_k, 2\pi) \approx \text{mod}(\hat{\phi}_{k-1}, 2\pi) + \text{mod}(\hat{\dot{\phi}}[k]T_I, 2\pi) \tag{3.27}$$

after noticing that $\hat{\dot{\phi}}[k] = 2\pi\hat{f}[k]$. Considering the modulo 2π phase values, it can be seen that this is a similar version to the expression characterizing phase and rate feedback NCOs as in Equation 3.17. In conclusion, the analog version or frequency NCO yields a similar version of the phase and rate NCO transfer function:

$$N_F(z) = N_{PR}(z) = \frac{T_I z^{1-D}}{z-1} \tag{3.28}$$

On the other hand, the instantaneous phase $\hat{\phi}_{k+1}[n]$ generated for each sampling interval T_S at the $(k+1)$th epoch, can be expressed as:

$$\hat{\phi}_{k+1}[n] = 2\pi\hat{f}[k]nT_S + \hat{\varphi}[k] \tag{3.29}$$

In fact, it is thanks to this instantaneous phase generation that the correlation operation is possible in order to maintain carrier/code synchronization.

3.4 Discriminators

As explained in the beginning of this chapter and in Section 3.2, and as shown in Figure 3.1, in-phase I and quadradure phase Q correlators are used by the discriminator to generate the estimation error between the incoming phase signal and the locally generated one. Looking into the expressions of I and Q correlators, various functions $f_\phi(I, Q)$ called discriminators or PD have been designed in literature [37] to accurately estimate the phase error, but this has proven to be an arduous task in low CNR conditions.

The ideal behavior of the PD output is a straight line of unity slope which passes through the origin at zero phase error $\delta\phi = 0$ or frequency error $\delta\omega = 0$ in case of FLL PD. However, the PD inputs are noisy signals and the discriminator functions usually yield a nonlinear function mainly a sine function of the phase error. For very small phase errors, $\sin(x) \approx x$ is a good approximation, however, depending on the discriminator, a high level of noise power can distort the ideal line, decreasing the slope at the origin and introducing nonlinearities at the edges of the PD output. Other discriminators do not depend on the noise but are characterized by a narrow error range and call for a normalization factor which depends on the received signal power.

3.4.1 DFLL discriminators

A typical frequency discriminator estimates the frequency difference or error between the incoming signal and the local reference signal. Various frequency discriminators are present in the literature, and are divided into two main categories: coherent discriminators as Cross Product (CP) and four-quadrant arctangent (ATAN2), and noncoherent discriminators as Decision Directed Cross Product (DDCP) and Differential Two-quadrant Arctangent (DATAN) which are considered in [6] and described in [37]. Noncoherent discriminators are essential when dealing with data channels because they are insensitive to half cycle changes while coherent discriminators are sensitive to these changes and are mainly used with pilot channels.
A typical frequency discriminator can be derived by considering a typical phase discriminator and applying the definition of angular frequency with respect to phase. In fact, the two-quadrant Arctangent (ATAN) noncoherent DPLL phase discriminator is

the maximum likelihood estimator [39]. It estimates a noncoherent phase error at the kth epoch mid-interval $\delta\phi_k$ in the following manner:

$$e_p[k] = \arctan(Q[k]/I[k]) \approx [\delta\phi_k]_\pi \tag{3.30}$$

where the notation $[x]_\pi$ means that the variable x is limited over the range $[-\pi/2, \pi/2]$ by adding or subtracting π. Assuming that the estimated phase error at various epochs can be approximated by a continuous signal, the estimated angular frequency error can be expressed as:

$$e_f[k] = \frac{e_p[k] - e_p[k-1]}{T_I} \approx \frac{de_p[k]}{dt} \tag{3.31}$$

This is the intuition behind differentiating the phase with respect to time in order to obtain a frequency discriminator. In the following, an abuse of notation is tolerated for simplicity, replacing $e_p[k]$ by e_k. So let us first differentiate $\tan e_k = Q[k]/I[k]$ with respect to the phase:

$$\begin{aligned} \frac{d}{de_k}\tan(e_k) &= \frac{d}{de_k}\frac{\sin e_k}{\cos e_k} \\ &= \frac{\cos^2 e_k + \sin^2 e_k}{\cos^2 e_k} = \frac{1}{\cos^2 e_k} \end{aligned} \tag{3.32}$$

Due to the definition of the phase with respect to the correlation values I[k] and Q[k] at epoch k, $\tan e_k = Q[k]/I[k]$ and looking at it in a geometrical sense, e_k is the angle of a triangle rectangle or a right triangle with legs I[k] and Q[k] and the hypothenuse expressed as $\sqrt{I[k]^2 + Q[k]^2}$ applying the Pythagorean theorem. Therefore, it can be written that:

$$\cos e_k = \frac{I[k]}{\sqrt{I[k]^2 + Q[k]^2}} \tag{3.33}$$

Now, together with this identity, the Chain rule can be applied to obtain the derivative of e_k with respect to time:

$$\begin{aligned}
\frac{d}{dt}\tan e_k &= \frac{d}{dt}\frac{Q[k]}{I[k]} \\
\frac{d}{de_k}\tan e_k\frac{de_k}{dt} &= \frac{Q[k]^{'}I[k]-Q[k]I[k]^{'}}{I[k]^2} \\
\frac{1}{\cos^2 e_k}\frac{de_k}{dt} &= \frac{Q[k]^{'}I[k]-Q[k]I[k]^{'}}{I[k]^2} \\
\frac{de_k}{dt} &= \frac{Q[k]^{'}I[k]-Q[k]I[k]^{'}}{I^2[k]+Q^2[k]} \\
e_f[k] &= \frac{1}{T_I}\frac{Q[k]I[k-1]-I[k]Q[k-1]}{I^2[k]+Q^2[k]}
\end{aligned} \tag{3.34}$$

where $Q[k]^{'}=\frac{Q[k]-Q[k-1]}{T_I}$ and $I[k]^{'}=\frac{I[k]-I[k-1]}{T_I}$. Taking the mean value of such a discriminator function for a given frequency and phase error $\delta\omega[k]$ and $\delta\phi[k]$ respectively, and assuming that the code phase estimate is accurate enough to yield a unity spreading code correlation value, it can be shown that:

$$\begin{aligned}
E[e_f[k]T_I] &\approx \frac{Pd[k]d[k-1]sinc^2(\frac{\delta\omega[k]T_I}{2})\sin(\delta\phi[k]-\delta\phi[k-1])}{Psinc^2(\frac{\delta\omega[k]T_I}{2})} \\
E[e_f[k]T_I] &\approx d[k]d[k-1]\sin(\delta\phi[k]-\delta\phi[k-1])
\end{aligned} \tag{3.35}$$

Assuming that the data bits can be demodulated correctly or considering pilot channel tracking, the first two terms can be treated as unity and the frequency error estimate $e_f[k]$ at epoch k depends on the coherent phase error and thus is a coherent DFLL discriminator. Its corresponding PD transfer function is written as:

$$D(z)=\frac{e(z)}{\delta\phi(z)}=\frac{1-z^{-1}}{T_I} \tag{3.36}$$

The discriminator defined by Equation 3.34 will be named Basic Cross Product (BCP) discriminator in this dissertation. It will be seen in the following that it is one form of several discriminators often mentioned in literature, that is the CP DFLL discriminator but with a different normalization. In fact, the CP DFLL discriminator uses a normalization by the signal power at the antenna input which is more accurate than signal power estimation using correlation. Another digital discriminator, called the DATAN

in this paper, can be derived if the optimum phase discriminator is applied directly to Equation 3.31, thus yielding [23]:

$$e_f[k] = \frac{1}{T_I}\left[\arctan(Q[k], I[k]) - \arctan(Q[k-1], I[k-1])\right] \tag{3.37}$$

This frequency discriminator yields an output frequency error, which is an approximation of the difference between two consecutive noncoherent phase error values, averaged by the integration interval:

$$e_f[k] \approx \frac{[\delta\phi_k]_\pi - [\delta\phi_{k-1}]_\pi}{T_I} = \delta\omega[k] \tag{3.38}$$

It is worth mentioning that the BCP, CP and DDCP discriminators yield a sinusoidal approximation of the true frequency error $\delta\omega$, while the DATAN and ATAN2 discriminators yield an exact linear output as shown in [37]. In fact, the DATAN discriminator has a similar response compared to the ATAN2 discriminator because both of these PD yield a linear approximation of $\delta\omega$ as demonstrated in Equation 3.38. The distinction of two important regions in the analysis of digital PD, the stability range and the linear range, stems from the previous considerations regarding the coherence/noncoherence and the linear/nonlinear output nature.

- The stability range is the range of input frequency or phase error which results in an output frequency or phase error estimate of the same sign as that of the input, that is a discriminator output that reacts in the same direction as the input error.
- The linear range is the range of input frequency or phase error which results in a proportial output frequency or phase error estimate.

Therefore, noncoherent frequency discriminators are defined on a stability range of $\pm\pi/(2T_I)$ radians/s while coherent frequency discriminators offer a range of $\pm\pi/T_I$ radians/s. The resulting maximum pull-in frequency corresponding to the PD is thus $\pm\pi/T_I$ rad/s or $\pm 1/(2T_I)$ Hz. Note that this is completely determined by T_I the integration interval and not the update interval. Indeed, increasing the update interval does not bring any improvement whatsoever in increasing the pull-in frequency range through the discriminator. Since we are dealing with pilot channel tracking, we will focus on coherent discriminators in the following. Let us list the main differences between the ATAN2 and CP coherent discriminators:

- ATAN2 relies heavily on the input CNR while CP is independent of it.
- CP should be normalized by the received signal power while ATAN2 is already normalized by itself.
- In ideal conditions, ATAN2 has a wider single-sided frequency pull-in range $1/(2T_I)$ compared to $1/(8T_I)$ for the CP discriminator.

To better understand these differences, it is appropriate to present some results of these two discriminators in different conditions of CNR and integration time.

3.4.1.1 DFLL ATAN2 discriminator and optimum integration time

The coherent DFLL ATAN2 discriminator input-output characteristic can be defined in terms of the in-phase and quadrature phase correlator values of two successive epochs $I[k], I[k-1], Q[k], Q[k-1]$ as in [37]:

$$e_f[k] = \frac{1}{T_I}\text{atan2}(\text{Dot}, \text{Cross}) \tag{3.39}$$

where

$$\begin{aligned} \text{Dot} &= I[k]I[k-1] + Q[k]Q[k-1] \\ \text{Cross} &= Q[k]I[k-1] - I[k]Q[k-1] \end{aligned} \tag{3.40}$$

Figures 3.5 and 3.6 show the output frequency error in terms of input frequency error for a specific integration time $T_I = 4$ ms and $T_I = 40$ ms and various CNR. It is worth noting some points here:

- The stability range is equal to the inverse of the integration time and is usually greater than the linear range.
- The linear range shrinks as the CNR decreases and for relatively low CNR almost becomes absent.
- The PD slope at the origin decreases as the CNR decreases and for relatively low CNR almost yields a flat output.
- For low CNR value, increasing the integration time may either increase the slope of the straight line output or yield a linear range, even if it is relatively narrow.

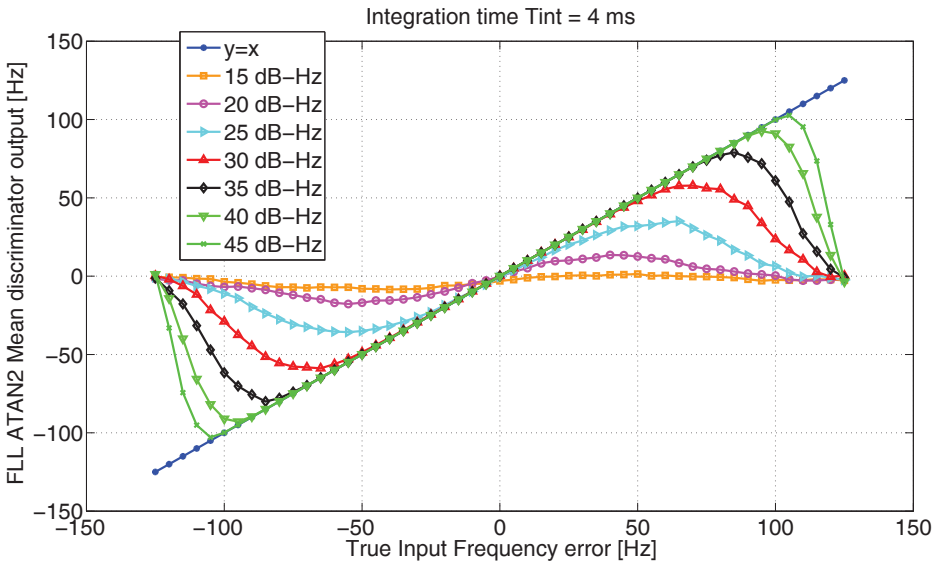


Figure 3.5: Mean ATAN2 DFLL discriminator output with $T_I = 4$ ms - Mean ATAN2 discriminator output for varying CNR and $T_I = 4$ ms demonstrating the absence of linear range for the 15 dB-Hz curve.

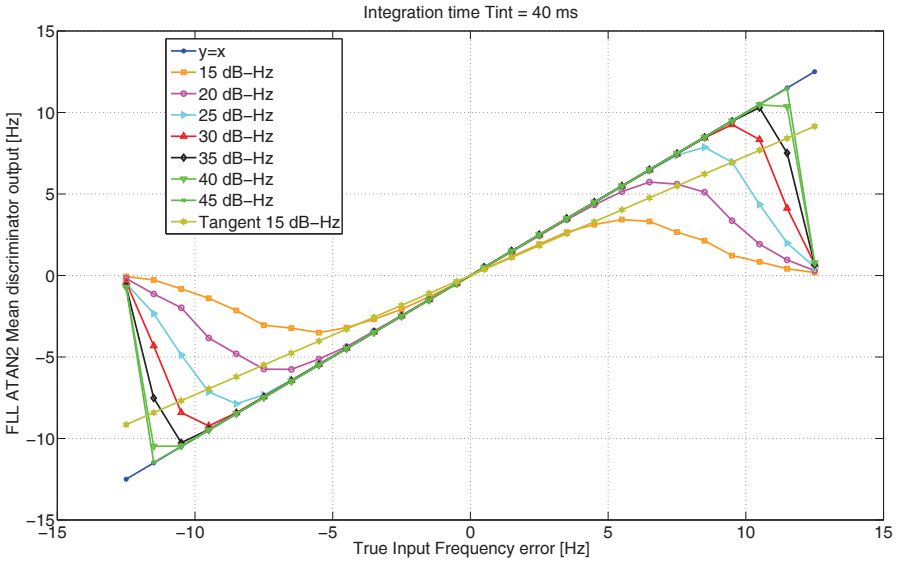


Figure 3.6: Mean ATAN2 discriminator output with $T_I = 40$ ms - Mean ATAN2 discriminator output for varying CNR showing a relatively small linear frequency range for $C/N_0 = 15$ dB-Hz.

Based on these observations, an intuitive solution for choosing the optimum integration time that guarantees a wide linear range with a unity slope, would be to choose the minimum allowable integration time and to rotate the PD output curves relative to low CNR in Figure 3.5. These curves have slopes at the origin lower than one but introducing a PD gain would compensate the low CNR effect, such that the slopes eventually become equal to one. However, for relatively low integration time and CNR values, the output line can be flat (for 15 dB-Hz in Figure 3.5) and so there would be no linear range and no gain can do any compensation. Increasing the integration time for the same CNR yields a linear range even if the frequency error range is as small as ± 5 Hz as seen in Figure 3.6 where a straight line is plotted which passes through the origin with identical slope relative to the output due to a 15 dB-Hz. Thus, the choice of the integration time should be a trade-off between assuring a linear response from the PD and specifying a relatively wide frequency range for the PD. The aim of this section is to determine that optimum integration time.

Phase detector gain The idea developed in [6] is an intuitive solution that compensates for the PD gain. However, it does not address the issue of optimum choice of the integration time that corresponds to a certain input frequency error tolerance or a frequency pull-in range. It introduces a PD gain K_D, i.e. the slope, together with a noise term $n_W[k]$ as shown in Figure 3.3 to compensate for the low PD performance in low CNR signals. Indeed, this approach should be applied very carefully because it assumes that the tracking takes place in the linear range. Whenever the linear range is exceeded, the tracking loop incurs cycle slips and could go out of lock as a result. For this reason, given that the linear range or the range of frequency errors that the FLL is able to track reliably is determined by the integration time, it is of utmost importance to compensate with the PD gain only when an appropriate integration time T_I is determined as a function of the maximum expected input frequency error or input frequency pull-in range. The maximum input frequency error can be an estimate coming from both the expected dynamics of the receiver as well as the quality of the earlier acquisition process.

First, some basic variables are introduced that are instrumental towards the endeavor

of deriving the optimum integration time. As expected, the PD gain K_D depends on the CNR, and indeed, a signal quality metric SNR_C is defined in [6] such that:

$$\text{SNR}_C = \frac{2PT_I}{N_0} = 2T_I C/N_0 \tag{3.41}$$

The PD gain is then defined for a specific discriminator as a function of SNR_C by using the probability distribution function $p(\theta)$ of an apparent phase θ induced by thermal noise due to $n_W[k]$. Indeed, the gain for an ATAN2 PD is given by [6]:

$$K_D = 1 - 4\pi \int_0^{\pi} p(\theta)p(\theta - \pi)d\theta \tag{3.42}$$

where the probability distribution of the phase error is given by [40]:

$$p(\theta) = \frac{e^{\frac{-\text{SNR}_C}{2}}}{2\pi}\left(1 + \sqrt{\frac{\pi\text{SNR}_C}{2}}\cos(\theta)e^{\frac{1}{2}\text{SNR}_C\cos^2(\theta)}\left(1 + \text{erf}\left\{\frac{\sqrt{\text{SNR}_C}\cos(\theta)}{\sqrt{2}}\right\}\right)\right) \tag{3.43}$$

with $-\pi \leq \theta \leq \pi$ and $\text{erf}(x)$ the Gaussian error function.

Optimum DFLL ATAN2 integration time To determine the optimum integration time, two conditions have to be satisfied:

- A linear range should be present, i.e. the mean frequency estimate has to be equal or very close to the input frequency error possibly after applying the PD gain K_D.
- The integration time should yield the widest linear range.

Assuming a frequency error $\delta\omega$ which is constant over the integration time, the mean output or frequency estimate of the ATAN2 discriminator is defined in [6] :

$$\mu_e = \frac{1}{T_I}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} f(\delta\omega T_I + \theta_2 - \theta_1)p(\theta_1)p(\theta_2)d\theta_1 d\theta_2 \tag{3.44}$$

where $f(x)$ is the noise-free or ideal frequency discriminator input-output characteristic which is a straight line passing through the origin with unity slope:

$$f(x) = x\left(u(x + \pi) - u(x - \pi)\right) \tag{3.45}$$

where $u(x)$ is the Heaviside step function. Satisfying the first condition for optimum integration time yields:

$$\left|\frac{\mu_e}{K_D} - \delta\omega\right| < \epsilon \tag{3.46}$$

where ϵ is the error tolerance in frequency estimation, arbitrarily set to 10% of the ideal value $\delta\omega$ rad/s. Satisfying the second condition entails solving for the maximum frequency range $\delta\omega$ that satisfies condition 3.46, in other words:

$$\delta\omega_{max} = \max_{\left|\frac{\mu_e}{K_D} - \delta\omega\right| < \epsilon} \delta\omega \tag{3.47}$$

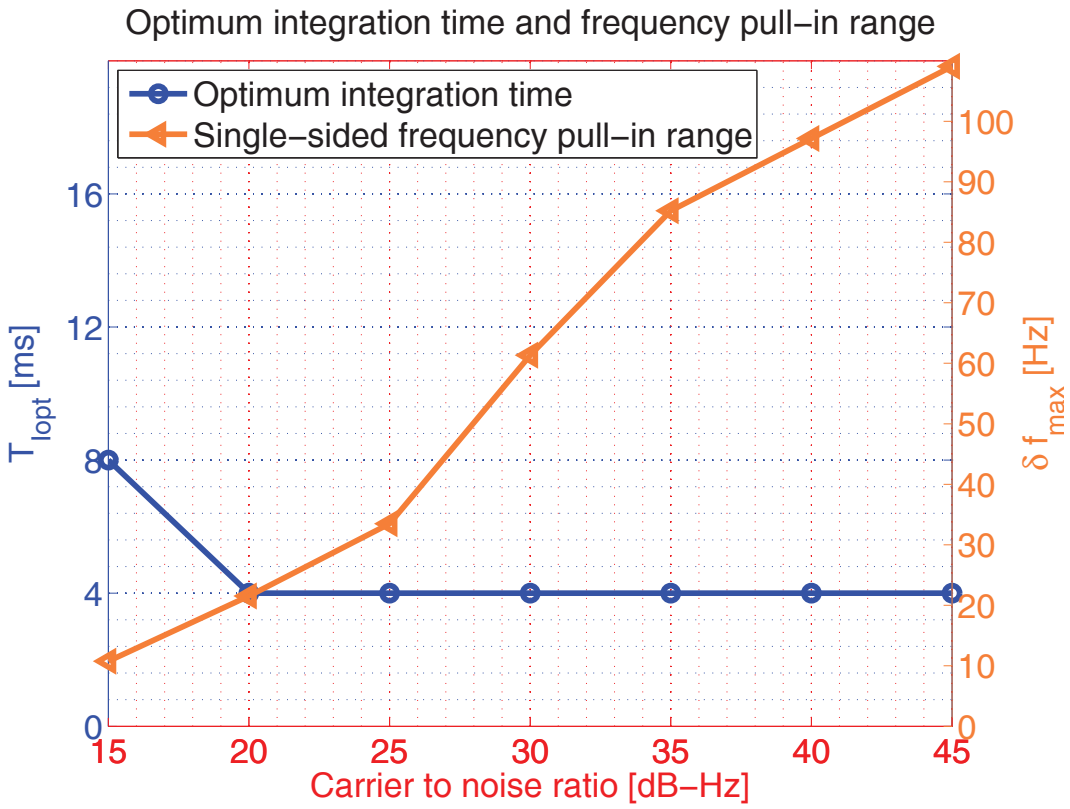


Figure 3.7: Whole region optimum integration time and corresponding frequency pull-in range - assuming an error tolerance of 10% for every point in the linear range.

Solving for $\delta\omega_{max}$ will result in the optimum integration time T_I. The optimal integration time and the corresponding frequency pull-in range as shown in Figures 3.7 and 3.8 are obtained here by numerical integration due to the complexity of the problem. The former figure is the result of checking condition 3.46 is satisfied for the whole range, while the latter figure is the result of checking just one frequency error point at the boundary of the linear range. It is worth noting that due to the very low CNR at 15

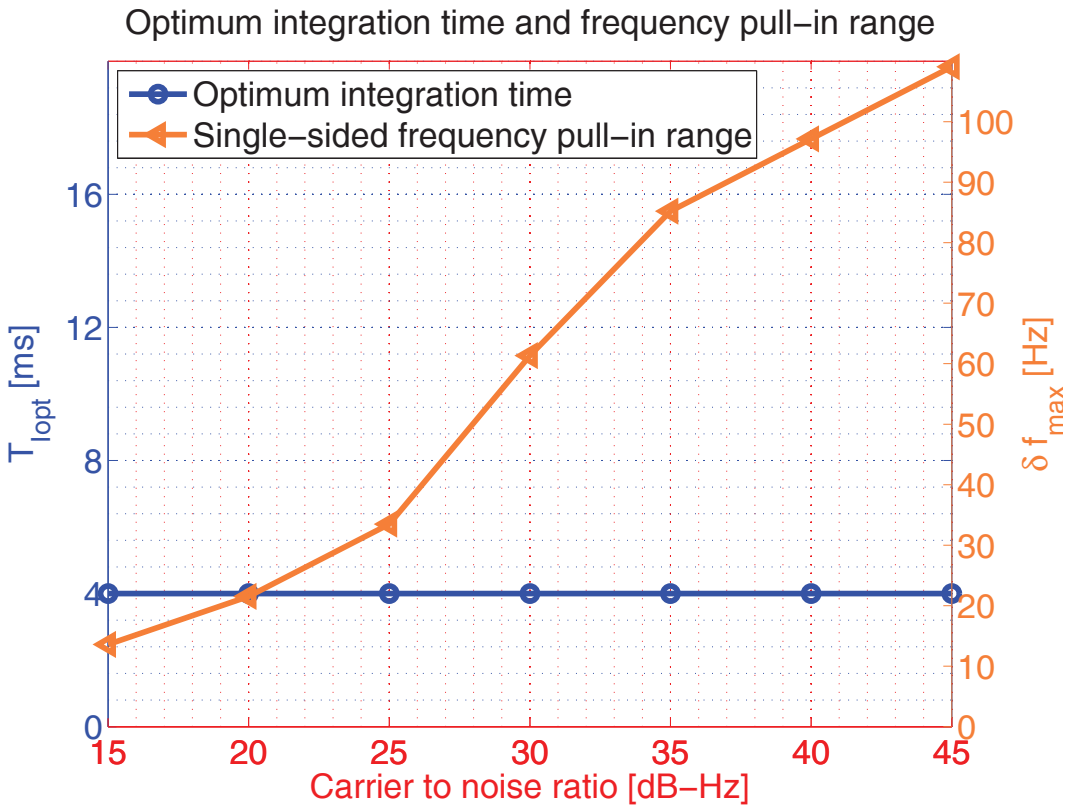


Figure 3.8: Singular point optimum integration time and corresponding frequency pull-in range - assuming an error tolerance of 10% and considering the boundary point in the linear range.

and 20 dB-Hz the optimum integration time and the corresponding frequency pull-in range derived in both of these figures are inconsistent between various simulation runs and hence unreliable. The reason is that the noise floor is much higher than the signal, such that the discriminator response is more determined by noise than the signal, and an error tolerance of 10% yields inconsistent results instead of declaring unreliable or false tracking. Indeed, condition 3.46 is no longer satisfied when the error tolerance is lowered to 1% and the result is infinite integration time together with a zero frequency pull-in range.

Another way to look at this problem, is to set the sub-optimal integration time, given the maximum frequency error that is expected to sustain. In other words,

$$\epsilon_{min} = \min_{\delta\omega=\delta\omega_{max}} \left| \frac{\mu_e}{K_D} - \delta\omega \right| \tag{3.48}$$

Checking that $\epsilon_{min} \leq \epsilon$ is satisfied, will yield a solution of an acceptable integration time for the expected input frequency error $\delta\omega_{max}$.

3.4.1.2 DFLL CP discriminator

As previously mentioned in Section 3.4.1, the CP DFLL discriminator is another form of the BCP discriminator defined by Equation 3.34 which is derived on the basis of the optimum DPLL discriminator. The only difference is the choice of input signal power normalization instead of correlation power normalization. The coherent DFLL CP discriminator input-output characteristic can thus be defined in terms of the in-phase and quadrature phase of two successive correlator values:

$$e_f[k] = \frac{1}{\sqrt{P}T_I} \left(Q[k]I[k-1] - I[k]Q[k-1] \right) \tag{3.49}$$

and where unlike the ATAN2, the coherent CP discriminator is then normalized by the input signal power $\sqrt{P}$. If the receiver estimates the incoming signal power, the CP discriminator is characterized by a CNR independent input-output response with a corresponding PD gain K_D equal to one as seen in Figure 3.9. However, if the BCP PD is used with a correlation power normalization as in Equation 3.34, then the input-output characteristic depends on the CNR as shown in Figure 3.10 and the corresponding PD gain decreases as in the case of the ATAN2 discriminator.

Moreover, the CP PD has a relatively narrow input single-sided frequency pull-in range $\pi/(8T_I)$ as compared to the ATAN2 discriminator. The integration time can be set to the minimum value in this case whatever low is the input CNR. This is in contrast with the $2\pi/T_I$ tolerance for the ATAN2 discriminator, where the integration time should be increased with very low CNR as discussed previously in setting the optimum integration time. In conclusion, it has been shown in [6] that in weak signal conditions, a more robust response in terms of lower phase tracking jitter is expected to be gained from CP discriminators. Consequently, the coherent CP discriminator can be switched to operate instead of the ATAN2 discriminator, whenever the CNR indicator goes below a threshold value of the SNR_C.

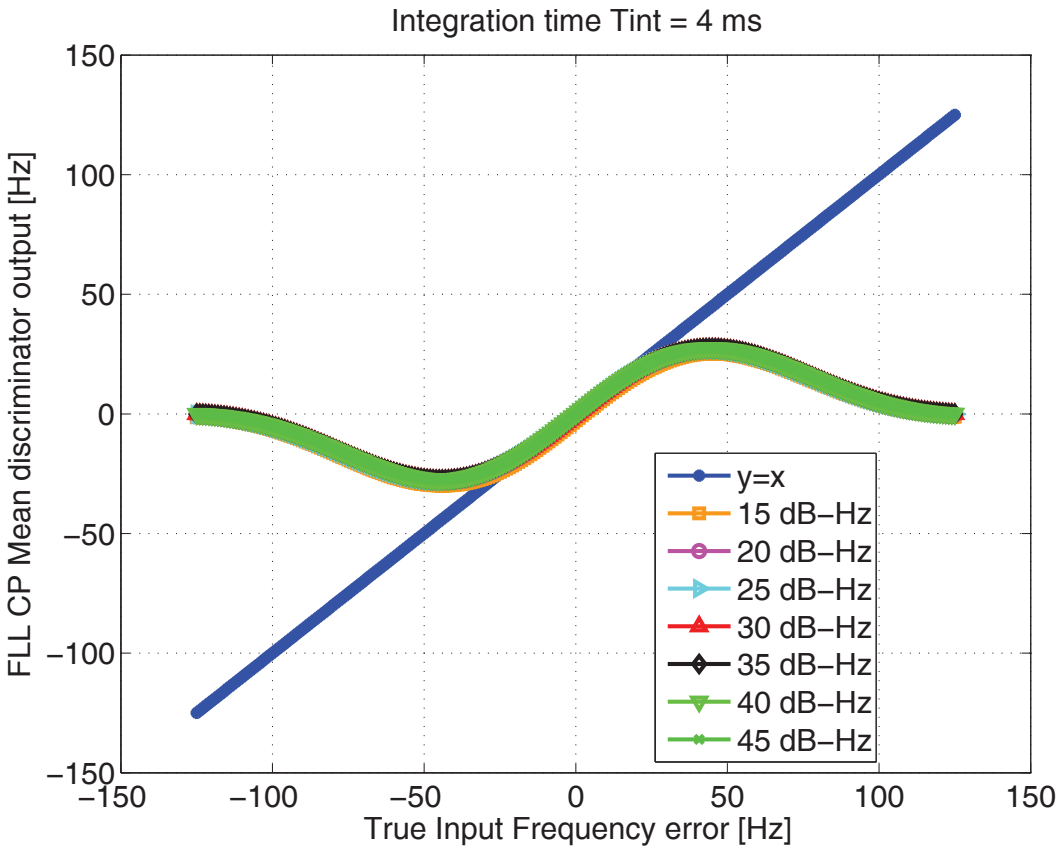


Figure 3.9: Mean CP discriminator output with $T_I = 4$ ms - Mean CP discriminator output does not depend on CNR nor integration time.

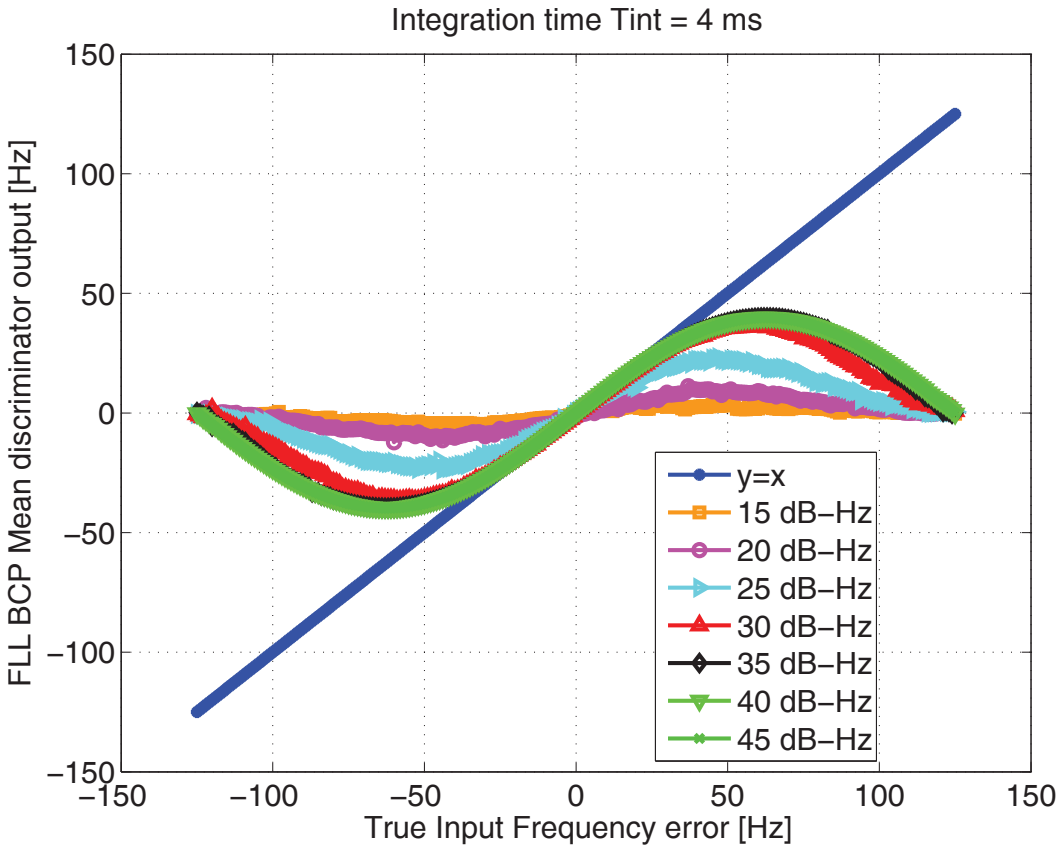


Figure 3.10: Mean BCP discriminator output with $T_I = 4$ ms - depends on CNR as it is normalized by the correlation power.

3.4.2 DPLL discriminators

Similarly to frequency discriminators, there are different phase discriminators described in literature, which are divided into two main categories: coherent and noncoherent. Each discriminator has a series of characteristics favoring it over another:

- Data channels require the phase discriminator to be insensitive to the inherent phase jumps due to data bits. In this case, Costas or noncoherent PD like the Dot Product (DP) [9], Decision Directed Dot Product (DDDP), Ratio RAT and ATAN discriminators, the latter of which is the optimal phase discriminator are favored over others [37]. Moreover, the maximum coherent integration time is limited to the duration of the data bit, i.e. 1 or 4 ms for Galileo signals and 20 ms for GPS signals.

- Pilot channels are not modulated by data bits, and so can extend their integration time to much higher values without losing lock depending on the quality and dynamics of the received signal. In this case, the Coherent (COH) and the extended arctan or ATAN2 discriminators [37] are a better option, because of their wider linear range.

In other words, noncoherent phase discriminators have a stability range limited to $\pm\pi/2$ radians while coherent phase discriminators stability range is defined over the whole circle $\pm\pi$ radians. These phase discriminators attempt to directly estimate the input phase error $\delta\phi$. It is worth mentioning that only the ATAN and ATAN2 discriminators yield a linear output estimate of the input phase error while it is expected a sinusoidal input phase estimate from the remaining discriminators. Therefore, given the approximation $e_p[k] \approx \delta\phi_k$ for both cases in low phase error conditions, the ATAN and ATAN2 discriminators are characterized by a unity transfer function for high CNR:

$$D(z) = \frac{e(z)}{\delta\phi(z)} = 1 \tag{3.50}$$

Furthermore, the COH discriminator needs to be normalized by the average power of the signal in order to provide insensitivity to high and low CNR. Since we are dealing with pilot channels, we will focus on coherent phase discriminators in the following. Similarly to the differences between the coherent DFLL discriminators, let us list the main differences between the coherent DPLL discriminators:

- ATAN2 input-output response relies heavily on the input CNR while CP is independent of it.
- COH should be normalized by the received signal power while ATAN2 is already normalized by itself.
- In ideal conditions, ATAN2 has a wider linear tracking phase range $\pm\pi$ compared to $\pm\pi/4$ for the COH discriminator.

To better understand these differences, it is appropriate to present some results of these two discriminators in different conditions of CNR and integration time.

3.4.2.1 DPLL ATAN2 discriminator and optimum integration time

The DPLL ATAN2 discriminator is a coherent variant of the noncoherent ATAN optimum DPLL discriminator. Its input-output characteristic can be defined in terms of the in-phase and quadrature phase correlator values at a given epoch k [37]:

$$e_p[k] = \text{atan2}(Q[k], I[k]) \tag{3.51}$$

Figures 3.11 and 3.12 show the output phase error in terms of input phase error for a specific integration time $T_I = 4$ ms and $T_I = 40$ ms and various CNR. It is worth noting some points here:

- The stability range is equal to $\pm\pi$ and is usually wider than the linear range.
- The linear range shrinks as the CNR decreases and for relatively low CNR almost becomes absent.
- The PD slope at the origin decreases as the CNR decreases and for relatively low CNR almost yields a flat output.
- For low CNR value, increasing the integration time may increase the slope of the straight line output or yield a linear range, even if it is relatively narrow.

Indeed, it can be seen in Figure 3.12 that a relatively large integration time of 40 ms applied on an input signal characterized by a CNR of 15 dB-Hz, yields a limited linear range of around $\pm\pi/2$ using an ATAN2 DPLL PD. Any phase variation at the input signal of amplitude larger than $\pm\pi/2$ would result in cycle slips and may also lead to loss

of lock depending on the duration and size of the phase variations. In some applications where these phase variations are quite frequent and large (ionospheric scintillation), it is desired that the linear range covers the whole possible range, that is $\pm\pi$.

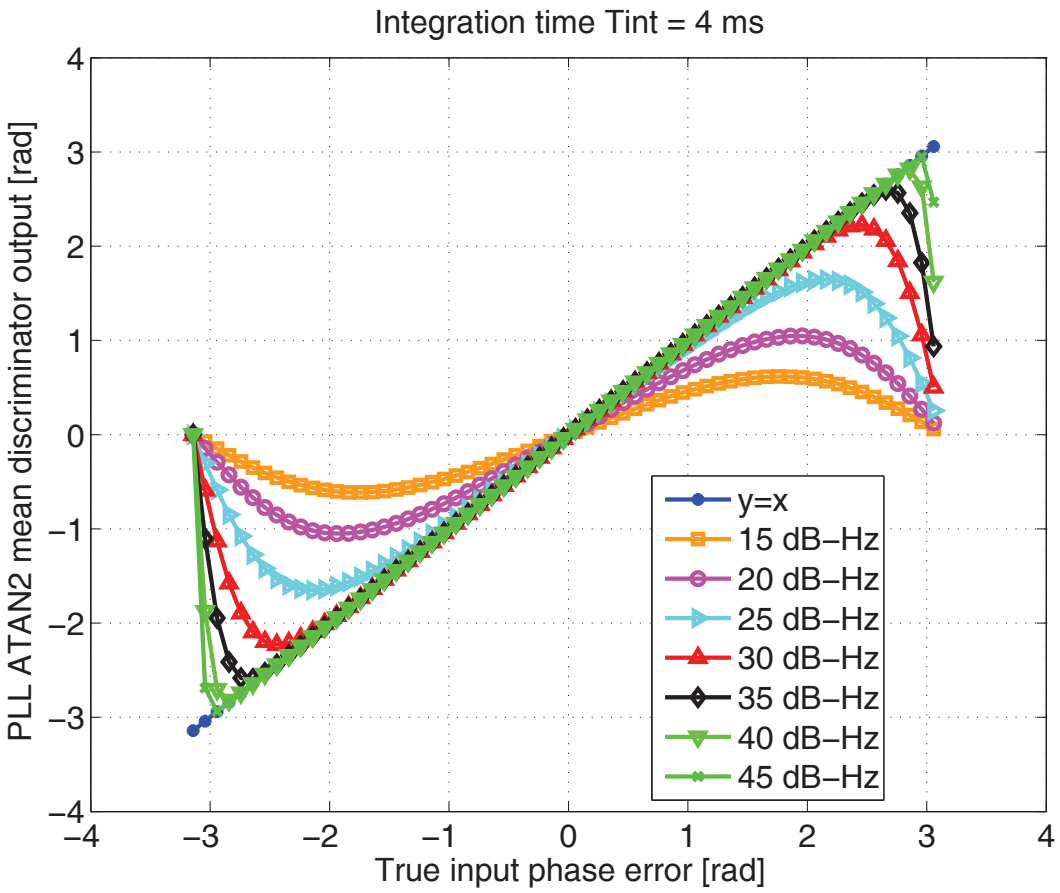


Figure 3.11: Mean DPLL ATAN2 discriminator output with $T_I = 4$ ms - and varying CNR demonstrating the absence of linear range for the 15 dB-Hz curve.

Optimum integration time It can be seen that increasing the integration time improves the output of the DPLL PD, however the integration time inside the DPLL is limited by the desired noise equivalent bandwidth which is an estimate of the amount of noise at the input of the DPLL. In the classical DPLL of a GNSS receiver, the loop update rate is the inverse of the chosen integration interval T_I. Therefore, as the integration time increases, the loop sampling rate or update rate decreases. To satisfy the

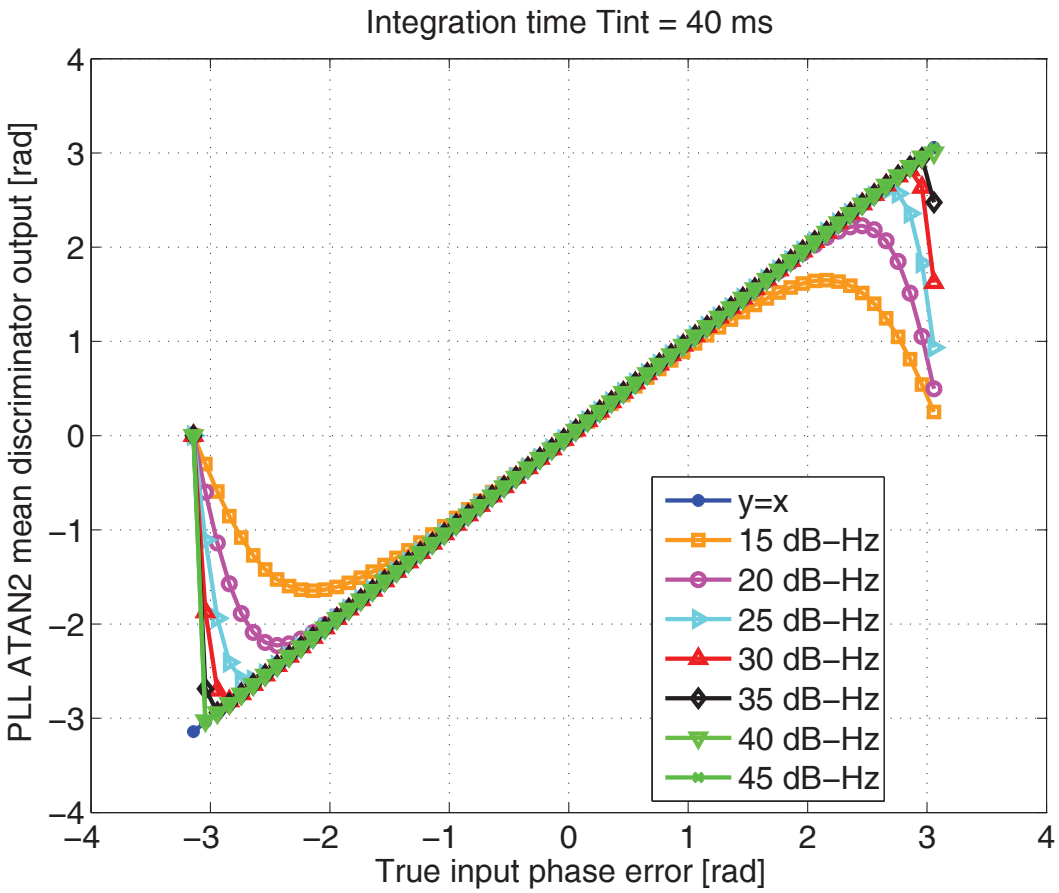


Figure 3.12: Mean DPLL ATAN2 discriminator output with $T_I = 40$ ms - restoring a certain linear range for the 15 dB-Hz curve.

Nyquist sampling frequency theorem, the noise equivalent bandwidth B_N can not go beyond $1/(2T_I)$ or conversely $T_I < 1/(2B_N)$. Knowing that a specific noise threshold exists, the design value of the noise equivalent bandwidth to be passed to the DPLL can not go below 5 Hz for example and consequently the integration time can not go beyond 100 ms in this case. However, the linear range should be as wide as $\pm\pi$ radians to account for any large phase variations that is possible to encounter at the input signal. Looking into Figure 3.13, it can be seen that the minimum CNR yielding this wide $\pm\pi$ linear range is 35 dB-Hz, or even down to 20 dB-Hz if a slightly higher tolerance of the linear range is considered $\pm 0.86\pi$. Taking the noise equivalent bandwidth threshold into account, this study can help identify the minimum CNR of a signal that is possible to track without incurring cycle slips and using the DPLL discriminator mainly the ATAN2 PD which has a slope that is highly dependent on both the CNR and integration time.

Alternatively, the loop update or sampling rate F_L can be increased with respect to the inverse of the integration interval T_I such that:

$$F_L = m/T_I \tag{3.52}$$

and thus the noise equivalent bandwidth can not go beyond $F_L/2 = m/(2T_I)$. Considering the same noise equivalent bandwidth threshold of 5 Hz, it results that $T_I < m/10$, or that T_I can be extended m times more with respect to its value using the classical approach, that is setting the loop update rate as the inverse of the integration interval.

3.4.2.2 DPLL COH discriminator

The COH DPLL discriminator is defined as the imaginary part of the correlator output or the quadra-phase component [37]-[6]. It does not depend on the CNR or the integration time as long as it is normalized by the input signal power P:

$$e_p[k] = \frac{Q[k]}{\sqrt{P}} \tag{3.53}$$

This is shown in Figure 3.14 where the COH PD output is simulated for different values of the input signal CNR and yields the same output. Moreover, the COH PD has a relatively narrow linear range $\pm\pi/3$ as compared to that of the ATAN2 discriminator $\pm\pi$. The integration time can be set to the minimum value in this case whatever low is the

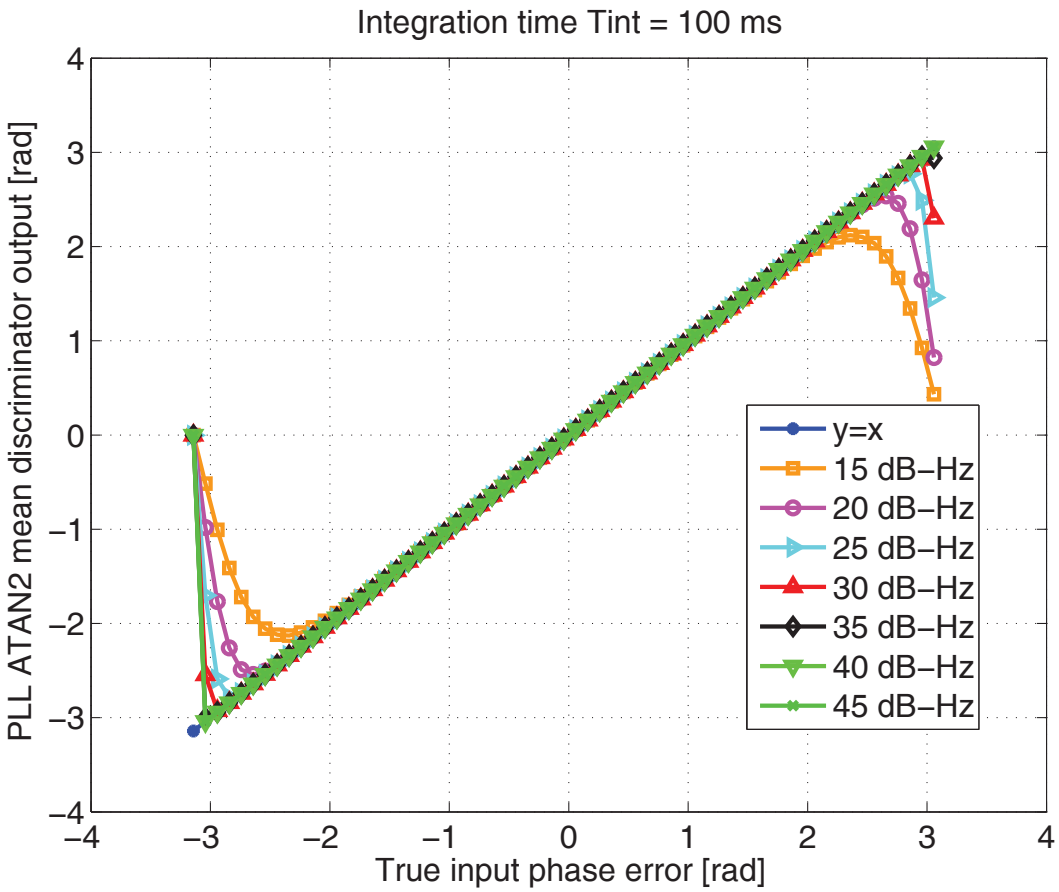


Figure 3.13: Mean DPLL ATAN2 discriminator output with $T_I = 100$ ms - showing the minimum CNR of 35 dB-Hz that yields a linear range over the whole $\pm\pi$ radians.

input CNR. This is in contrast with the ATAN2 discriminator, where the integration time should be increased with very low CNR as to restore or widen the linear range. In conclusion, it has been shown in [6] that in moderately weak to strong signal conditions, the choice of PD is not straightforward and depends on the application at hand, trying to reach a compromise between transient and steady state response. On the other hand, in weak signal conditions, a more robust response in terms of lower phase tracking jitter is expected to be gained from the COH discriminator. Consequently, in general conditions, the best approach seems to switch on the COH discriminator to operate instead of the ATAN2 discriminator, whenever the CNR indicator goes below a threshold value of the SNR_C for the same integration time. It is also worth noting that increasing the integration time results in choosing the ATAN2 PD instead, just because it has a wider linear range. In this sense, it is also worth noting the importance of choosing an optimum integration time. However, the choice of the discriminator between ATAN2 and COH depends on the specific application, and for ionospheric scintillation cases, it is better to employ an ATAN2 PD with an optimum integration time in order to absorb large phase variations.

3.4.3 DFLL-assisted DPLL discriminators

There are also DFLL-assisted DPLL discriminators, as the Unambiguous Frequency Aided (UFA) discriminator [4]. The UFA is a DPLL PD with the difference of having an accumulator which counts phase jumps from one epoch to another, something that a normal DPLL PD would otherwise not be able to detect. This is done by correcting the usual PD output in the following manner [4]:

$$e_u[k] = e_p[k] - I_\pi(e_p[k] - e_u[k-1]) \tag{3.54}$$

where $I_\pi(x) = x - [x]_{2\pi}$ and the operator $[.]_{2\pi}$ keeps the x value inside the interval $[-\pi, \pi]$ when dealing with pilot channels.

3.4.4 DDLL discriminators

The DDLL discriminator's function is to estimate the code phase τ of the received signal's spreading code. To understand the methodology widely followed in literature, it is useful to consider the spreading code separately, assuming a perfect wipe-off of

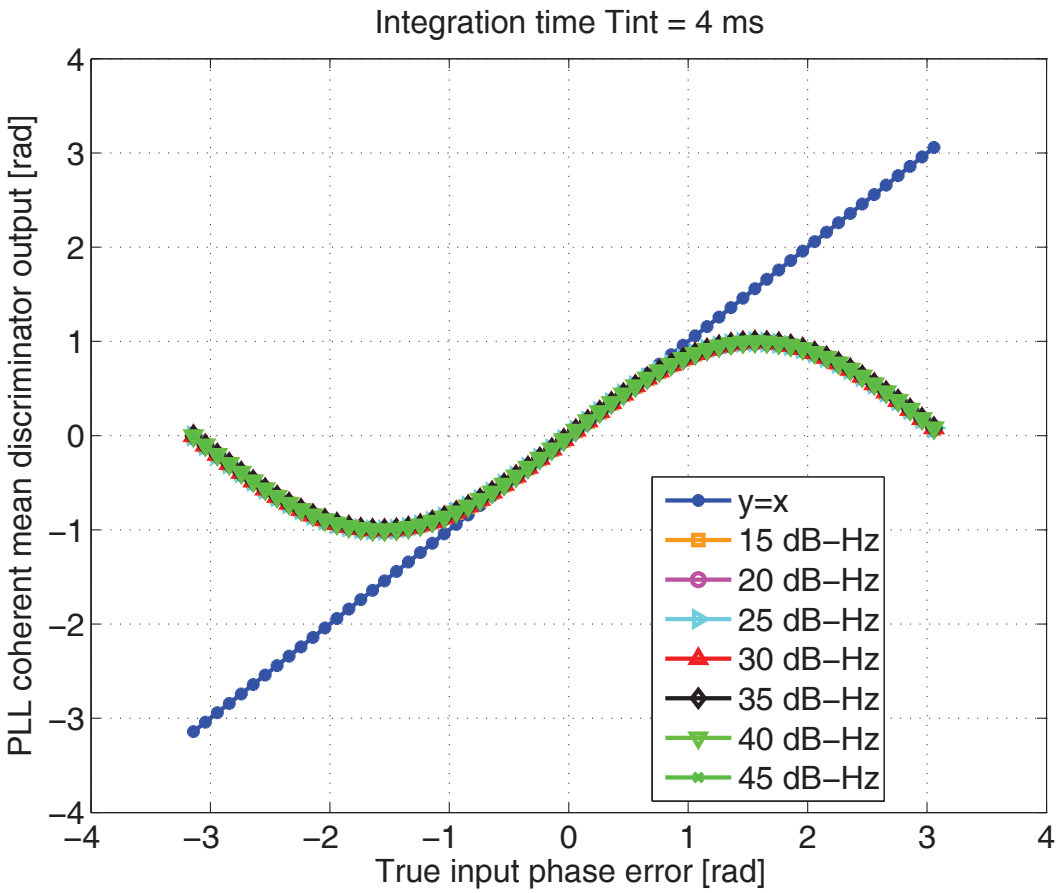


Figure 3.14: Mean DPLL COH discriminator output with $T_I = 4$ ms - showing the dependence of the input-output characteristic from the CNR.

the carrier signal, which is seldom the case in a GNSS receiver. Moreover, in order to acquire or track the signal, consider the correlation of the input signal by the spreading code shifted by an appropriate τ to obtain a correlation value.

Clearly the auto-correlation function as seen in Figures 2.6(a) 2.6(b) is even since the CA code is real, and there is only one point or one chip which corresponds to the unity peak. This means that only one code phase τ in terms of chips can yield the maximum correlation value. However, looking into the auto-correlation in terms of samples, $R(\tau)$ is significantly higher than non-zero for a number of points or samples depending on the sampling frequency of the digital signal. It is thus crucial to select the correct sample which corresponds to the peak yielding the best synchronization and positioning accuracy.
Due to the shape of $R(\tau)$ as explained above, and given that the acquisition engine provides a good estimate of the code phase in the range of the peak of the auto-correlation function (less than a chip), the most natural way to accomplish this is to use several correlators, typically three complex point correlators, early $Z_E = I_E + jQ_E$ prompt Z_P and late Z_L which are separated by an early minus late spacing d_s defined in terms of a fraction of a chip. The prompt correlator uses the PD code phase estimate to compute I_P and Q_P while the early and late correlators are obtained by using shifted versions of the local PRN code by $\pm d_s/2$ chips.

3.4.4.1 Types of DDLL discriminators

There are several DDLL discriminators documented in the literature, some of them use only the early and late correlators while others use all three correlators. The first category of these PD comprises early minus late discriminators while the second category, DP discriminators. Discriminators are typically normalized to yield an unbiased code phase, that is approximately equal to the true code phase. No normalization is applied if it is desired to conduct adaptive processing where the discriminators output reflects changes in the input signal's parameters such as amplitude, correlator integration time, correlator spacing, code chip width, noise power... These correlators hold complex numbers, where the absolute operation applied on the complex early correlator output $|Z_E|$ is equivalent to $\sqrt{I_E^2 + Q_E^2}$. Indeed, the in-phase and quadrature early

point correlators of the DDLL relative to a BOC signal can be written as:

$$I_E[k] = \sum_{n=(k-1)T_I}^{k\cdot T_I} C_E[n]\cdot S_E[n]\cdot \mathrm{Im}((\cos\hat{\phi}_k[n] - j\sin\hat{\phi}_k[n])\cdot s[n]) \quad (3.55)$$

$$Q_E[k] = \sum_{n=(k-1)T_I}^{k\cdot T_I} C_E[n]\cdot S_E[n]\cdot \mathrm{Re}((\cos\hat{\phi}_k[n] - j\sin\hat{\phi}_k[n])\cdot s[n]) \quad (3.56)$$

where both the spreading code and subcarrier are shifted by $d_s/2$ chips. In a similar way, the expressions of the prompt and late correlators can be derived. Typical DDLL discriminators' expressions [37] [2] are listed herein:

- $S_{\mathrm{EMLP}} = |Z_E|^2 - |Z_L|^2$ characterizes the noncoherent Early Minus Late Power (EMLP) discriminator without normalization.
- $S_{\mathrm{EMLPn}} = \dfrac{|Z_E|^2 - |Z_L|^2}{|Z_E|^2 + |Z_L|^2}$ characterizes the noncoherent (EMLP) discriminator with early and late correlators normalization.
- $S_{\mathrm{EMLEn}} = \dfrac{|Z_E| - |Z_L|}{|Z_E| + |Z_L|}$ for the noncoherent Early Minus Late Envelope (EMLE) discriminator with early and late correlators normalization.
- $S_{\mathrm{EMLEnp}} = \dfrac{|Z_E| - |Z_L|}{|Z_P|}$ for the noncoherent EMLE discriminator with prompt correlator normalization.
- $S_{\mathrm{DP}} = (I_E - I_L)I_p + (Q_E - Q_L)Q_p$ for the noncoherent DP discriminator with no normalization.
- $S_{\mathrm{DPn}} = \dfrac{I_E - I_L}{I_p} + \dfrac{Q_E - Q_L}{Q_p}$ for the noncoherent DP discriminator with normalization by the prompt correlators I_P^2 and Q_P^2 respectively.
- $S_{\mathrm{DDDP}} = (I_E - I_L)\mathrm{sign}(I_p) + (Q_E - Q_L)\mathrm{sign}(Q_p)$ for the Decision Directed (DD) coherent DP discriminator with no normalization.
- $S_{\mathrm{EMLc}} = I_E - I_L + Q_E - Q_L$ for the coherent Early Minus Late (EML) discriminator with no normalization.
- $S_{\mathrm{DPc}} = (I_E - I_L)I_p$ for the coherent DP discriminator with no normalization.
- $S_{\mathrm{DDDPc}} = (I_E - I_L)\mathrm{sign}(I_p)$ for the DD coherent DP discriminator with no normalization.

Noncoherent discriminators are insensitive to carrier phase errors and use the sum of the squared value of both in-phase and quadrature correlators $I^2 + Q^2$ or the product of two in-phase correlators. On the other hand, coherent discriminators only use the sum or difference of I and Q correlators as they are.

3.4.4.2 Galileo E1 DDLL auto-correlation and error functions

The Galileo E1 signal with its unique CBOC(6,1,1/11) subcarrier exhibits a multi-peaked auto-correlation function as seen in Figure 2.5, which is typical of BOC signals in general. Figure 3.15 highlights the difference in shape between a BOC(1,1) signal and a CBOC(6,1,1/11), which is mainly a higher slope around the auto-correlation peak coupled with multiple piecewise linear functions characterized by very small slopes.

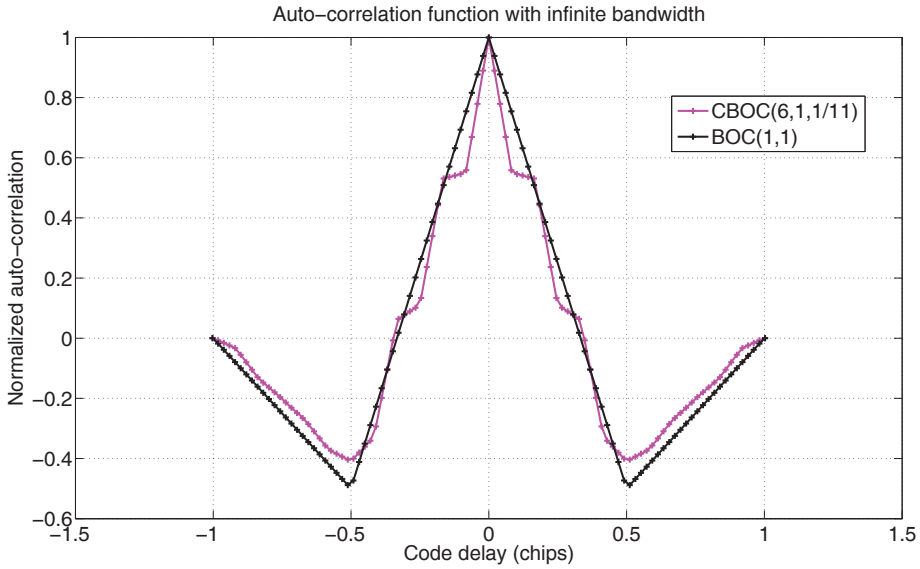


Figure 3.15: Comparison of BOC(1,1) and CBOC(6,1,1/11) auto-correlation functions - a higher slope for the CBOC(6,1,1/11) pilot channel subcarrier, assuming infinite bandwidth.

To assess the consequence of the existence of multiple peaks, it is useful to write the expression of the discriminators or error functions in terms of the auto-correlation function and examine their shape. The expression of the unnormalized typical DDLL discriminators, EMLP EMLE and DP are considered herein. Substituting equations

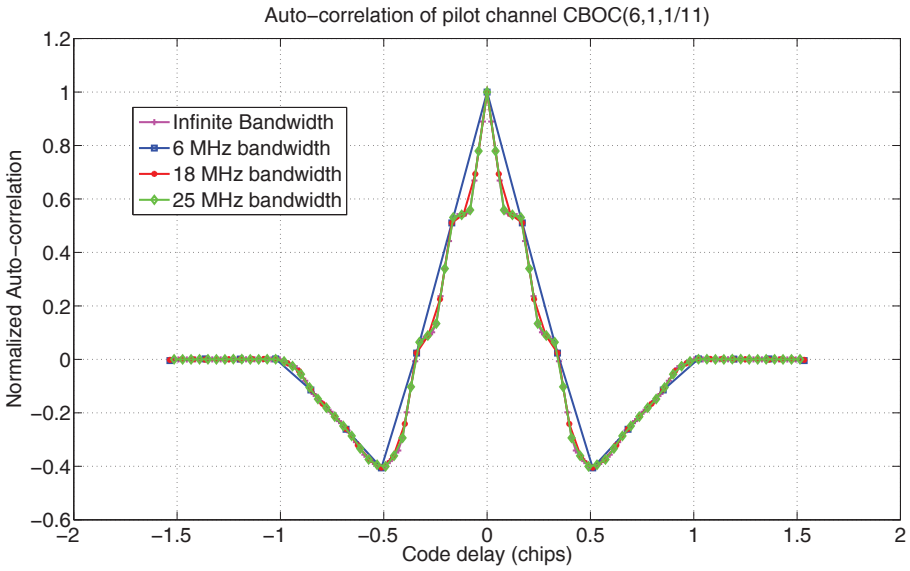


Figure 3.16: Comparison of CBOC(6,1,1/11) auto-correlation function for different front-end bandwidths - the sharp slope of the main peak of the pilot channel disappears for a low bandwidth of 6 MHz.

2.3-2.4 into the EMLP discriminator expression without normalization, results in:

$$S_{\text{EMLP}} = P * (R^2(\tau_E) - R^2(\tau_L)) \tag{3.57}$$

Similarly, the EMLE discriminator yields:

$$S_{\text{EMLE}} = \sqrt{P} * (R(\tau_E) - R(\tau_L)) \tag{3.58}$$

And finally the noncoherent DP discriminator without normalization:

$$S_{\text{DP}} = P * (R(\tau_E) - R(\tau_L)) * R(\tau_P) \tag{3.59}$$

Assuming an infinite front-end bandwidth, Figure 3.17 plots all these three discriminators. A sampling frequency much greater than the chip rate is assumed. It can be seen that all three error functions corresponding to CBOC(6,1,1/11) display two false lock points (zero-crossing points with a positive slope) at around ±0.5 chips error, in addition to the one corresponding to zero code delay error, and this is true for both types of subcarriers BOC(1,1) and CBOC(6,1,1/11). This is the general impact of the BOC signals. Note that the ultimate goal of a DDLL discriminator is to lock into a

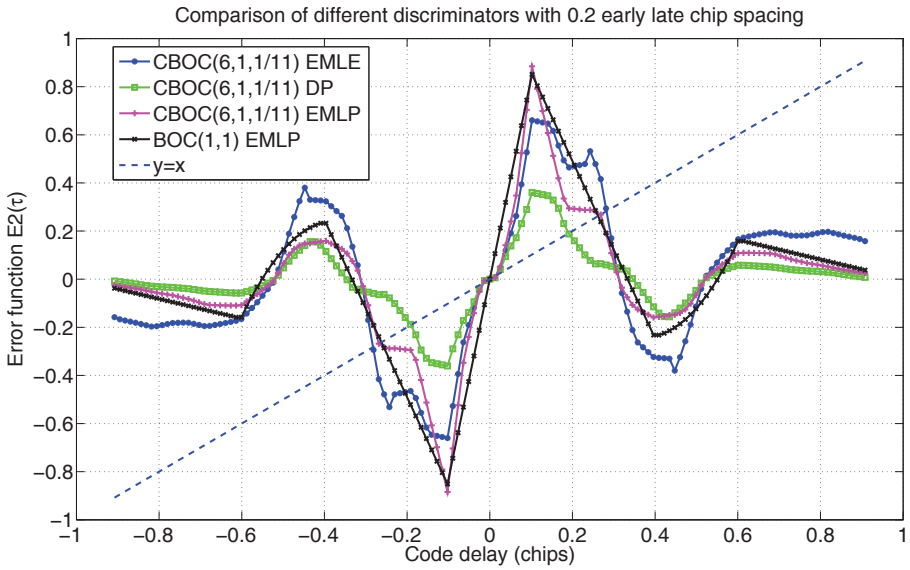


Figure 3.17: Comparison of traditional DDLL discriminators using Galileo CBOC(6,1,1/11) and BOC(1,1) with no normalization - using EMLE, DP, EMLP for CBOC(6,1,1/11) and EMLP for BOC(1,1) subcarriers, assuming an infinite (50 MHz) bandwidth and an early-late spacing of 0.2 chips.

zero-crossing point because it is much easier than find the maximum of a function. Obviously, the zero-crossing point should be unique in the discriminator function to avoid false lock points.

3.4.4.3 Galileo E1 DDLL discriminators

The multiple peaks problem was expected as it is clearly shown in Figure 2.5 that there are two side peaks at ±0.5 chips. These false lock points exist for both types of error functions, EML as well as DP and pose a major threat for a false lock, as well as biased or ambiguous tracking, and ultimately a position error of the order of 150 meters (= 0.5 chips $\cdot 3 \cdot 10^8$ m/s /1.023 Mcps). In order to circumvent the problem of the multiple false lock points and biased tracking, many techniques have been proposed in the literature each with its set of pros and cons.

The single side lobe or SSB technique treats each BOC side lobe as a single BPSK signal, but needs additional hardware [2]. In addition, it suffers a 3dB loss if only

one lobe is tracked, and faces a challenging task of securing sharp roll-off filters for narrow-spaced signals (in terms of bandwidth) when trying to combine both lobes non-coherently. This is the case for BOC(1,1) as well as CBOC(6,1,1/11) signals. Another technique stems from the simple idea of multiplying the received signal by a combination of both in-phase and quadra-phase subcarriers. This technique is termed the SCC technique, it creates a correlation function in terms of steps and a single correlation peak, and is shown to have equal performance with the SSB technique [7]. Other techniques include BOC tracking with multiple gate discriminators MGD [8], the bump jumping algorithm, the ASPeCT algorithm [9], the DE as explained in [2] - [10] and the two-step Galileo CBOC tracking algorithm [11]. The bump jumping algorithm has the downside of taking too long to converge while the ASPeCT and dual estimator which has a third tracking loop dedicated to the subcarrier both yield a more complex loop implementation.
Considering that both SSB and SCC do not either fully or efficiently exploit the subcarrier presence, and that BJ, ASPeCT and DE are either not efficient or add too much complexity, the MGD seems a good alternative. For that reason, in the following, the VEMLE discriminator will be examined which is a MGD showing an unambiguous curve by using two pairs of early and late correlators, i.e. the usual early and late correlators and two additional very early and very late correlators.

3.4.4.4 Four taps VEMLE discriminator

The basic idea of the simple four taps VEMLE discriminator is to use four points of the Galileo CBOC(6,1,1/11) code auto-correlation function in order to suppress the side peaks at ±0.5 chips as seen in Figure 3.17. Different versions of this main idea can be implemented, as each sample of the auto-correlation function can be weighted by different values [41]. The simple VEMLE discriminator expression can be defined as:

$$S_{VEMLE} = \sqrt{I_E^2 + Q_E^2 + I_{VE}^2 + Q_{VE}^2} - \sqrt{I_L^2 + Q_L^2 + I_{VL}^2 + Q_{VL}^2} \tag{3.60}$$

The corresponding discriminator or error function using an early minus late spacing of 0.5 chips and a very early minus very late chip spacing of 1 chip is shown in Figure 3.18. For the discriminator to yield unbiased code phase error estimates, a normalization process is applied, that is typically:

$$N_{VEMLE} = \sqrt{I_E^2 + Q_E^2 + I_{VE}^2 + Q_{VE}^2} + \sqrt{I_L^2 + Q_L^2 + I_{VL}^2 + Q_{VL}^2} \tag{3.61}$$

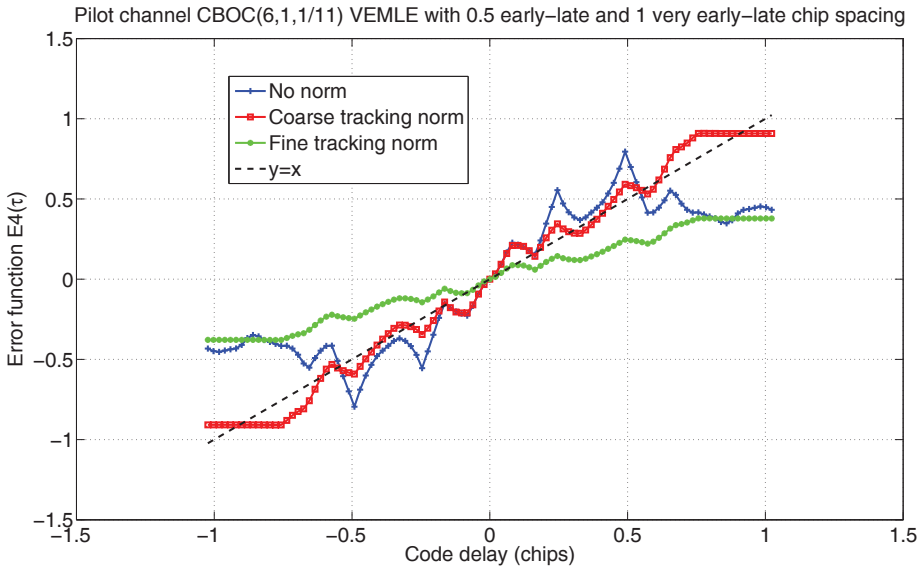


Figure 3.18: Different normalization strategies for the VEMLE discriminator - no normalization, normalization for coarse tracking, and normalization for fine tracking around zero code delay error.

This normalization is performed using the (very early, early, late, very late) correlators, ideally spaced at (-0.5, -0.25, 0.25, 0.5) chips from the true code delay. In addition, a normalization factor is needed such that the piece-wise linear discriminator function has a unity slope for a specific piece in consideration. In fact, the idea of coarse and fine tracking as mentioned in [11] stems from the fact that the discriminator function can be approximated by a collection of several piece-wise linear functions which have different slopes. This is due to the CBOC(6,1,1/11) auto-correlation shape as shown in Figure 3.15. Due to this CBOC shape, namely the quasi zero slope of the auto-correlation function main peak around ±0.1 chips, there is an inherent risk of confining to sub-optimal code tracking, incurring a position error of the order of 150 meters as previously noted.

For this reason, coarse and fine tracking need to be carried out in a special setting. Coarse tracking ensures a wide linear range between ±0.5 chips albeit with some undulations and partially between ±1 chips, as shown in Figure 3.18. In the same figure, it can be seen that fine tracking, on the other hand, ensures a narrow linear range but specifically concentrated around zero code delay error. This yields fine estimates of

the code phase error. Coarse tracking is initiated and a test is performed to assess the feasibility of moving to fine tracking. In other words, this needs checking the range of code delay values where the tracking is being performed, which can be easily done by monitoring the outputs of the VEMLE discriminator during tracking. A simple arbitrary monitoring test consists of looking back over the last second of tracking, and moving to fine tracking if 90% of the code delay values are less than $d_s/4 = 0.125$ chips. On the other hand, another condition has to be satisfied to remain in fine tracking, that is checking that 50% of the code delay values in the last 100 ms are less than 0.125 chips.

The normalization factors depend on the slope of the code auto-correlation function main peak as well as the early minus late chip spacing. Analytical expressions of these normalization factors can be found in [11] where an additional multiplicative factor of 2 is used for coarse tracking, and 0.8333 for fine tracking. Figure 3.18 shows all three versions, without normalization, normalization for coarse and fine tracking. Moreover, it is essential not to underestimate the role of the front-end bandwidth in the discriminator's shape. In fact, it has been seen in Figure 3.16 that the sharp slope of the auto-correlation peak exhibited by the CBOC subcarrier is reduced to the value of the BOC(1,1) when a 6 MHz bandwidth is chosen. As can be seen in Figure 3.19, in case of coarse tracking, the discriminators relative to 18 MHz and 25 MHz bandwidth are very close to the infinite bandwidth case and closely follow a piece-wise unity slope line. These ondulations cause a longer transient time than the usual BPSK signal case. The error function relative to the 6 MHz bandwidth front-end is more problematic as it does not fit closely to the unity slope piece-wise linear function. In fine tracking mode however, the difference between 18 and 25 MHz bandwidth performance is much clearer as shown in Figure 3.20, and it can be said that a 25 MHz bandwidth is enough to capture and harness the CBOC auto-correlation shape.

3.4.4.5 Galileo E5a/b DDLL discriminators

The Galileo E5a or E5b spreading code signals are basically BPSK signals when considered separately instead of the whole Galileo E5 AltBOC signal. After wiping off the carrier, the Galileo E5a or E5b code which is a PRN sequence is characterized by an auto-correlation function with a single peak at zero time lag as can be seen in Figure 2.9. Figure 3.21 shows a comparison between the BOC(1,1) and Galileo E5a BPSK

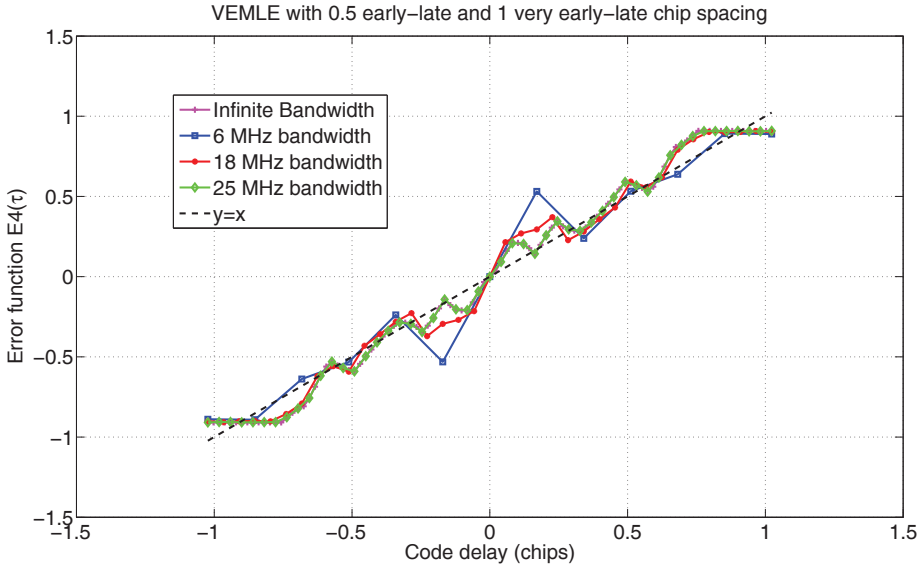


Figure 3.19: VEMLE discriminator performance during coarse tracking for different bandwidths - 6 MHz, 18 MHz, 25 MHz and a theoretical infinite bandwidth of 50 MHz.

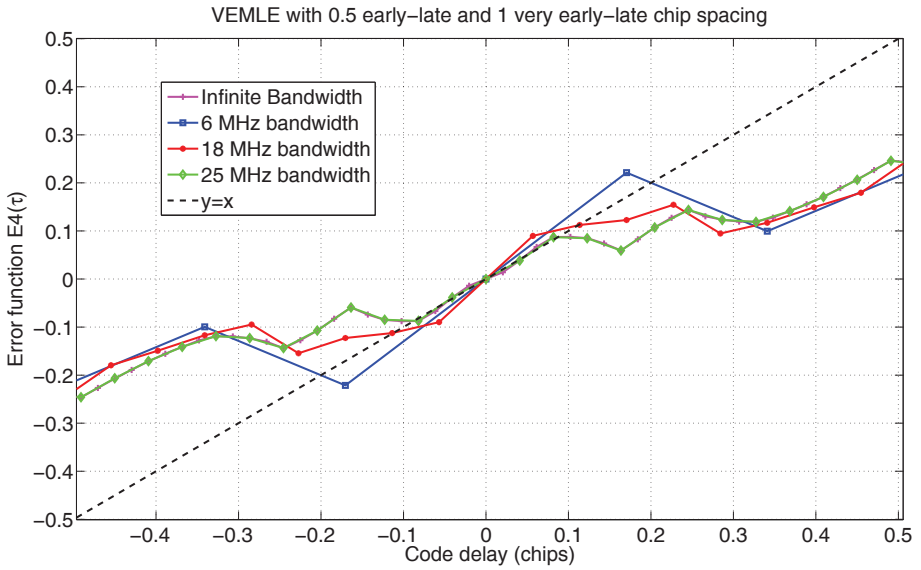


Figure 3.20: VEMLE discriminator performance during fine tracking for different bandwidths - 6 MHz, 18 MHz, 25 MHz and a theoretical infinite bandwidth of 50 MHz.

auto-correlation function using a large bandwidth. On the other hand, Figure 3.22 compares the auto-correlation function for different bandwidth values.

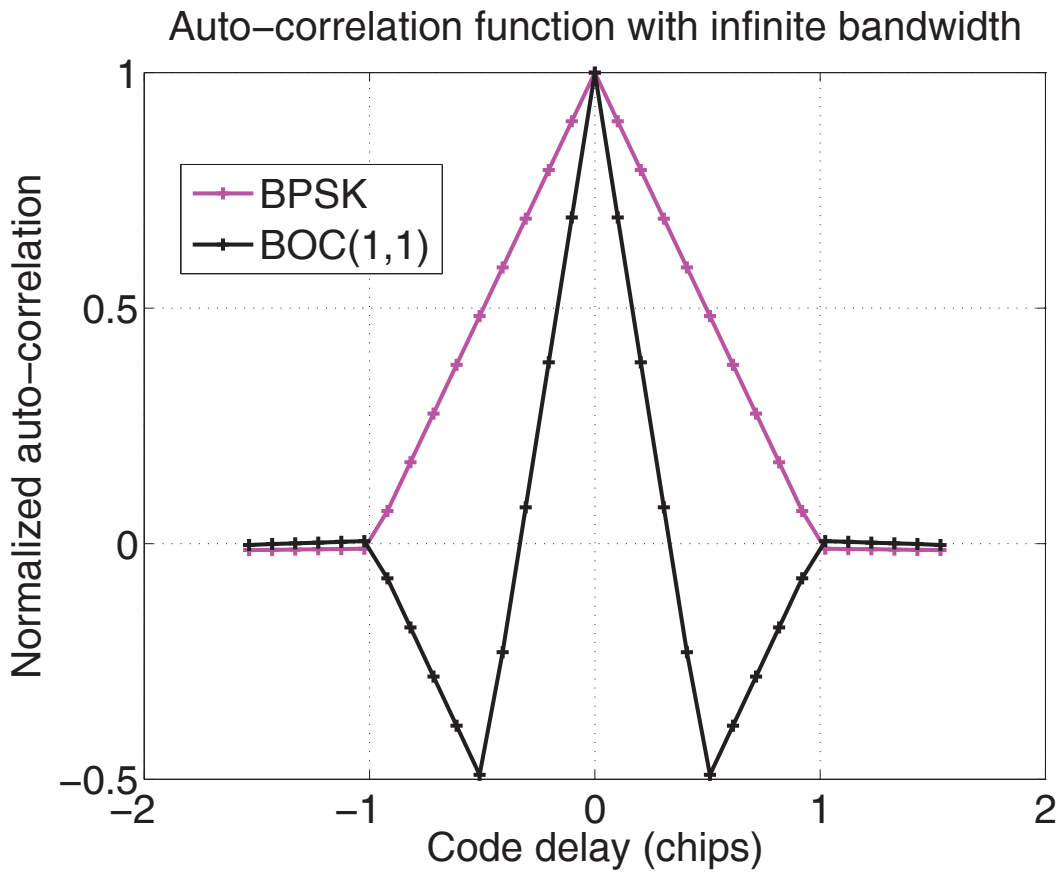


Figure 3.21: Comparison of BOC(1,1) and Galileo E5a BPSK auto-correlation functions - using a large bandwidth, a higher slope for the BOC(1,1) with respect to the BPSK E5a spreading code.

Given the expression of the unnormalized discriminators or error functions in Equations 3.57 - 3.59 in terms of the corresponding auto-correlation functions seen in Figures 2.9 and 2.10, it is possible to examine the discriminator shape by simulation. The unnormalized expressions of the typical DDLL discriminators presented earlier, EMLP EMLE and DP are considered herein.

Assuming an infinite or relatively large front-end bandwidth of 100 MHz, Figure 3.23 shows a plot of all these three discriminators. A sampling frequency much greater than the chip rate is assumed. As expected, the linear tracking region extends over half the

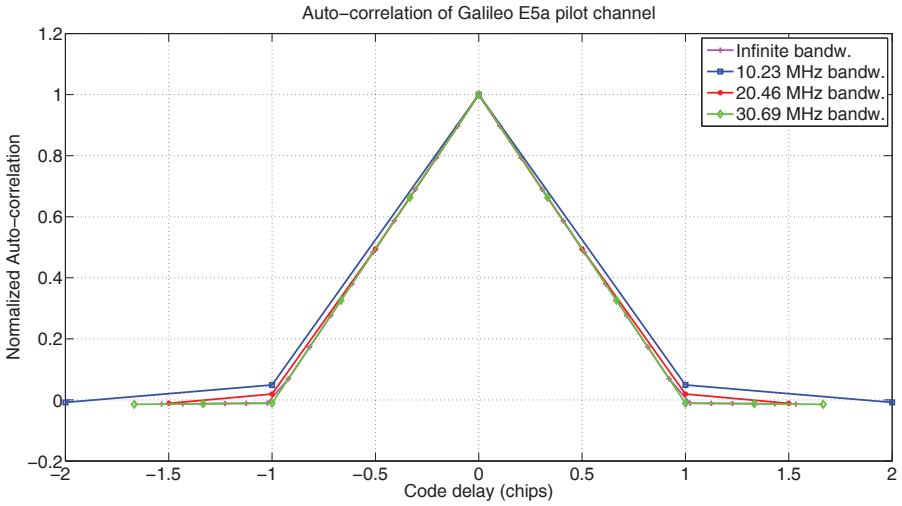


Figure 3.22: Comparison of Galileo E5a code auto-correlation function for different front-end bandwidths - 20.46 MHz is the minimum bandwidth.

early late chip spacing which is 0.5 chip, hence around ±0.25 chip error. This is true for the EMLE and EMLP discriminators whereas the DP discriminator has a very narrow linear range around ±1/10 chip error.

3.4.5 Galileo E1 DE discriminators

As stated in section 3.4.4.3, numerous approaches have been presented in literature, to deal with the ambiguous code tracking due to the BOC signal. The DE in particular, introduces an additional locked loop, the Subcarrier Locked Loop (SLL) that tracks the subcarrier in unison with the PLL and DLL. It is clear that both the subcarrier and spreading code are perfectly aligned or synchronized as explained in section 2.2.1, however, it is also possible to separate those signals in tracking, in order to get rid of the ambiguity in the correlation function. Tracking the spreading code and subcarrier separately in a dual estimator DE fashion means adding a Digital Subcarrier Locked Loop (DSLL) in addition to the DDLL [10]. This is possible due to the periodic nature of the subcarrier over the spreading code chips. Given that the period of the spreading code T_c is a multiple of the CBOC(6,1,1/11) subcarrier subchip counterpart $T_c = 12T_{sc}$, it is possible to adjust the estimated spreading code delay $\hat{\tau}_c$ by the estimated subcarrier code delay $\hat{\tau}_{sc}$ which offers higher resolution and lower tracking Root Mean Square

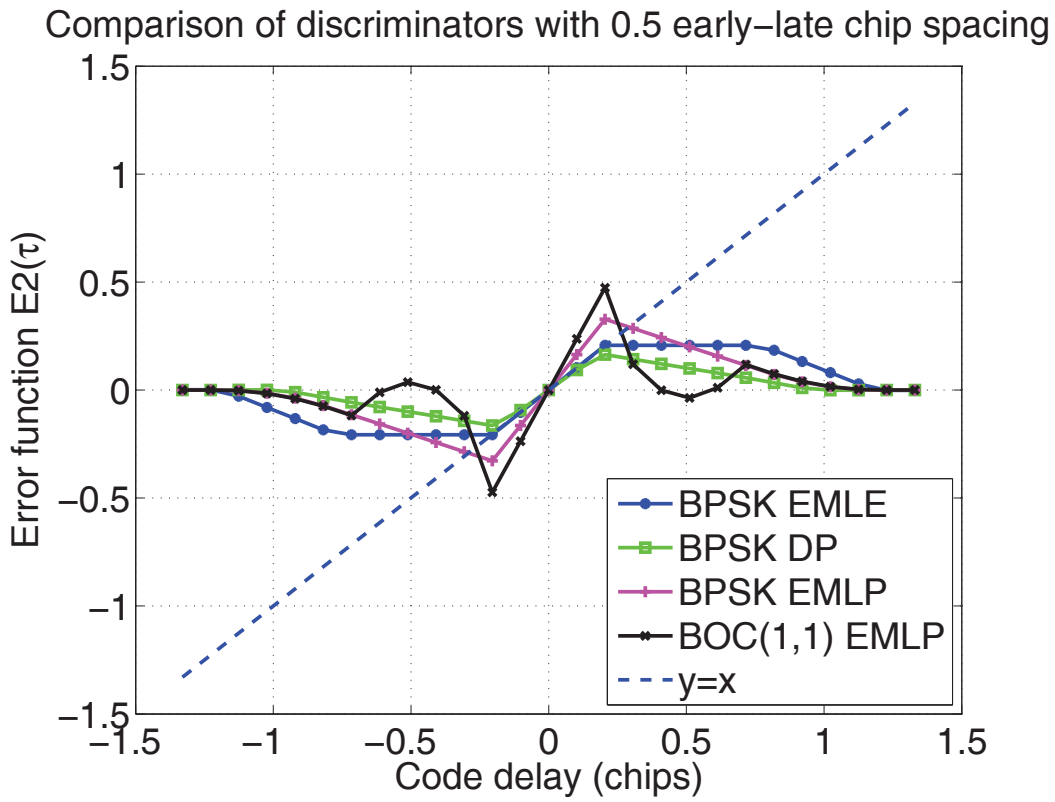


Figure 3.23: Comparison of traditional DDLL discriminators applied on Galileo E5a BPSK signals and theoretical BOC(1,1) signals - with a constant normalization of 1/2, an early minus late chip spacing of 0.5 chips and EMLE, DP, EMLP discriminators applied on BPSK signals and only EMLP on BOC(1,1) signals, assuming an infinite (100 MHz) bandwidth.

(RMS) jitter (because of the faster subcarrier modulation):

$$\hat{\tau}_c^+ = \hat{\tau}_{sc} + round\left(\frac{\hat{\tau}_c - \hat{\tau}_{sc}}{T_{sc}}\right) T_{sc} \tag{3.62}$$

The DSLL discriminator would thus use a prompt version of the spreading code (estimated by the DDLL) and an early and late version of the subcarrier. In other words, the in-phase and quadrature early correlator of the DSLL can be written as:

$$I_{SE}[k] = \sum_{n=(k-1)T_I}^{k \cdot T_I} C_P[n] \cdot S_E[n] \cdot \mathrm{Im}((\cos\hat{\phi}_k[n] - j\sin\hat{\phi}_k[n]) \cdot s[n]) \tag{3.63}$$

$$Q_{SE}[k] = \sum_{n=(k-1)T_I}^{k \cdot T_I} C_P[n] \cdot S_E[n] \cdot \mathrm{Re}((\cos\hat{\phi}_k[n] - j\sin\hat{\phi}_k[n]) \cdot s[n]) \tag{3.64}$$

Similarly, the in-phase and quadrature late correlator of the DSLL are expressed as:

$$I_{SL}[k] = \sum_{n=(k-1)\cdot T_I}^{k \cdot T_I} C_P[n] \cdot S_L[n] \cdot \mathrm{imag}((\cos(\hat{\phi}_k[n]) - j\sin(\hat{\phi}_k[n])) \cdot s[n]) \tag{3.65}$$

$$Q_{SL}[k] = \sum_{n=(k-1)\cdot T_I}^{k \cdot T_I} C_P[n] \cdot S_L[n] \cdot \mathrm{real}((\cos(\hat{\phi}_k[n]) - j\sin(\hat{\phi}_k[n])) \cdot s[n]) \tag{3.66}$$

where C_P is the prompt spreading code, which corresponds to the true estimated code phase and S_E S_L are the early and late versions of the subcarrier signal which correspond to the true estimated subcarrier phase shifted by the early late subcarrier chip spacing d_{sc}.

On the other hand, the DDLL uses the prompt version of the subcarrier S_P (estimated by the DSLL) and an early and late version of the spreading code C_E C_L. Therefore, the in-phase and quadrature early correlators of the DDLL in the DE are expressed as:

$$I_{CE}[k] = \sum_{n=(k-1)\cdot T_I}^{k \cdot T_I} S_P[n] \cdot C_E[n] \cdot \mathrm{imag}((\cos(\hat{\phi}_k[n]) - j\sin(\hat{\phi}_k[n])) \cdot s[n]) \tag{3.67}$$

$$Q_{CE}[k] = \sum_{n=(k-1)\cdot T_I}^{k \cdot T_I} S_P[n] \cdot C_E[n] \cdot \mathrm{real}((\cos(\hat{\phi}_k[n]) - j\sin(\hat{\phi}_k[n])) \cdot s[n]) \tag{3.68}$$

Similarly, the in-phase and quadrature late correlators of the DDLL are expressed as:

$$I_{CL}[k] = \sum_{n=(k-1)\cdot T_I}^{k \cdot T_I} S_P[n] \cdot C_L[n] \cdot \mathrm{imag}((\cos(\hat{\phi}_k[n]) - j\sin(\hat{\phi}_k[n])) \cdot s[n]) \tag{3.69}$$

$$Q_{CL}[k] = \sum_{n=(k-1)\cdot T_I}^{k \cdot T_I} S_P[n] \cdot C_L[n] \cdot \mathrm{real}((\cos(\hat{\phi}_k[n]) - j\sin(\hat{\phi}_k[n])) \cdot s[n]) \tag{3.70}$$

The prompt correlator is just the product of a prompt version of all three signals, i.e. subcarrier, spreading code and carrier.

$$I_{CP}[k] = \sum_{n=(k-1)\cdot T_I}^{k\cdot T_I} S_P[n]\cdot C_P[n]\cdot \text{imag}((\cos(\hat{\phi}_k[n]) - j\sin(\hat{\phi}_k[n]))\cdot s[n]) \quad (3.71)$$

$$Q_{CP}[k] = \sum_{n=(k-1)\cdot T_I}^{k\cdot T_I} S_P[n]\cdot C_P[n]\cdot \text{real}((\cos(\hat{\phi}_k[n]) - j\sin(\hat{\phi}_k[n]))\cdot s[n]) \quad (3.72)$$

3.4.5.1 Ideal DSLL and DDLL auto-correlation and cross-correlation functions

Along the lines of section 3.4.4, it is useful to assume that a perfect estimation of the spreading code delay is performed which results in a perfect wipe-off of the spreading code. This is because the spreading code values are typically ±1. In these conditions, the auto-correlation function of the Galileo E1 CBOC signal reduces to the auto-correlation function of its subcarrier as shown in Figure 3.24 where different front-end bandwidth values are compared. Assuming near infinite bandwidth, Figure 3.25 compares the BOC(1,1) to the CBOC(6,1,1/11) subcarrier auto-correlation functions.

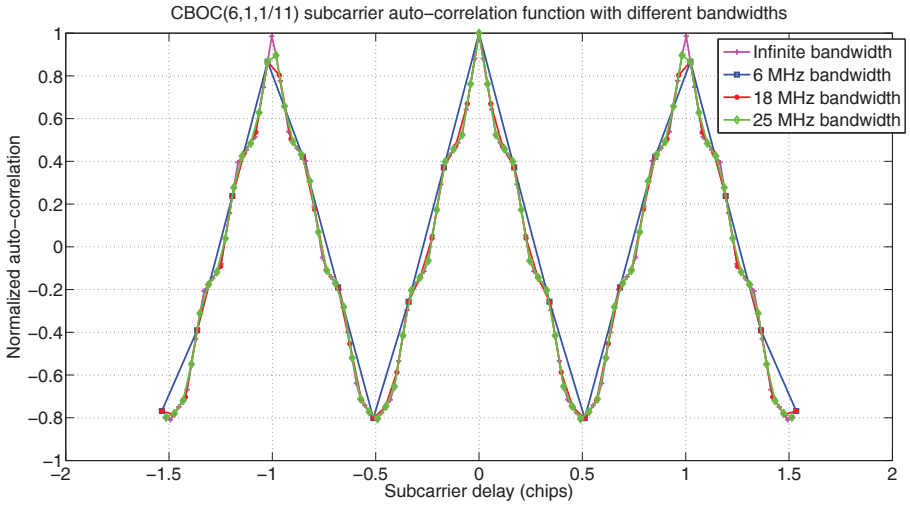


Figure 3.24: The CBOC(6,1,1/11) subcarrier auto-correlation function - for different front-end bandwidths.

However, assuming a perfect estimation of the subcarrier phase does not result in a perfect wipe-off of the subcarrier as it takes on values of $\pm(\alpha \pm \beta)$. Therefore, the

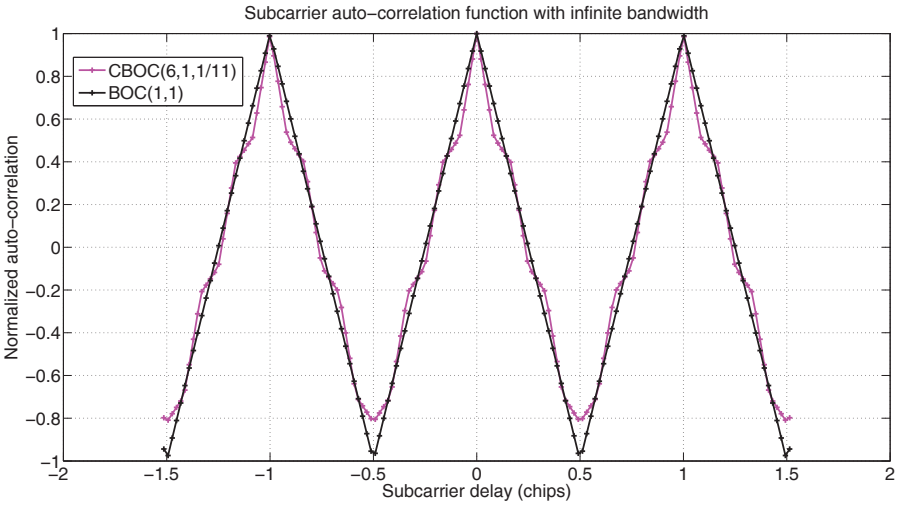


Figure 3.25: A comparison of BOC(1,1) and CBOC(6,1,1/11) subcarrier auto-correlation function - using infinite bandwidth.

auto-correlation function of the Galileo E1 CBOC(6,1,1/11) signal reduces to a cross-correlation between a modified version of the spreading code and the local spreading code which is a BPSK signal. Figure 3.26 shows indeed that some ondulations exist around the resulting cross-correlation function which is comparable to the well-noted triangular function of BPSK signals. In fact, in case of a BOC(1,1) subcarrier which takes on ± 1 values, it is expected to obtain an auto-correlation function of the spreading code characteristic of a triangular function of BPSK signals.

3.4.5.2 DSLL and DDLL discriminators

Comparison of different versions of EML and DP discriminators as listed in Subsection 3.4.4.1 inside the DSLL is shown in Figure 3.27. It can be seen several false locks at multiples of ± 0.5 chips which induce 150 meters of positioning error as pointed out earlier in Section 3.4.4.3. For this reason, a pure coherent EML discriminator with no normalization is used in the DSLL, defined as:

$$S_{EML} = I_{SE} - I_{SL} + Q_{SE} - Q_{SL} \tag{3.73}$$

It is further normalized by a normalization factor of 1/8 to yield a unity slope line, with unbiased subcarrier delay estimates as shown in Figure 3.28 where an infinite

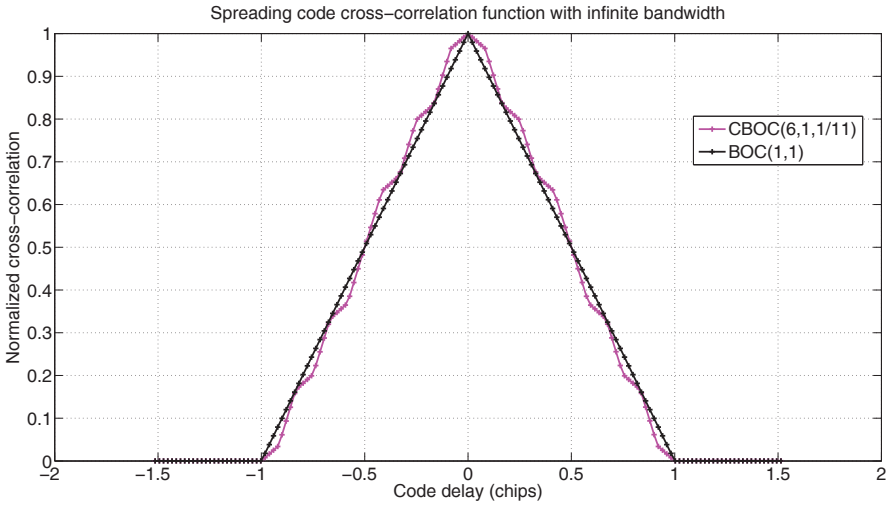


Figure 3.26: The Galileo E1 spreading sequence cross-correlation function - after multiplying by the subcarrier, either BOC(1,1) and CBOC(6,1,1/11) and assuming infinite bandwidth.

bandwidth is assumed. The DSLL pure EML discriminator is further shown in Figure 3.29 considering a set of bandwidth values 6, 18, and 25 MHz. The linear tracking region is shown to become narrower with respect to the infinite bandwidth case, going as low as half of the early minus late chip spacing 0.5/2=0.25 chips.

On the other hand, the DDLL EML code PD used in the DE, does not have the false locking issue as shown in Figure 3.30. A comparison of different discriminators with an early late chip spacing of 0.25 chips shows that unbiased code phase estimates are obtained if a normalization factor of 1/2 is applied for all except the power discriminators where the normalization is 1/3 factor. Figures 3.31 shows the EML DDLL used in a dual estimator in the case of infinite bandwidth (comparing CBOC(6,1,1/11) with BOC(1,1)). Figure 3.32 on the other hand, shows the same discriminator function considering the CBOC signal only and assuming a set of bandwidth values. Similar to the DSLL, the linear tracking region of the DDLL is shown to become narrower with respect to the infinite bandwidth case, going as low as half of the early minus late chip spacing 0.5/2=0.25 chips.

The code and subcarrier delay errors are then filtered by the first order loop filter which estimates the spreading code frequency or chip rate f_c and the subcarrier frequency f_{sc}.

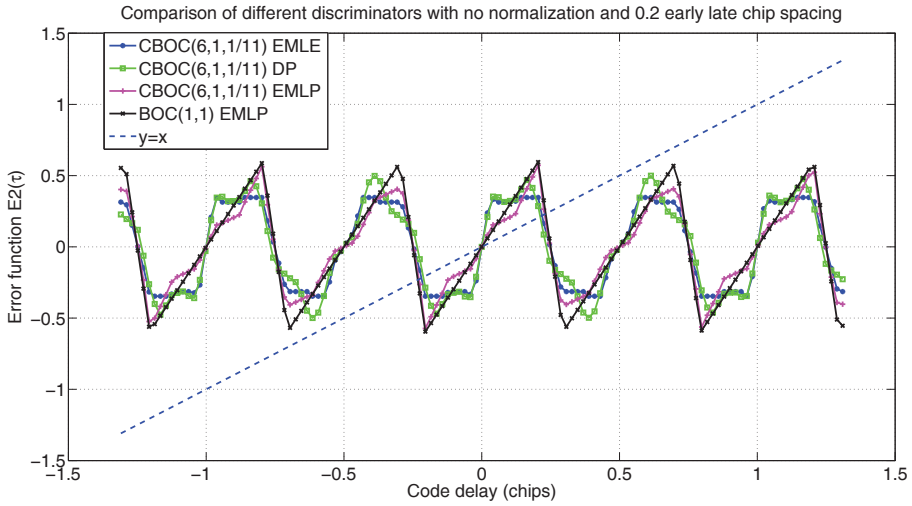


Figure 3.27: Comparison of DSLL discriminators after wiping off the spreading code - using EMLE, DP, EMLP for CBOC(6,1,1/11) and EMLP for BOC(1,1) subcarriers, assuming an infinite (50 MHz) bandwidth and an early-late spacing of 0.2 chips without any normalization.

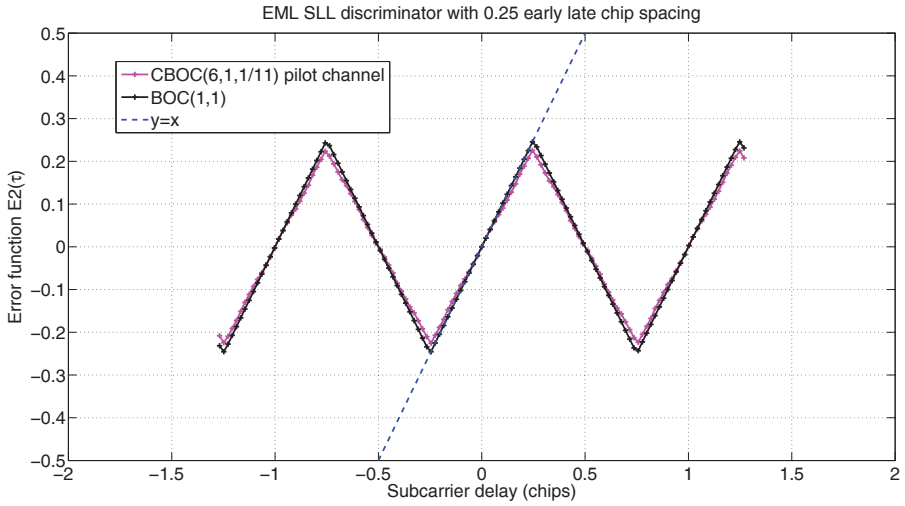


Figure 3.28: Normalized EML discriminator used in DSLL dealing with CBOC(6,1,1/11) subcarrier with infinite bandwidth - and an early late chip spacing of 0.25 chips and a constant normalization factor of 1/8.

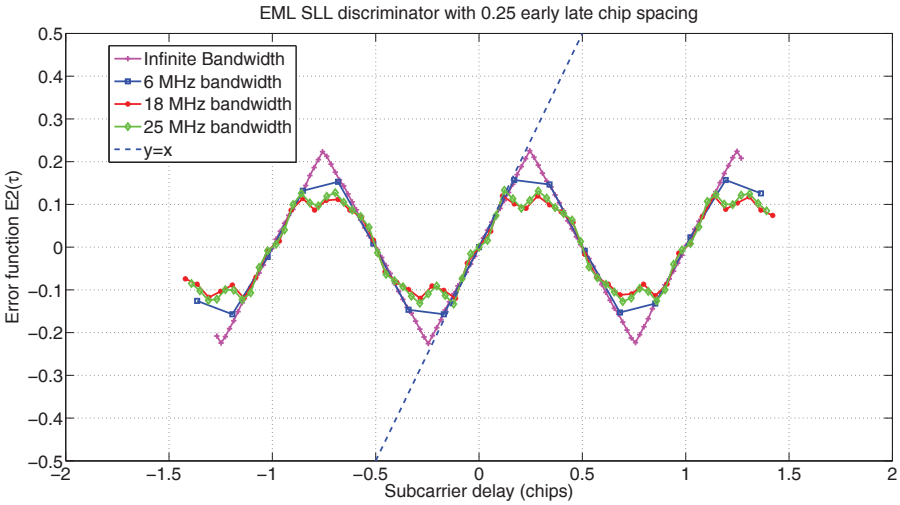


Figure 3.29: Normalized EML discriminator used in DSLL dealing with CBOC(6,1,1/11) subcarrier with different bandwidths - and an early late chip spacing of 0.25 chips and a constant normalization factor of 1/8.

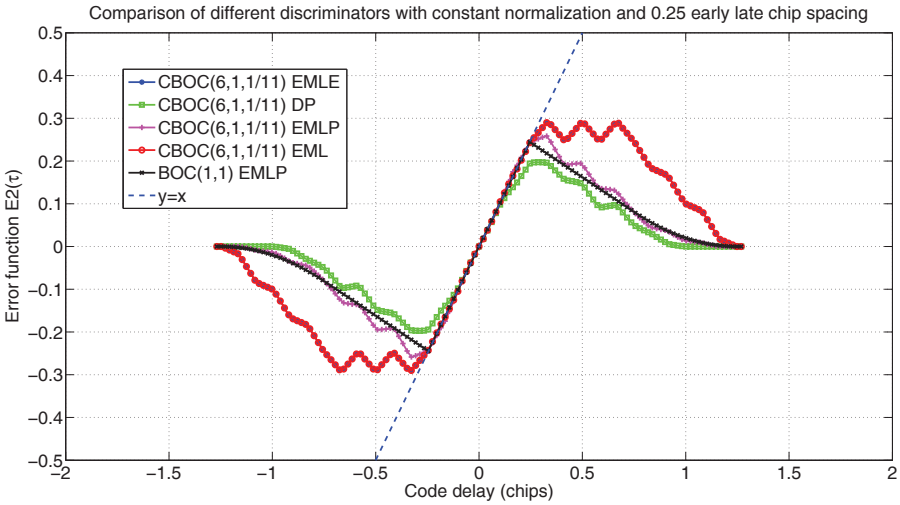


Figure 3.30: Comparison of DDLL discriminators in a DE dealing with CBOC(6,1,1/11) signals - using an early late chip spacing of 0.25 chips, a constant normalization and infinite bandwidth.

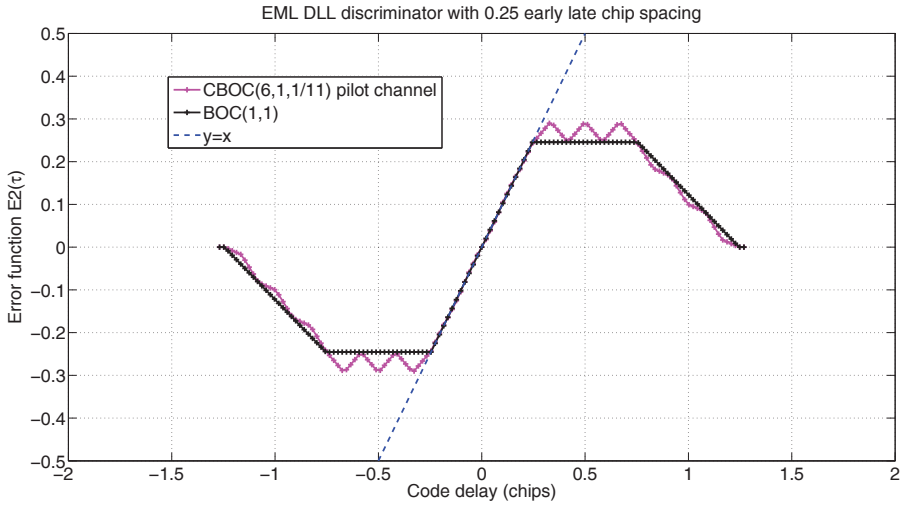


Figure 3.31: Normalized EML discriminator used in a dual estimator DDLL with infinite bandwidth - and an early late chip spacing of 0.25 chips and a constant normalization factor of 1/2.

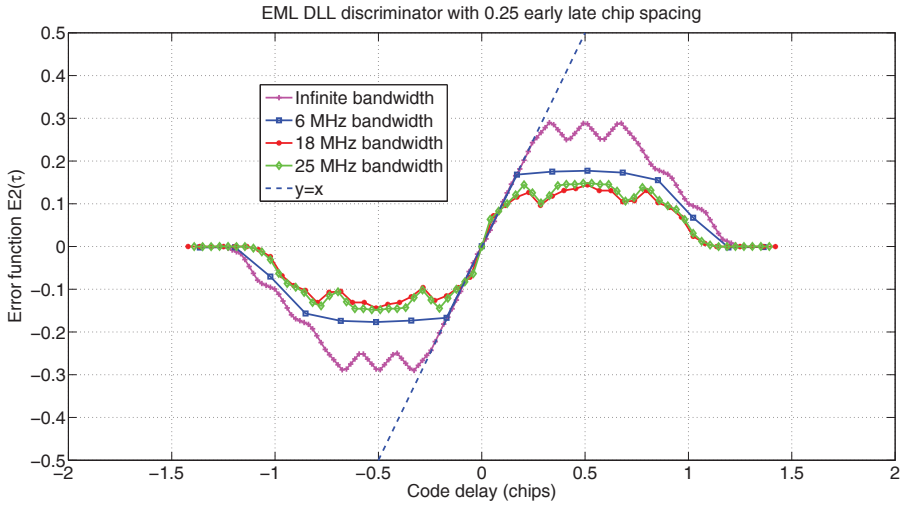


Figure 3.32: Normalized EML discriminator used in a dual estimator DDLL with different bandwidths - and an early late chip spacing of 0.25 chips and a constant normalization factor of 1/2.

These frequency estimates include the carrier Doppler frequency effect and are used to generate a precise code frequency at every integration interval. The expression of the estimated spreading code / subcarrier delay in terms of code/subcarrier frequency is:

$$\hat{\tau}_c = \hat{\tau}_{co} - (\hat{f}_c - 12f_c)T_I F_s/(12f_c) \tag{3.74}$$

$$\hat{\tau}_{sc} = \hat{\tau}_{co} - (\hat{f}_{sc} - 12f_c)T_I F_s/(12f_c) \tag{3.75}$$

where the additional subscript o designates the original code/subcarrier phase estimate returned by the acquisition engine. Substituting Equations 3.74 and 3.75 into 3.62 yields the aforementioned precise code frequency estimate:

$$\hat{f}_c^+ = \hat{f}_{sc} - \frac{1}{T_I F_S} round[(\hat{f}_{sc} - \hat{f}_c)T_I F_S] \tag{3.76}$$

Based on this frequency, the NCO generates both subcarrier and spreading code reference signals to pass it on to the inputs of the DSLL and DDLL respectively.

3.5 Loop filter design

In the GNSS literature dealing with phase locked loops, loop filters are often chosen to be Infinite Impulse Response (IIR) filters and designed in the analog domain. The main function of the loop filter is to smooth the PD output and filter out the AWGN by averaging previous and current PD outputs or loop filter inputs. Digital locked loops of first, second and third order are able to track with zero steady state error, a phase step, a phase ramp and a frequency ramp inputs respectively. In case the computation delay D is zero, the number of poles of the loop transfer function determines the order of the FLL/PLL/DLL. The type of the loop on the other hand is defined as the number of integrators used in the loop [35].
Numerous loop filters are considered in the GNSS literature [37] [42], without taking care of the consequences of the approximation or transformation of the optimum loop filter from analog to digital domain as previously mentioned in Section 3.2 [36]. Indeed, it has been common place in literature to use loop filters designed in the analog domain and resort to methods which transform these filters into digital ones, i.e. box-car mapping or impulse invariance and bilinear transformation, instead of designing digital loop filters from scratch. A frequent criteria used to design these analog loop filters [38] is to minimize both phase output noise power and transient energy. Another criteria could be to only minimize the transient energy or the output noise power separately. In the following, those loop filters adopted in the literature are visited and a derivation of the optimum digital loop filters for a phase step, phase ramp and frequency ramp inputs is presented showing the zeros and poles locations as well as a performance comparison between loop filters designed in the analog vs digital domain.

3.5.1 Classical mapped digital loop filters

Zero, first, and second order loop filter characteristics used in the first commercial GPS receiver design are reported in [37] whereas only first order loop filters (usually resulting in second order loops) are considered in [42]. However, these loop filters are designed in the analog domain and in order to obtain a discrete transfer function, that is a digital implementation, it is necessary to apply analog to digital mapping, either the boxcar digital mapping (impulse invariance) or the bilinear mapping [37]. It is shown in [5] that the boxcar digital mapping results in the mapping of the left hand S plane into a

small circle in the right hand side of the unit circle whereas the bilinear mapping maps the entire left hand S plane into the unit circle. It is also shown that the percentage increase in desired bandwidth due to digital mapping is greater for a boxcar mapping as compared to a bilinear mapping [5]. For that reason, in the following, a derivation of the digital implementation is presented using a bilinear mapping:

$$s = \frac{2}{T_I}\frac{1-z^{-1}}{1+z^{-1}} \tag{3.77}$$

3.5.1.1 Zero order loop filter

A zero order filter is represented by a constant coefficient and as reported in [37], this constant is equal to the natural frequency w_n related to the noise equivalent bandwidth $B_N = w_n/4$. Setting B_N is equivalent do define the zero order loop filter coefficient as:

$$F_0(z) = 4B_N \tag{3.78}$$

which, in time domain, is equivalent to the output:

$$y_0[k] = 4B_N x_0[k] \tag{3.79}$$

3.5.1.2 First order loop filter

The first order loop filter in the analog domain is defined in [37]:

$$F_1(s) = \frac{{w_n}^2}{s} + a_2 w_n = \frac{{w_n}^2 + a_2 w_n s}{s} \tag{3.80}$$

Applying the bilinear transformation directly on this analog loop filter by substituting Equation 3.77 into $F_1(s)$:

$$F_1(s) \overset{s=\frac{2}{T_I}\frac{1-z^{-1}}{1+z^{-1}}}{\longmapsto} F_1(z) \tag{3.81}$$

yields the digital loop filter transfer function:

$$\begin{aligned} F_1(z) &= \frac{T_I(1+z^{-1})}{2(1-z^{-1})}\left({w_n}^2 + \frac{2a_2 w_n}{T_I}\frac{1-z^{-1}}{1+z^{-1}}\right) \\ &= \frac{T_I {w_n}^2(1+z^{-1}) + 2a_2 w_n(1-z^{-1})}{2(1-z^{-1})} \\ &= \frac{A_1 + A_2 z^{-1}}{1-z^{-1}} \end{aligned} \tag{3.82}$$

which, in the time domain, is equivalent to the output:

$$y_1[k] = y_1[k-1] + A_1 x_1[k] + A_2 x_1[k-1] \tag{3.83}$$

and where $a_2 = \sqrt{2}$ and the loop filter coefficients are given by:

$$A_1 = \frac{T_I {w_n}^2}{2} + a_2 w_n; \quad A_2 = \frac{T_I {w_n}^2}{2} - a_2 w_n \tag{3.84}$$

The natural frequency on the other hand, is determined by the choice of the noise equivalent bandwidth B_N:

$$w_n = \frac{4 a_2 B_N}{1 + {a_2}^2} \tag{3.85}$$

The loop filter parameters obtained as such will be referenced by method 1 in the following section where a performance comparison is carried out between various first order loop filters.

A different analog first order loop filter is proposed in [35] and [42]:

$$F_1(s) = \frac{1}{s} \frac{\tau_2 s + 1}{\tau_1} \tag{3.86}$$

and where the digital domain loop filter is derived after performing bilinear mapping on the resulting system analog transfer function $H_1(s)$:

$$H_1(s) \overset{s=\frac{2}{T_I}\frac{1-z^{-1}}{1+z^{-1}}}{\longmapsto} H_1(z) \tag{3.87}$$

which yields a loop filter transfer function:

$$F_1(z) = \frac{C_1 + C_2 - C_1 z^{-1}}{1 - z^{-1}} \tag{3.88}$$

where

$$C_1 = \frac{1}{K_D} \frac{8\zeta w_n T_I}{4 + 4\zeta w_n T_I + (w_n T_I)^2}, \tag{3.89}$$

$$C_2 = \frac{1}{K_D} \frac{4 (w_n T_I)^2}{4 + 4\zeta w_n T_I + (w_n T_I)^2} \tag{3.90}$$

and K_D is the loop gain, ζ is the damping ratio often found in control theory and the natural frequency w_n defined by:

$$w_n = \frac{8\zeta B_N}{4\zeta^2 + 1} \tag{3.91}$$

Indeed, the second order PLL is widely known to be characterized by those two parameters, damping factor or ratio and the natural frequency. In the time domain, this loop filter is equivalent to the output:

$$y_1[k] = y_1[k-1] + (C_1 + C_2)x_1[k] - C_1 x_1[k-1] \tag{3.92}$$

The loop filter parameters obtained as such will be referenced by method 2 in the following section where a performance comparison is carried out between various first order loop filters.

On the other hand, impulse invariance mapping on the loop filter transfer function:

$$F_1(s) \overset{s=\frac{1-z^{-1}}{T_I}}{\longmapsto} F_1(z) \tag{3.93}$$

yields a different digital first order loop filter:

$$\begin{aligned} F_1(z) &= \frac{T_I}{1-z^{-1}} \frac{\tau_2(1-z^{-1}) + T_I}{\tau_1 T_I} \\ &= \frac{A_1 + A_2 z^{-1}}{1-z^{-1}} \end{aligned} \tag{3.94}$$

where

$$A_1 = \frac{\tau_2 + T_I}{\tau_1}; \quad A_2 = -\frac{\tau_2}{\tau_1} \tag{3.95}$$

or in terms of the damping ratio and natural frequency:

$$A_1 = \frac{2\zeta w_n + T_I {w_n}^2}{K_D}; \quad A_2 = -\frac{2\zeta w_n}{K_D} \tag{3.96}$$

The loop filter parameters obtained as such will be referenced by method 3 in the following section where a performance comparison is carried out between various first order loop filters.

3.5.1.3 Second order loop filter

A second order loop filter in the analog domain is defined in [37]:

$$\begin{aligned} F_2(s) &= \left(\frac{{w_n}^3}{s} + a_3 {w_n}^2\right)\frac{1}{s} + b_3 w_n \\ &= \frac{{w_n}^3 + a_3 {w_n}^2 s + b_3 w_n s^2}{s^2} \end{aligned} \tag{3.97}$$

Substituting Equation 3.77 into $F_2(s)$ yields the digital loop filter transfer function:

$$\begin{aligned} F_2(z) &= \frac{{T_I}^2(1+z^{-1})^2}{4(1-z^{-1})^2}\left({w_n}^3 + \frac{2a_3{w_n}^2(1-z^{-1})}{T_I(1+z^{-1})} + \frac{4b_3w_n(1-z^{-1})^2}{{T_I}^2(1+z^{-1})^2}\right) \\ &= \frac{{T_I}^2{w_n}^3(1+z^{-1})^2 + 2a_3T_I{w_n}^2(1-z^{-1})(1+z^{-1}) + 4b_3w_n(1-z^{-1})^2}{4(1-z^{-1})^2} \\ &= \frac{A_1 + A_2z^{-1} + A_3z^{-2}}{1-2z^{-1}+z^{-2}} \end{aligned} \tag{3.98}$$

where $a_3 = 1.1$, $b_3 = 2.4$ and the loop filter coefficients are given by:

$$A_1 = \frac{{T_I}^2{w_n}^3}{4} + \frac{a_3T_I{w_n}^2}{2} + b_3w_n; \quad A_2 = \frac{{T_I}^2{w_n}^3}{2} - 2b_3w_n; \quad A_3 = \frac{{T_I}^2{w_n}^3}{4} - \frac{a_3T_I{w_n}^2}{2} + b_3w_n \tag{3.99}$$

and where the natural frequency is determined by the choice of the noise equivalent bandwidth B_N:

$$w_n = \frac{4B_N(a_3b_3 - 1)}{a_3{b_3}^2 + {a_3}^2 - b_3} \tag{3.100}$$

The expression of this loop filter in the time domain is equivalent to the output:

$$y_2[k] = 2y_2[k-1] - y_2[k-2] + A_1x_2[k] + A_2x_2[k-1] + A_3x_2[k-2] \tag{3.101}$$

3.5.2 Optimum digital loop filters

The optimum loop filter which minimizes both phase output noise power and transient energy has been first considered in [38] in the analog domain, and then revisited in [4] and [5].

3.5.2.1 Design criteria

The main goal is to minimize the following functional:

$$Q_{opt} = \sigma_N^2 + \xi\sum_{k=0}^{\infty}{\delta\phi_0}^2[k] \tag{3.102}$$

where σ_N^2 is the output or estimated phase noise variance due to input noise only and $\delta\phi_0[k]$ is the error between the incoming and estimated phase in the absence of noise at the k-th integration interval. The parameter ξ controls the tradeoff between how much transient energy vs noise power is minimized. Q_{opt} is a functional which means that it is a function which depends on another function. The variance of the output

phase noise component is expressed as a function of the system noise transfer function and the PSD N_i of the input phase noise component which is assumed to be white:

$$\sigma_N^2 = \frac{N_i}{j2\pi} \oint H_N(z) H_N(z^{-1}) z^{-1} dz \tag{3.103}$$

while the transient error energy can be expressed as a function of the system error transfer function in the absence of noise (derived in Section 3.1) and the input phase PSD after applying Parseval's theorem:

$$\begin{aligned} \sum_{k=0}^{\infty} {\delta\phi_0}^2[k] &= \frac{1}{j2\pi} \oint \delta\phi_0(z)\delta\phi_0(z^{-1}) z^{-1} dz \\ &= \frac{1}{j2\pi} \oint H_e(z) H_e(z^{-1}) \phi(z)\phi(z^{-1}) z^{-1} dz \\ &= \frac{1}{j2\pi} \oint (1 - H(z))(1 - H(z^{-1}))\phi(z)\phi(z^{-1}) z^{-1} dz \end{aligned} \tag{3.104}$$

Minimizing Q_{opt} consists in finding a function $H(z)$ (not a variable) among all possible functions which minimizes the expression of Q_{opt}. Since the NCO transfer function is arbitrary and given the expression of $H(z)$ in Equation 3.3, it is intuitive to define a new transfer function $W(z)$ as:

$$W(z) = \frac{H(z)}{N(z)} \tag{3.105}$$

and solve the minimization problem in terms of $W(z)$. The optimum loop filter will then be a function of the optimum solution $W_0(z)$:

$$F(z) = \frac{W_0(z)}{1 - W_0(z)N(z)} \tag{3.106}$$

after substituting Equation 3.105 into Equation 3.3 and solving for $F(z)$. Minimizing Q_{opt} is indeed a problem of variational calculus, and when the functional is an integral, it can be solved by applying the Euler Lagrange theorem. However, the Euler Lagrange theorem is quite complex to be applied in a direct way. A better approach is to use the same steps adopted in the demonstration of the Euler Lagrange theorem to the functional Q_{opt}. These calculus operations are quite complex but they are extremely useful to have an insight on the loop design criterion. The optimum solution for $W(z)$ [38] - [5] is found to be:

$$W_0(z) = \frac{zX(z)}{\psi(z)} \tag{3.107}$$

where $\psi(z)$ and $\psi(z^{-1})$ can be found from:

$$\psi(z)\psi(z^{-1}) = [N_i + \xi\phi(z)\phi(z^{-1})]N(z)N(z^{-1}) \tag{3.108}$$

and

$$X(z) = \left[\frac{\xi N(z^{-1})\phi(z)\phi(z^{-1})}{z\psi(z^{-1})}\right]_+ \tag{3.109}$$

Note that the $[\,]_+$ operator takes the part which poles are inside the unit disc. In the following, the optimum loop filter for a phase step input, a frequency step input and a frequency ramp input will be derived. It is assumed that a single unit delay of loop sampling interval $D = 1$ is present between the computation of the phase estimate and the generation of the local phase signal. This is of utmost importance because a single delay can change the location of the poles and the structure of the optimum loop filter, adding or substracting a pole in the transfer function of $F(z)$. It should be noted that similar derivation has already been shown in [4] and later in [5] but the former only considers a delay $D = 2$, and does not set the NCO transfer function as an arbitrary block that can be customized as it is desired. The latter on the other hand, solves even for a higher or third order phase input known as phase jerk, however, it has to solve for a set of nonlinear equations which consists in initializating several parameters that will determine the poles and zeros of the system transfer function. The search for the optimum loop transfer function can thus be artificial in the sense that all the derivations are computed with respect to these parameters.

It is worth mentioning that in all these derivations, it is assumed that the PD has a unity gain transfer function. However, if the PD is characterized by a non-unity gain $K_D \neq 1$ then the optimum loop filter $F(z)$ has to account for it. The reason for this can be clearly seen after incorporating K_D into the expression of the NCO transfer function $N(z)$ (since $N(z)$ is always multiplied by $D(z)$ in Equation 3.3). It is then possible to conclude that $\psi(z)$ and $\psi(z^{-1})$ are proportional to K_D, while $X(z)$ is independent of it and $W_0(z)$ and $F(z)$ are inversely proportional to it. In conclusion, if the PD gain is $K_D \neq 1$, then the optimum loop filter $F(z)$ has to account for it in terms of an additional gain of $1/K_D$.

3.5.2.2 Phase step input

A phase step input in the time domain is transformed into the z domain as :

$$\phi[n] = ru[n] \rightarrow \phi(z) = \frac{r}{1-z^{-1}} = \frac{-rz}{1-z} \tag{3.110}$$

with a region of convergence $|z| > 1$. Assuming a delay D of 1 unit of loop sampling interval in the locked loop, and considering the phase and rate NCO transfer function $N_{PR}(z)$ as in Equation 3.18, it can be concluded that in this case:

$$N(z) = \frac{T_I}{z-1} \qquad N(z^{-1}) = \frac{-zT_I}{z-1} \tag{3.111}$$

Substituting the phase step input together with the considered NCO transfer function into Equation 3.108, it can be derived that:

$$\begin{aligned} \psi(z)\psi(z^{-1}) &= \left(N_i - \frac{\xi r^2 z}{(1-z)^2}\right) N(z)N(z^{-1}) \\ &= \frac{(z-1)^2 - \alpha z}{(z-1)^2} \cdot \frac{-N_i z {T_I}^2}{(z-1)^2} \\ &= \frac{-N_i {T_I}^2 z(z-z_1)(z-z_2)}{(z-1)^4} \end{aligned} \tag{3.112}$$

where $\alpha = \dfrac{\xi r^2}{N_i}$ and z_1 and z_2 are the zeros of $(z-1)^2 - \alpha z = 0$. These zeros can be derived noticing that the expression will equal to $\psi(z)\psi(z^{-1})$ and so they will be the inverse of each other. Hence the zeros of the second order equation can be derived from

$$z_{1,2} = \frac{\alpha + 2 \pm \sqrt{\alpha^2 + 4\alpha}}{2} \tag{3.113}$$

From Equation 3.112 $\psi(z)$ and $\psi(z^{-1})$ can be derived as,

$$\psi(z) = \frac{\sqrt{N_i} T_I z(z-z_1)}{(z-1)^2}$$

and

$$\begin{aligned} \psi(z^{-1}) &= \frac{-\sqrt{N_i} T_I (z-z_2)}{(z-1)^2} \\ &= \frac{-\sqrt{N_i} T_I (z^{-1})^2 (z - {z_1}^{-1})}{(z^{-1}-1)^2} \\ &= \frac{-\sqrt{N_i} T_I (z^{-1})^2 (z_1 z - 1))}{z_1 (z^{-1}-1)^2} \\ &= \frac{\sqrt{N_i} T_I z^{-1} (z^{-1} - z_1)}{z_1 (z^{-1}-1)^2} \end{aligned} \tag{3.114}$$

However, in order to have an exact symmetric expression of $\psi(z)$ and its inverse, we rewrite

$$\psi(z) = \frac{\sqrt{N_0}T_I z(z-z_1)}{\sqrt{z_1}(z-1)^2} \tag{3.115}$$

$$\psi(z^{-1}) = \frac{\sqrt{N_i}T_I z^{-1}(z^{-1}-z_1)}{\sqrt{z_1}(z^{-1}-1)^2} \tag{3.116}$$

Moreover, substituting $\psi(z^{-1})$ and the expressions of the phase input and NCO transfer function into Equation 3.109, it can be derived that:

$$\begin{aligned} X(z) &= \left[\frac{\xi \frac{zT_I}{z-1}\frac{r^2 z}{(z-1)^2}\sqrt{z_1}(z^{-1}-1)^2}{z\sqrt{N_i}T_I z^{-1}(z^{-1}-z_1)}\right]_+ \\ &= \left[\frac{\xi r^2\sqrt{z_1}}{\sqrt{N_0}}\frac{z}{(z-1)(1-z_1 z)}\right]_+ \\ &= \frac{\xi r^2\sqrt{z_1}}{\sqrt{N_i}}\frac{1}{(z-1)(1-z_1)} \end{aligned} \tag{3.117}$$

Substituting $X(z)$ into Equation 3.107 the expression of the optimum $W_0(z)$ can be derived as:

$$W_0(z) = \frac{\alpha z_1(z-1)}{T_I(1-z_1)(z-z_1)} = \frac{A_1(z-1)}{T_I(z-z_1)} \tag{3.118}$$

where A_1 is the optimum loop filter parameter as will be shown next and is defined as:

$$A_1 = \frac{\alpha z_1}{1-z_1} \tag{3.119}$$

Using this definition and the fact that z_1 is a root of $(z-1)^2 - \alpha z = 0$ yields:

$$z_1 + A_1 = \frac{-{z_1}^2 + \alpha z_1 + z_1}{1-z_1} = 1 \tag{3.120}$$

Finally the optimum loop filter transfer function is a zero order filter and is obtained after substituting $W_0(z)$ into Equation 3.106:

$$\begin{aligned} F_0(z) &= \frac{A_1(z-1)}{T_I\left[(z-z_1)-A_1\right]} \\ &= \frac{A_1(z-1)}{T_I(z-1)} \\ &= \frac{A_1}{T_I} \end{aligned} \tag{3.121}$$

which is equivalent to the output:

$$y_0[k] = \frac{A_1}{T_I}x_0[k] \tag{3.122}$$

in the time domain. Thus, the optimum loop filter $F_0(z)$ consists of a constant which varies as a function of the desired normalized noise equivalent bandwidth as shown in Figure 3.33. To determine the optimum loop filter parameter A_1, the first step is to

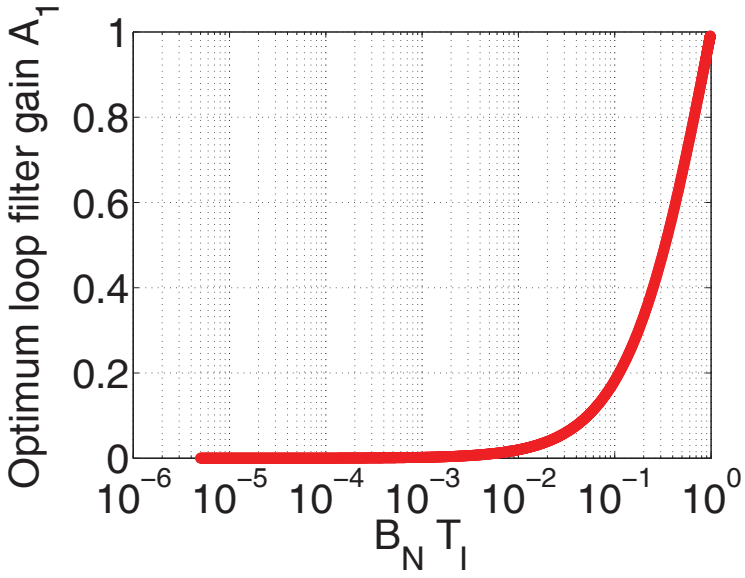


Figure 3.33: The zero order optimum loop filter gain - for a phase step input.

solve for z_1. And this is where the designer specifications come into play, that is ξ and more generally α in the form of the noise equivalent bandwidth B_N. In fact, B_N is a theoretical bandwidth of an ideal box-car unity filter which results in the same output power as that due to the real practical filter $H(z)$, and can be expressed as:

$$2B_N T_I = \frac{1}{j2\pi} \oint H(z)H(z^{-1})z^{-1}dz \tag{3.123}$$

The contour integral can be evaluated using the Cauchy Residue Method where:

$$2B_N T_I = \frac{\sigma_N^2}{N_i} = \sum_i \lim_{z \to z_i} (z - z_i)H(z)H(z^{-1})z^{-1} \tag{3.124}$$

where $H(z)$ is easily determined by recalling Equation 3.105,

$$H(z) = \frac{A_1}{z - z_1} \tag{3.125}$$

and z_i are the poles of $H(z)$ inside the unit circle, in this case z_1. But in order to compute those residues, it is imperative to have an expression of $H(z)$ and $F(z)$, which

means also the optimum loop filter parameters. This is why the standard approach is to scan all possible values of α, derive the corresponding values of z_1 and z_2 and therefore the loop filter parameters and it is only in the last step that the noise equivalent bandwidth is computed. Only after this analysis, it will be possible to pick a desired noise equivalent bandwidth and immediately obtain the loop filter parameter values. This is because there is no closed form solution yielding the loop filter parameters in terms of the desired noise equivalent bandwidth. In fact, A_1 can only be expressed in terms of α which includes ξ as well. For these reasons, it is necessary to approximate the piece-wise linear function as shown in Figure 3.35 which relates the $B_N T_I$ and the α parameter in the log scale. This can be written as:

$$\log_{10}(B_N T_I) = a_i \log_{10}\alpha + b_i \tag{3.126}$$

where a_i and b_i are the slope and intercept values of each piece-wise linear function region denoted by i and which is identified by a range of $B_N T_I$ values. In fact, dividing the aforementioned curve into 10 piece-wise linear functions defined on $B_N T_I$ ranges, yields the slopes and intercepts as listed in Table 3.1. Setting a desired normalized noise equivalent bandwidth $B_N T_I$ and substituting those slope and intercept values in Equation 3.126 yields the parameter α value. Subsequently, substituting this value into Equations 3.113 and 3.119 yields the optimum loop filter parameters, in this case the loop filter gain A_1. Figure 3.34 shows the loop filter gain estimation quality after interpolation or approximation following the aforementioned procedure.
Furthermore, Figure 3.36 shows the location of the closed loop transfer function $H(z)$ pole as a function of the desired normalized noise equivalent bandwidth. Stability is verified given that the pole is inside the unit disc.

3.5.2.3 Frequency step input

A frequency step input translates into a phase ramp input where the time domain and z transform expressions are:

$$\phi[n] = rnu[n] \rightarrow \phi(z) = \frac{rz^{-1}}{(1-z^{-1})^2} = \frac{rz}{(z-1)^2} = \phi(z^{-1}) \tag{3.127}$$

with a region of convergence $|z| > 1$. Assuming a delay D of 1 unit of loop sampling interval in the locked loop, the phase and rate NCO transfer function $N_{PR}(z)$ is

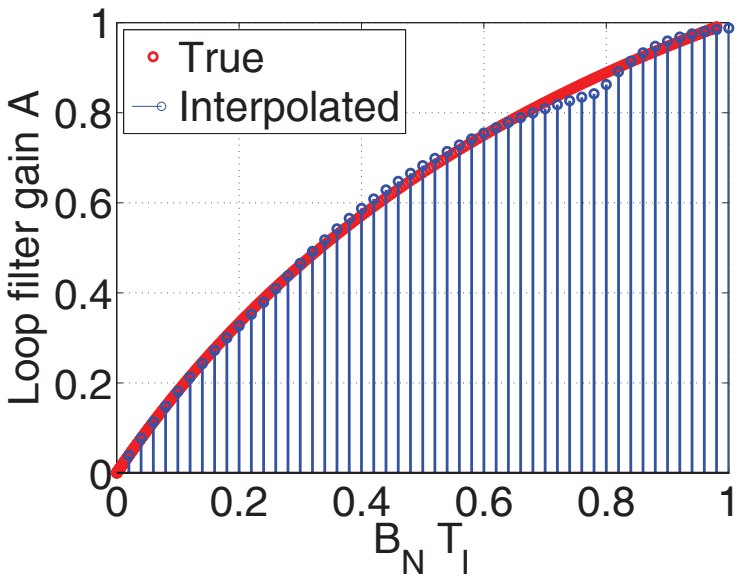


Figure 3.34: Comparison between true and estimated loop filter gain A_1 - after the log function approximation procedure and assuming a phase step input.

$B_N T_I$ **range i**	a_i	b_i
5×10^{-6} to 2.32×10^{-5}	0.5	-0.3010
2.32×10^{-5} to 1.07×10^{-4}	0.5	-0.3011
1.07×10^{-4} to 5×10^{-4}	0.5	-0.3008
5×10^{-4} to 2.32×10^{-3}	0.4999	-0.3019
2.32×10^{-3} to 1.08×10^{-2}	0.5005	-0.2987
1.08×10^{-2} to 4.98×10^{-2}	0.4978	-0.3078
4.98×10^{-2} to 0.23	0.5035	-0.2965
0.23 to 0.79	0.3973	-0.3673
0.79 to 1.02	0.0827	-0.1575

Table 3.1: Slope a_i and intercept b_i of the piece-wise linear approximation of the function relating the loop filter normalized noise equivalent bandwidth to the parameter α corresponding to the optimum loop filter designed according to a phase step input.

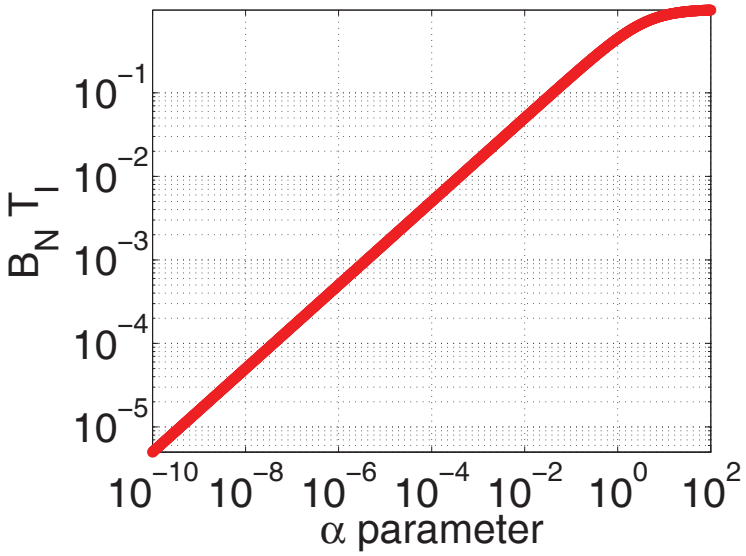


Figure 3.35: The zero order optimum loop filter normalized noise equivalent bandwidth versus the α parameter - for a phase step input.

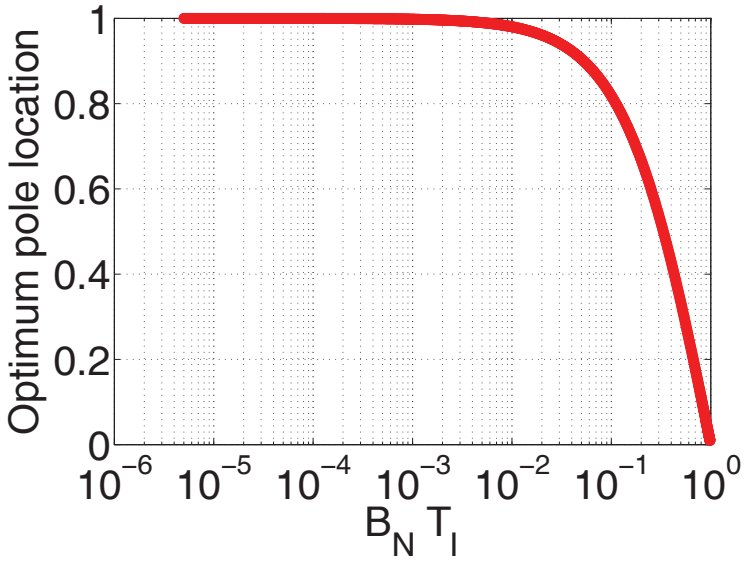


Figure 3.36: Closed loop system pole vs desired normalized noise equivalent bandwidth corresponding to the optimum zero order loop filter - and a phase step input.

considered as in Equation 3.18. Substituting the phase ramp input together with the considered NCO transfer function into Equation 3.108, it can be derived that:

$$\begin{aligned}\psi(z)\psi(z^{-1}) &= \left(N_i + \frac{\xi r^2 z^2}{(1-z)^4}\right) N(z)N(z^{-1}) \\ &= \frac{(z-1)^4 + \alpha z^2}{(z-1)^4} \cdot \frac{-N_i z {T_I}^2}{(z-1)^2} \\ &= \frac{-N_i {T_I}^2 z(z-z_1)(z-z_2)(z-z_3)(z-z_4)}{(z-1)^6} \end{aligned} \tag{3.128}$$

where $\alpha = \dfrac{\xi r^2}{N_i}$ and z_1 and z_2 are the zeros of $(z-1)^4 + \alpha z^2 = 0$ and are inside the unit disc while $z_3 = {z_1}^{-1}$ and $z_4 = {z_2}^{-1}$ are the zeros outside the unit disc. These zeros can be derived noticing that the expression will equal to $\psi(z)\psi(z^{-1})$ and so two zeros will be the inverse of each other. Hence the zeros of the second order equation can be derived from

$$\begin{aligned}(z-1)^2 &= \sqrt{-\alpha} z \\ (z-1)^2 &= j\sqrt{\alpha} z \\ z^2 - (2 + j\sqrt{\alpha})z + 1 &= 0 \\ \Delta &= (2 + j\sqrt{\alpha})^2 - 4 = -\alpha + j4\sqrt{\alpha} \\ z_{1,3} &= 1 + j\frac{\sqrt{\alpha}}{2} \pm \frac{1}{2}\sqrt{-\alpha + j4\sqrt{\alpha}} \end{aligned} \tag{3.129}$$

and

$$\begin{aligned}(z-1)^2 &= -\sqrt{-\alpha} z \\ (z-1)^2 &= -j\sqrt{\alpha} z \\ z^2 - (2 - j\sqrt{\alpha})z + 1 &= 0 \\ \Delta &= (2 - j\sqrt{\alpha})^2 - 4 = -\alpha - j4\sqrt{\alpha} \\ z_{2,4} &= 1 - j\frac{\sqrt{\alpha}}{2} \pm \frac{1}{2}\sqrt{-\alpha - j4\sqrt{\alpha}} \end{aligned} \tag{3.130}$$

From Equation 3.128 $\psi(z)$ and $\psi(z^{-1})$ can be derived as,

$$\psi(z) = \frac{\sqrt{N_i} T_I z(z-z_1)(z-z_2)}{(z-1)^3}$$

and

$$\begin{aligned}\psi(z^{-1}) &= \frac{-\sqrt{N_i}T_I(z-z_3)(z-z_4)}{(z-1)^3} \\ &= \frac{\sqrt{N_i}T_I(z^{-1})^3(z-{z_1}^{-1})(z-{z_2}^{-1})}{(z^{-1}-1)^3} \\ &= \frac{\sqrt{N_i}T_I z^{-1}(1-{z_1}^{-1}z^{-1})(1-{z_2}^{-1}z^{-1})}{(z^{-1}-1)^3} \\ &= \frac{\sqrt{N_i}T_I z^{-1}(z^{-1}-z_1)(z^{-1}-z_2)}{z_1 z_2(z^{-1}-1)^3}\end{aligned} \tag{3.131}$$

However, in order to have an exact symmetric expression of $\psi(z)$ and its inverse, we rewrite

$$\psi(z) = \frac{\sqrt{N_i}T_I z(z-z_1)(z-z_2)}{\sqrt{z_1 z_2}(z-1)^3} \tag{3.132}$$

$$\psi(z^{-1}) = \frac{\sqrt{N_i}T_I z^{-1}(z^{-1}-z_1)(z^{-1}-z_2)}{\sqrt{z_1 z_2}(z^{-1}-1)^3} \tag{3.133}$$

Moreover, substituting $\psi(z^{-1})$ and the expressions of the phase input and NCO transfer function into Equation 3.109, it can be derived that:

$$\begin{aligned}X(z) &= \left[\frac{-\xi\frac{zT_I}{z-1}\frac{r^2z^2}{(z-1)^4}\sqrt{z_1z_2}(z^{-1}-1)^3}{z\sqrt{N_i}T_I z^{-1}(z^{-1}-z_1)(z^{-1}-z_2)}\right]_+ \\ &= \left[\frac{\xi r^2}{\sqrt{N_0}}\frac{z^2\sqrt{z_1z_2}}{(1-z_1z)(1-z_2z)(z-1)^2}\right]_+ \\ &= \frac{\xi r^2\sqrt{z_1z_2}}{\sqrt{N_i}}\left(\frac{1}{(z-1)^2(z_1-1)(z_2-1)} - \frac{(z_1+z_2-2)}{(z-1)(z_1-1)^2(z_2-1)^2}\right)\end{aligned} \tag{3.134}$$

Substituting $X(z)$ into Equation 3.107 the expression of the optimum $W_0(z)$ can be derived as:

$$W_0(z) = \frac{\alpha z_p(z-1)(A_1z+A_2)}{(z_1-1)^2(z_2-1)^2T_I(z_1-z)(z_2-z)} = \frac{(z-1)(A_1z+A_2)}{T_I(z_1-z)(z_2-z)} \tag{3.135}$$

where A_1 and A_2 are the optimum loop filter parameters as will be shown next and are defined as:

$$A_1 = 2 - z_s \qquad A_2 = z_p - 1 \tag{3.136}$$

where $z_p = z_1z_2$, $z_s = z_1 + z_2$ and it can be verified numerically that:

$$\frac{\alpha z_p}{(z_1-1)^2(z_2-1)^2} = 1 \tag{3.137}$$

Finally the optimum loop filter transfer function is a first order filter and is obtained after substituting $W_0(z)$ into Equation 3.106:

$$\begin{aligned} F_1(z) &= \frac{(z-1)(A_1 z + A_2)}{T_I\left[(z_1 - z)(z_2 - z) - (A_1 z + A_2)\right]} \\ &= \frac{(z-1)(A_1 z + A_2)}{T_I(z-1)^2} \\ &= \frac{A_1 z + A_2}{T_I(z-1)} \end{aligned} \tag{3.138}$$

due to

$$\begin{aligned} (z_1 - z)(z_2 - z) - (A_1 z + A_2) &= z_p - z_s z + z^2 - (2z - z_s z + z_p - 1) \\ &= z^2 - 2z + 1 \\ &= (z-1)^2 \end{aligned} \tag{3.139}$$

and where the zero location of the optimum loop filter $F_1(z)$ varies as a function of the desired noise equivalent bandwidth as shown in Figure 3.37. The expression of $F_1(z)$ yields in the time domain:

$$y_1[k] = y_1[k-1] + \frac{A_1}{T_I} x_1[k] + \frac{A_2}{T_I} x_1[k-1] \tag{3.140}$$

The loop filter parameters obtained as such will be referenced by method 4 in the following section where a performance comparison is carried out between various first order loop filters.

As previously shown, to determine the optimum loop filter parameters A_1 and A_2, the first step is to consider a range of values for α, solve for z_1 and z_2 in terms of α and compute the noise equivalent bandwidth B_N. To compute B_N, a contour integral must be evaluated as in Equation 3.124 where z_i are the poles of $H(z)$ inside the unit circle, in this case z_1 and z_2. It is good to recall Equation 3.105 to express the system transfer function in this case as:

$$H(z) = \frac{A_1 z + A_2}{(z_1 - z)(z_2 - z)} \tag{3.141}$$

Figure 3.38 shows the range of values of $B_N T_I$ as a function of the parameter α. From this curve, one can move in the opposite direction, moving from a desired normalized noise equivalent bandwidth to the corresponding α parameter, to the two zeros of $\psi(z)$, that is z_1 and z_2, and finally to the optimum loop filter parameters A_1 and A_2. In fact, dividing the aforementioned curve into 10 piece-wise linear functions defined

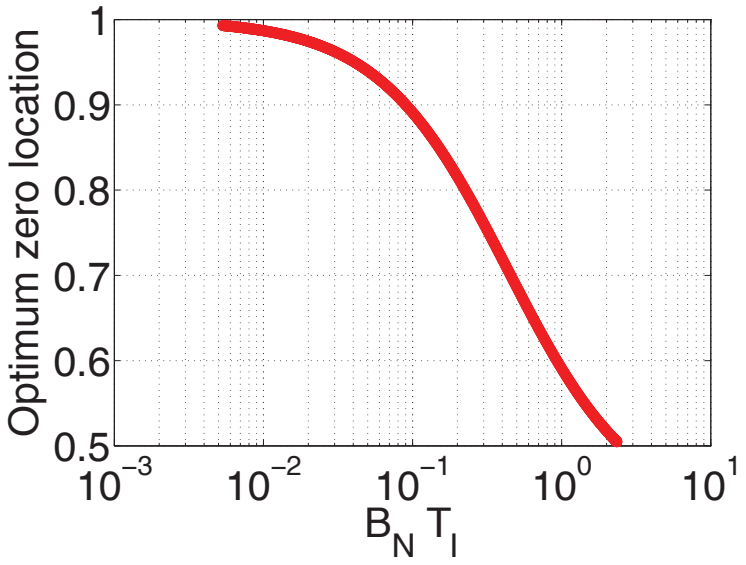


Figure 3.37: The first order optimum loop filter zero location - for a frequency step input.

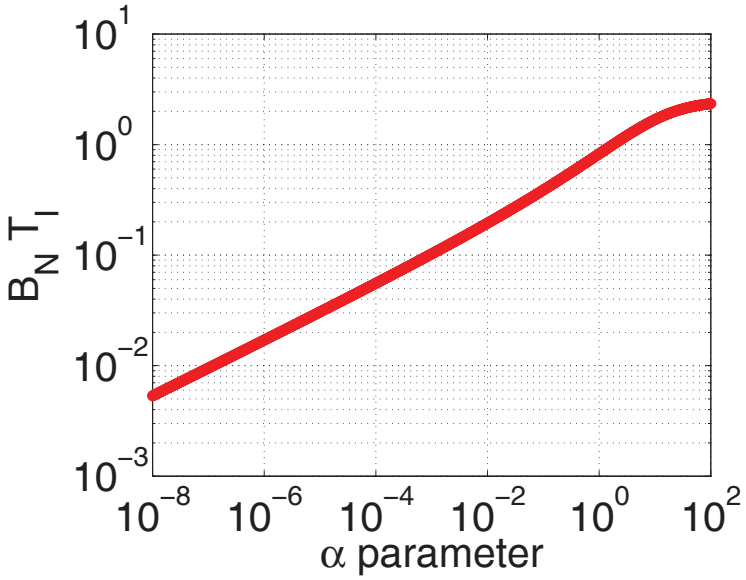


Figure 3.38: The first order optimum loop filter normalized noise equivalent bandwidth versus the α parameter - for a frequency step input.

on α ranges, yields the slopes and intercepts as listed in Table 3.2. Setting a desired normalized noise equivalent bandwidth $B_N T_I$ and substituting those slope and intercept values in Equation 3.126 yields the parameter α value. Subsequently, substituting this value into Equations 3.129 and 3.136 yields the optimum loop filter parameters, in this case the loop filter coefficients A_1 and A_2. Figures 3.39 and 3.40 show the loop filter coefficients estimation quality after interpolation or approximation following the aforementioned procedure.

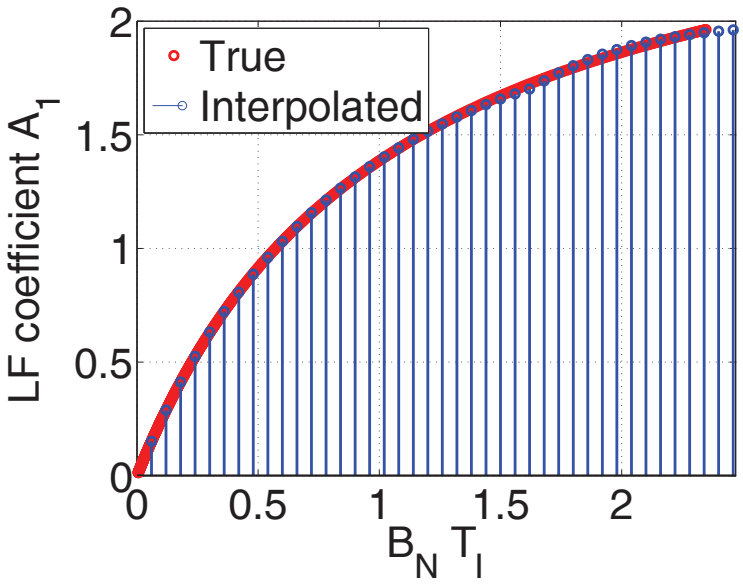


Figure 3.39: Comparison between true and estimated loop filter coefficient A_1 - after the log function approximation procedure and assuming a frequency step input.

Furthermore, Figure 3.41 shows the location of the closed loop transfer function $H(z)$ two poles as a function of the desired normalized noise equivalent bandwidth. Stability is verified given that all poles are inside the unit disc.

3.5.2.4 Frequency ramp input

A frequency ramp input translates into a second order polynomial phase input or phase acceleration where the time domain and z transform expressions are:

$$\phi[n] = \frac{\Delta w}{2} n^2 u[n] \rightarrow \phi(z) = \frac{\Delta w {T_I}^2 z(z+1)}{2(z-1)^3} \tag{3.142}$$

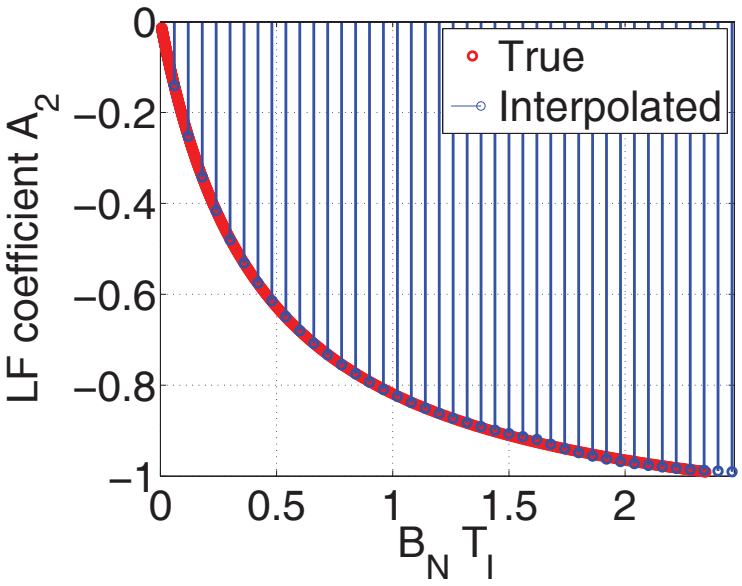


Figure 3.40: Comparison between true and estimated loop filter coefficient A_2 - after the log function approximation procedure and assuming a frequency step input.

$B_N T_I$ **range i**	a_i	b_i
5.32×10^{-3} to 1.01×10^{-2}	0.2516	-0.2606
1.01×10^{-2} to 1.94×10^{-2}	0.2530	-0.2509
1.94×10^{-2} to 3.73×10^{-2}	0.2558	-0.2351
3.73×10^{-2} to 7.26×10^{-2}	0.2609	-0.2112
7.26×10^{-2} to 0.145	0.2708	-0.1759
0.145 to 0.304	0.2887	-0.1322
0.304 to 0.692	0.3215	-0.0884
0.692 to 1.631	0.3352	-0.0853
1.631 to 2.47	0.1623	0.0683

Table 3.2: Slope a_i and intercept b_i of the piece-wise linear approximation of the function relating the loop filter normalized noise equivalent bandwidth to the parameter α corresponding to the optimum loop filter designed according to a frequency step input.

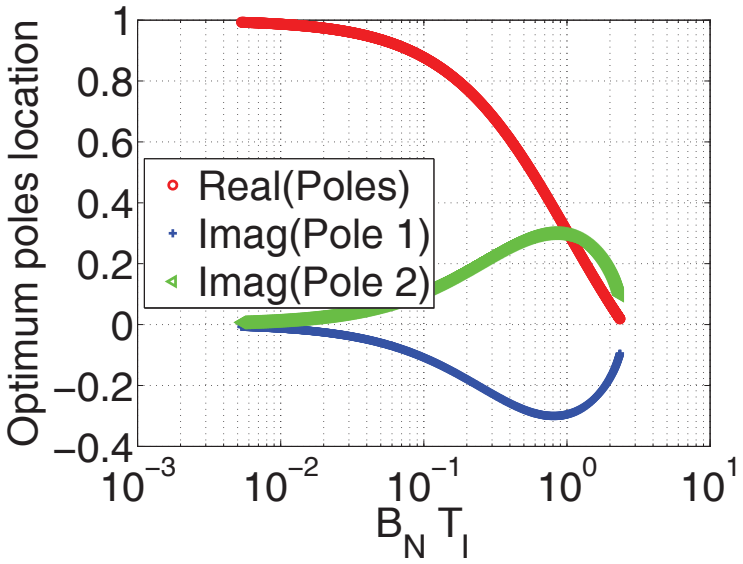


Figure 3.41: Closed loop system poles vs desired normalized noise equivalent bandwidth corresponding to the optimum first order loop filter - and a phase ramp or a frequency step input.

with a region of convergence $|z| > 1$. Similarly,

$$\phi(z^{-1}) = \frac{\frac{\Delta w}{2} {T_I}^2 z^{-1}(z^{-1}+1)}{(z^{-1}-1)^3} = \frac{\frac{\Delta w}{2} {T_I}^2 z^{-1}(z^{-1}+1)(-z^3)}{(z-1)^3} \tag{3.143}$$

Assuming a delay D of 1 unit of loop sampling interval in the locked loop, the phase and rate NCO transfer function $N_{PR}(z)$ is considered as in Equation 3.111. Substituting the frequency ramp input together with the considered NCO transfer function into Equation 3.108, it can be derived that:

$$\begin{aligned} \psi(z)\psi(z^{-1}) &= \left(N_i + \frac{\xi \Delta w^2 {T_I}^4 (z+1)(z^{-1}+1)(-z^3)}{4(z-1)^6}\right) N(z)N(z^{-1}) \\ &= \frac{(z-1)^6 - \alpha z^2 (z+1)^2}{(z-1)^6} \cdot \frac{-N_i z {T_I}^2}{(z-1)^2} \\ &= \frac{-N_i {T_I}^2 z (z-z_1)(z-z_2)(z-z_3)(z-z_4)(z-z_5)(z-z_6)}{(z-1)^8} \end{aligned} \tag{3.144}$$

where $\alpha = \dfrac{\xi \Delta w^2 {T_I}^4}{4N_i}$ and z_1, z_2 and z_3 are the zeros of $(z-1)^6 - \alpha z^2 (z+1)^2 = 0$ and are inside the unit disc while $z_4 = {z_1}^{-1}$ $z_5 = {z_2}^{-1}$ and $z_6 = {z_3}^{-1}$ are the zeros outside the unit disc. These zeros can be derived noticing that the expression will equal to

$\psi(z)\psi(z^{-1})$ and so three zeros will be the inverse of each other. Hence the zeros of the third order equation can be derived from

$$\begin{aligned}(z-1)^6 &= \alpha z^2(z+1)^2 \\ (z-1)^3 &= \sqrt{\alpha}z(z+1) \\ z^3-(3+\sqrt{\alpha})z^2+(3-\sqrt{\alpha})z-1 &= 0\end{aligned} \tag{3.145}$$

where z_1, z_2 and z_3 can be derived using the cubic equation solution first discovered by Scipione dal Ferro (1465-1526) professor of mathematics in the University of Bologna and then Nicolo Fontana Tartaglia of Brescia (1499-1557) and finally first published by Girolamo Cardano (1501-1576), i.e.

$$z_k = \frac{-1}{3a}\left(b + u_k C + \frac{\Delta_0}{u_k C}\right) \tag{3.146}$$

where $a = 1$, $b = -3-\sqrt{\alpha}$, $c = 3-\sqrt{\alpha}$ and $d = -1$ are the third order polynomial coefficients and Δ_0, u_k and C are defined as in [43]. The same can be applied to

$$\begin{aligned}(z-1)^3 &= -\sqrt{\alpha}z(z+1) \\ z^3-(3-\sqrt{\alpha})z^2+(3+\sqrt{\alpha})z-1 &= 0\end{aligned} \tag{3.147}$$

and z_4, z_5 and z_6 can be derived accordingly. From Equation 3.144 $\psi(z)$ and $\psi(z^{-1})$ can be derived as,

$$\psi(z) = \frac{\sqrt{N_i}T_I z(z-z_1)(z-z_2)(z-z_3)}{(z-1)^4}$$

and

$$\begin{aligned}\psi(z^{-1}) &= \frac{-\sqrt{N_i}T_I(z-z_4)(z-z_5)(z-z_6)}{(z-1)^4} \\ &= \frac{-\sqrt{N_i}T_I(z^{-1})^4(z-{z_1}^{-1})(z-{z_2}^{-1})(z-{z_3}^{-1})}{(z^{-1}-1)^4} \\ &= \frac{-\sqrt{N_i}T_I z^{-1}(1-{z_1}^{-1}z^{-1})(1-{z_2}^{-1}z^{-1})(1-{z_3}^{-1}z^{-1})}{(z^{-1}-1)^4} \\ &= \frac{\sqrt{N_i}T_I z^{-1}(z^{-1}-z_1)(z^{-1}-z_2)(z^{-1}-z_3)}{z_1z_2z_3(z^{-1}-1)^4}\end{aligned}$$

However, in order to have an exact symmetric expression of $\psi(z)$ and its inverse, we rewrite

$$\psi(z) = \frac{\sqrt{N_i}T_I z(z-z_1)(z-z_2)(z-z_3)}{\sqrt{z_1z_2z_3}(z-1)^4} \tag{3.148}$$

$$\psi(z^{-1}) = \frac{\sqrt{N_i}T_I z^{-1}(z^{-1}-z_1)(z^{-1}-z_2)(z^{-1}-z_3)}{\sqrt{z_1 z_2 z_3}(z^{-1}-1)^4} \tag{3.149}$$

Moreover, substituting $\psi(z^{-1})$ and the expressions of the phase input and NCO transfer function into Equation 3.109, it can be derived that:

$$\begin{aligned} X(z) &= \left[\frac{-\xi\frac{zT_I}{z-1}\frac{\Delta w^2 {T_I}^4(z+1)(z^{-1}+1)(-z^3)}{4(z-1)^6}\sqrt{z_1 z_2 z_3}(z^{-1}-1)^4}{z\sqrt{N_i}T_I z^{-1}(z^{-1}-z_1)(z^{-1}-z_2)(z^{-1}-z_3)}\right]_+ \\ &= \left[\frac{\xi\Delta w^2 T_I^4}{4\sqrt{N_i}}\frac{z^2(z+1)^2\sqrt{z_1 z_2 z_3}}{(1-z_1 z)(1-z_2 z)(1-z_3 z)(z-1)^3}\right]_+ \\ &= \frac{\xi\Delta w^2 T_I^4\sqrt{z_1 z_2 z_3}}{4\sqrt{N_i}}\left(\frac{\alpha_{11}}{(z-1)}+\frac{\alpha_{12}}{(z-1)^2}+\frac{\alpha_{13}}{(z-1)^3}\right) \end{aligned} \tag{3.150}$$

where the partial fraction decomposition has been applied for the multiple root $z = 1$ inside the unit disc and the constants α_{11}, α_{12} and α_{11} are derived using the generic expression in [44]:

$$\begin{aligned} \alpha_{11} &= \frac{1}{(3-1)!}\left[\frac{d^2}{dz^2}\frac{z^2(z+1)^2}{(1-z_1 z)(1-z_2 z)(1-z_3 z)}\right]_{z=1} \\ \alpha_{12} &= \frac{1}{(3-2)!}\left[\frac{d}{dz}\frac{z^2(z+1)^2}{(1-z_1 z)(1-z_2 z)(1-z_3 z)}\right]_{z=1} \\ \alpha_{13} &= \frac{1}{(3-3)!}\left[\frac{z^2(z+1)^2}{(1-z_1 z)(1-z_2 z)(1-z_3 z)}\right]_{z=1} \end{aligned}$$

More explicitly

$$\begin{aligned} \alpha_{11} &= -\frac{{z_p}^2+2z_1{z_2}^2{z_3}^2+{z_2}^2{z_3}^2+2{z_1}^2z_2{z_3}^2-8z_1z_2{z_3}^2-2z_2{z_3}^2}{(z_1-1)^3(z_2-1)^3(z_3-1)^3} \\ &= -\frac{{z_1}^2{z_3}^2-2z_1{z_3}^2+5{z_3}^2+2{z_1}^2{z_2}^2z_3-8z_1{z_2}^2z_3-2{z_2}^2z_3-8{z_1}^2z_2z_3}{(z_1-1)^3(z_2-1)^3(z_3-1)^3} \\ &= -\frac{16z_1z_2z_3+8z_2z_3-2{z_1}^2z_3+8z_1z_3-14z_3+{z_1}^2{z_2}^2-2z_1{z_2}^2}{(z_1-1)^3(z_2-1)^3(z_3-1)^3} \\ &= -\frac{5{z_2}^2-2{z_1}^2z_2+8z_1z_2-14z_2+5{z_1}^2-14z_1+13}{(z_1-1)^3(z_2-1)^3(z_3-1)^3} \end{aligned} \tag{3.151}$$

$$\alpha_{12} = \frac{4\,(z_d-2z_s+3)}{(z_1-1)^2(z_2-1)^2(z_3-1)^2} \tag{3.152}$$

$$\alpha_{13} = -\frac{4}{(z_1-1)\,(z_2-1)\,(z_3-1)} \tag{3.153}$$

where $z_p = z_1 z_2 z_3$, $z_s = z_1 + z_2 + z_3$ and $z_d = z_1 z_2 + z_1 z_3 + z_2 z_3$. Substituting $X(z)$ into Equation 3.107 the expression of the optimum $W_0(z)$ can be derived as:

$$W_0(z) = \frac{\alpha_1(z-1)^3 + \alpha_2(z-1)^2 + \alpha_3(z-1)}{T_I(z-z_1)(z-z_2)(z-z_3)} \tag{3.154}$$

where α_1, α_2 and α_3 are defined as $\alpha_1 = \alpha z_p \alpha_{11}$, $\alpha_2 = \alpha z_p \alpha_{12}$, $\alpha_3 = \alpha z_p \alpha_{13}$. Moreover, it can be shown that:

$$\frac{4\alpha z_1 z_2 z_3}{(z_1-1)^2(z_2-1)^2(z_3-1)^2} \approx 1 \tag{3.155}$$

Finally the optimum loop filter transfer function is obtained after substituting $W_0(z)$ into Equation 3.106:

$$F_2(z) = \frac{N_3(z)}{D_3(z)} = \frac{\alpha_1(z-1)^3 + \alpha_2(z-1)^2 + \alpha_3(z-1)}{T_I\left[(z-z_1)(z-z_2)(z-z_3) - \alpha_1(z-1)^2 - \alpha_2(z-1) - \alpha_3\right]} \tag{3.156}$$

where the zeros and poles of this optimum loop filter can be obtained by factoring out, the third order polynomials $N_3(z)$ and $D_3(z)$:

$$\begin{aligned} N_3(z) &= \alpha_1(z-1)^3 + \alpha_2(z-1)^2 + \alpha_3(z-1) \\ &= (z-1)(\alpha_1 z^2 + (\alpha_2 - 2\alpha_1)z + \alpha_1 - \alpha_2 + \alpha_3) \\ &= (z-1)(A_1 z^2 + A_2 z + A_3) \\ &= (z-1)(z-z_{z1})(z-z_{z2}) \end{aligned}$$

where

$$A_1 = \alpha_1 \qquad A_2 = \alpha_2 - 2\alpha_1 \qquad A_3 = \alpha_1 - \alpha_2 + \alpha_3 \tag{3.157}$$

and

$$z_{z1,z2} = \frac{2A_1 - A_2 \pm \sqrt{A_2^2 - 4A_1 A_3}}{2A_1} \tag{3.158}$$

Figure 3.42 shows the location of these zeros for a range of possible α values. On the other hand, the denominator $D_3(z)$ determines the poles of $F_2(z)$:

$$\begin{aligned} D_3(z) &= (z-z_1)(z-z_2)(z-z_3) - \alpha_1(z-1)^2 - \alpha_2(z-1) - \alpha_3 \\ &= z^3 - (\alpha_1 + z_s)z^2 + (z_d + 2\alpha_1 - \alpha_2)z - \alpha_1 + \alpha_2 - \alpha_3 - z_p \\ &= (z-z_{p1})(z-z_{p2})(z-z_{p3}) \end{aligned}$$

where z_{p1}, z_{p2} and z_{p3} can be obtained using Equation 3.146 yielding the third order polynomial roots, with polynomial coefficients $a = 1$, $b = -\alpha_1 - z_s$, $c = z_d + 2\alpha_1 - \alpha_2$

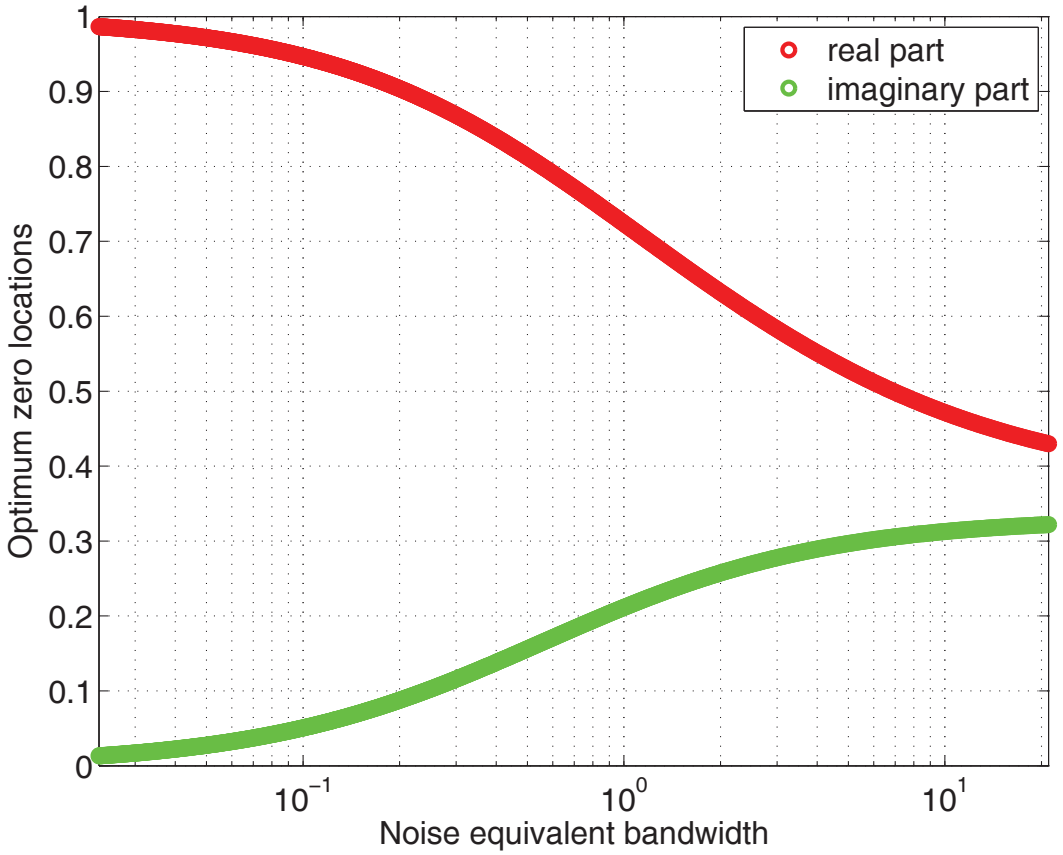


Figure 3.42: Second order filter zeros location of the optimum loop filter - for a frequency ramp input.

and $d = -\alpha_1 + \alpha_2 - \alpha_3 - z_p$. However, by noticing that for all the considered values of α, these coefficients reduce to $a = 1, b = -3, c = 3, d = -1$, it is concluded that $D_3(z)$ has a triple root at $z = 1$ and thus the optimum loop filter is a second order filter $F_2(z)$:

$$F_2(z) = \frac{(z - z_{z1})(z - z_{z2})}{T_I(z-1)^2} = \frac{A_1 z^2 + A_2 z + A_3}{T_I(z^2 - 2z + 1)} \tag{3.159}$$

where A_1, A_2, and A_3 are the optimum loop filter parameters. The expression of this loop filter in the time domain is equivalent to the output:

$$y_2[k] = 2y_2[k-1] - y_2[k-2] + \frac{A_1}{T_I}x_2[k] + \frac{A_2}{T_I}x_2[k-1] + \frac{A_3}{T_I}x_2[k-2] \tag{3.160}$$

As previously shown, to determine the optimum loop filter parameters A_1, A_2 and A_3, the first step is to consider a range of values for α, solve for z_1, z_2, and z_3 in terms of α and compute the noise equivalent bandwidth B_N. To compute B_N, a contour integral must be evaluated as in Equation 3.124 where z_i are the poles of $H(z)$ inside the unit circle, in this case z_1, z_2, and z_3. It is good to recall Equation 3.105 to express the system transfer function in this case as:

$$H(z) = \frac{(z - z_{z1})(z - z_{z2})}{(z - z_1)(z - z_2)(z - z_3)} = \frac{A_1 z^2 + A_2 z + A_3}{(z - z_1)(z - z_2)(z - z_3)} \tag{3.161}$$

This yields Figure 3.43. From this plot, one can move in the opposite direction, moving from a desired noise equivalent bandwidth to the corresponding α parameter, to the three zeros of $\psi(z)$, that is z_1, z_2 and z_3, to the intermediate parameters α_1, α_2 and α_3 and then finally to the optimum loop filter parameters A_1, A_2 and A_3.

In fact, dividing the aforementioned curve into 10 piece-wise linear functions defined on α ranges, yields the slopes and intercepts as listed in Table 3.3. Setting a desired normalized noise equivalent bandwidth $B_N T_I$ and substituting those slope and intercept values in Equation 3.126 yields the parameter α value. Subsequently, substituting this value into Equations 3.146 and 3.157 yields the optimum loop filter parameters, in this case the loop filter coefficients A_1 A_2 and A_3. Figures 3.44 3.45 and 3.46 show the loop filter coefficients estimation quality after interpolation or approximation following the aforementioned procedure.

Furthermore, Figure 3.47 shows the location of the closed loop transfer function $H(z)$ three poles as a function of the desired normalized noise equivalent bandwidth. Stability is verified given that all poles are inside the unit disc.

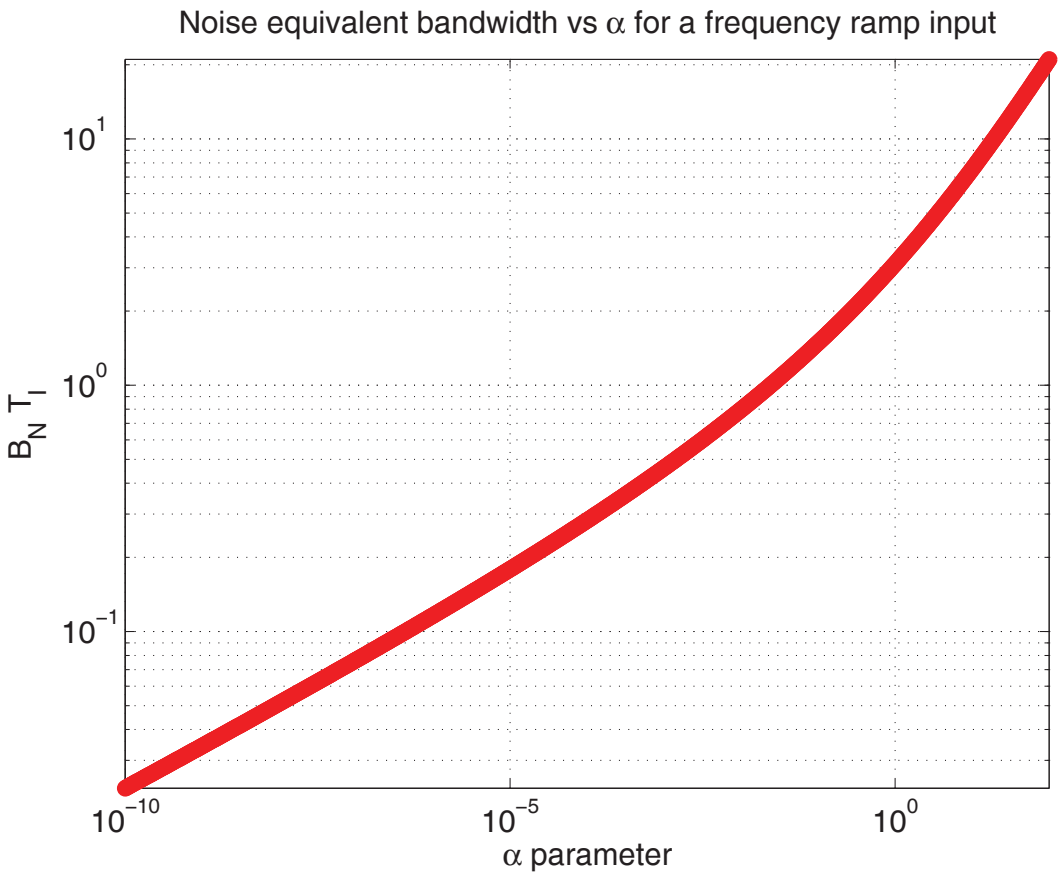


Figure 3.43: Second order filter noise equivalent bandwidth versus the α parameter - for a frequency ramp input.

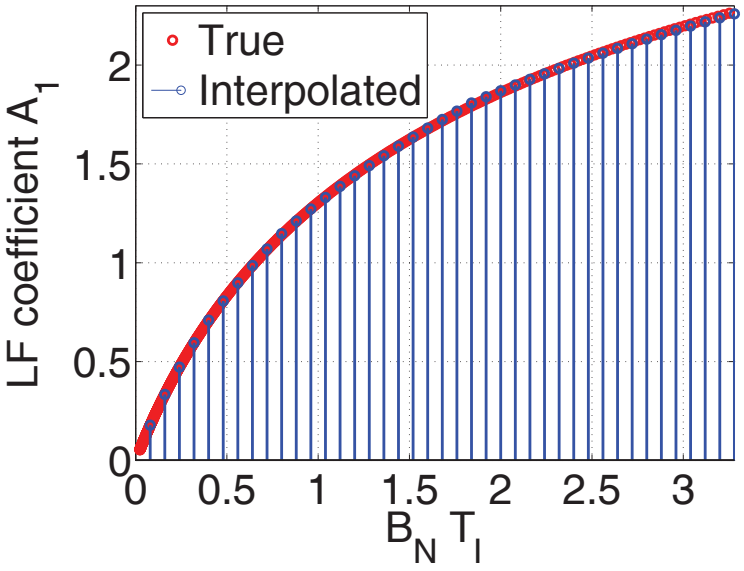


Figure 3.44: Comparison between true and estimated loop filter coefficient A_1 - after the log function approximation procedure and assuming a frequency ramp input.

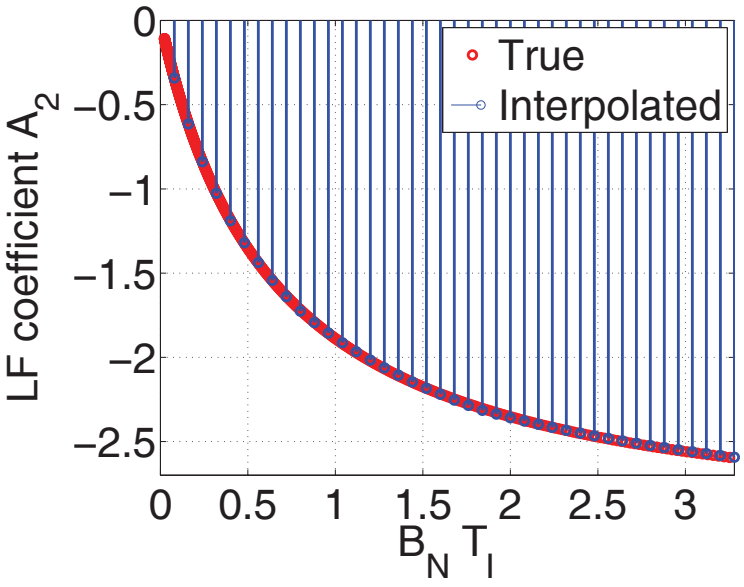


Figure 3.45: Comparison between true and estimated loop filter coefficient A_2 - after the log function approximation procedure and assuming a frequency ramp input.

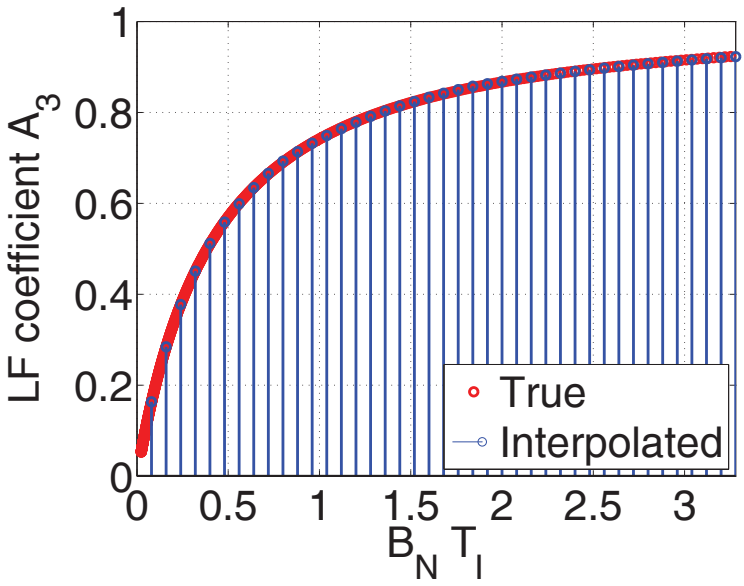


Figure 3.46: Comparison between true and estimated loop filter coefficient A_3 - after the log function approximation procedure and assuming a frequency ramp input.

$B_N T_I$ **range i**	a_i	b_i
2.31×10^{-2} to 3.91×10^{-2}	0.1713	0.0771
3.91×10^{-2} to 6.68×10^{-2}	0.1745	0.1042
6.68×10^{-2} to 0.12	0.1798	0.1431
0.12 to 0.21	0.1888	0.1971
0.21 to 0.39	0.2042	0.2689
0.39 to 0.79	0.2308	0.3578
0.79 to 1.84	0.2763	0.4488
1.84 to 5.4	0.3509	0.4985
20.67 to 5.4	0.4376	0.4407

Table 3.3: Slope a_i and intercept b_i of the piece-wise linear approximation of the function relating the loop filter normalized noise equivalent bandwidth to the parameter α corresponding to the optimum loop filter designed according to a frequency ramp input.

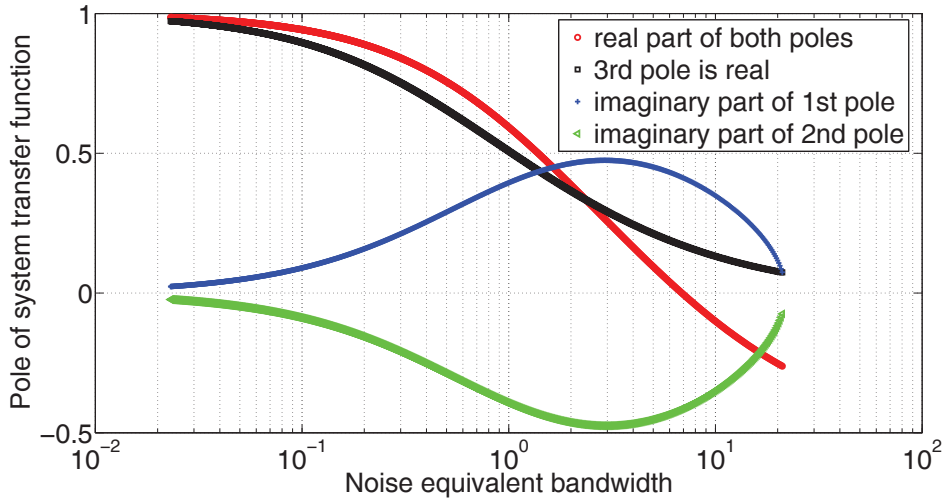


Figure 3.47: Closed loop system poles vs desired normalized noise equivalent bandwidth corresponding to the optimum second order loop - and a frequency ramp input.

3.5.3 Performance comparison

To assess the theoretical performance in terms of stability of such an optimized loop filter with respect to other loop filters, Bode plots are plotted together with root locus plots in z domain and a simple 2D plot of root values versus loop gain. Gain and phase margins are visited to have an understanding of the system degree of stability for a range of frequencies. To verify these expectations, simulated Galileo signals as described in Section 2.3.4 are also employed where the carrier signal is chosen to either be a phase ramp signal (frequency step) or a frequency ramp signal and the spreading code rate to be affected by a Doppler frequency step or a Doppler frequency ramp. It is useful nonetheless, to analyze the carrier and code tracking loops separately at first, and then move to the combined code and carrier tracking loop. Phase detector outputs, loop filter outputs, code rate and Doppler frequency estimates together with CNR and PLI metrics are plotted all along. As mentioned earlier, the various loop filters that are tested here are referenced by the number of the method:

- 1 for direct bilinear transformation of the optimum analog loop filter
- 2 for bilinear transformation of the system transfer function

- 3 for impulse invariance transformation of the system transfer function
- 4 for the input signal corresponding optimum loop filter

3.5.3.1 Root locus

The stability of the closed loop system transfer function is analyzed using root locus plots. In the following, it is assumed a linear model of the discriminator, yielding a unity transfer function $D(z) = 1$ although this condition is not always true especially for very low input signal CNR and high frequency or phase error between the incoming and reference signals. Depending on the situation, the gain of the loop is either decreased or increased, in which case a root locus plot becomes a perfect tool to show the trajectory of the closed loop poles with the loop gain K_D varying from 0 to infinity. Using Equation 3.1, the open loop transfer function can thus be rewritten as:

$$G(z) = F(z)N(z) \tag{3.162}$$

where $N(z)$ is substituted by the phase and rate NCO transfer function $N_{PR}(z)$ from Equation 3.18 with a delay of $D = 1$, such that $N(z) = T_I/(z-1)$ and F(z) by the different analytical expressions of the first order loop filter $F_1(z)$ designed according to methods 1, 2, 3, and 4. It is thus possible to make use of the built-in Matlab function rlocus to plot the root locus plots considering a range of loop gain K_D values varying from 0 to infinity. The first order F(z) yields a second order closed loop system and the resulting two poles set of locations or "root locus" is shown in Figures 3.48-3.51.

The root locus is plotted on the same figure for a set of normalized noise equivalent bandwidth $B_L = B_N T_I$ values together with an informative label showing the loop gain threshold beyond which an underdamped system becomes an overdamped system. These loop gain thresholds are further listed in Table 3.4 where it can be concluded that method 2 and 4 offer the highest gain thresholds while method 3 has the worst performance, unable to deal with $B_L > 0.4$ if not with a loop gain reduction. The superiority of methods 2 and 4 is further supported by looking at the percentage decrease of the overdamped loop gain threshold moving from the lowest to the highest B_L.

Furthermore, Figure 3.52 is plotted to have a grasp of the direction of these roots

Loop filter	B_L **= 0.16**	**0.2**	**0.4**	**0.48**	**0.48**	**0.5**	**% decrease**
method 1	1.63	1.56	1.25	1.15	1.15	1.12	31
method 2	1.68	1.62	1.43	1.37	1.37	1.36	19
method 3	1.36	1.25	0.85	0.74	0.74	0.72	47
method 4	1.69	1.63	1.44	1.39	1.39	1.37	19

Table 3.4: Loop gain thresholds beyond which the loop response switches to become an overdamped system for a set of normalized noise equivalent bandwidths.

with the increase of the loop gain. It can be seen that for low values of K_D, the two poles of the closed loop system are complex conjugate pairs (underdamped loop response) and then become two unequal real roots, one pole that converges to an open loop zero inside the unit disc and the other pole drives the system unstable by moving outside the unit disc.

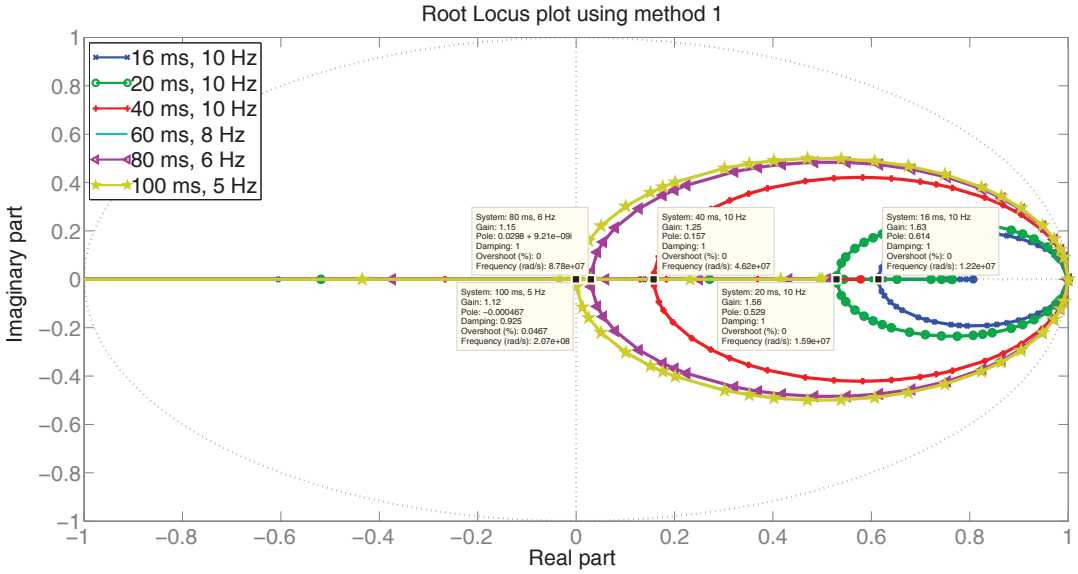


Figure 3.48: Root locus using method 1 for a range of B_L values - for various normalized noise equivalent bandwidth values showing the loop gain threshold beyond which the system switches to an overdamped system

Increasing B_L has a different impact on each of the closed loop systems derived using

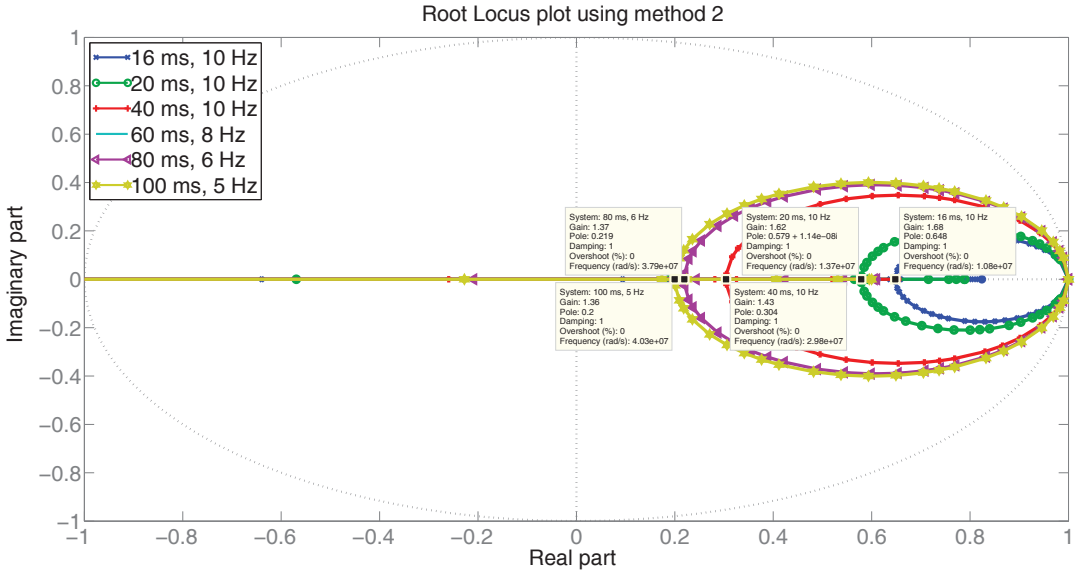


Figure 3.49: Root locus using method 2 for a range of B_L values - for various normalized noise equivalent bandwidth values showing the loop gain threshold beyond which the system switches to an overdamped system

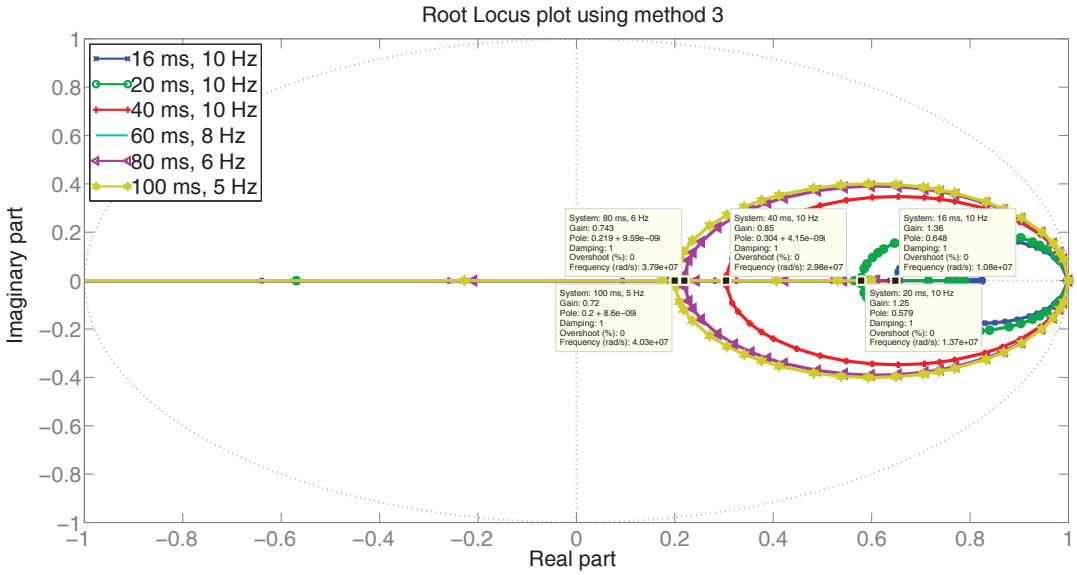


Figure 3.50: Root locus using method 3 for a range of B_L values - for various normalized noise equivalent bandwidth values showing the loop gain threshold beyond which system switches to an overdamped system

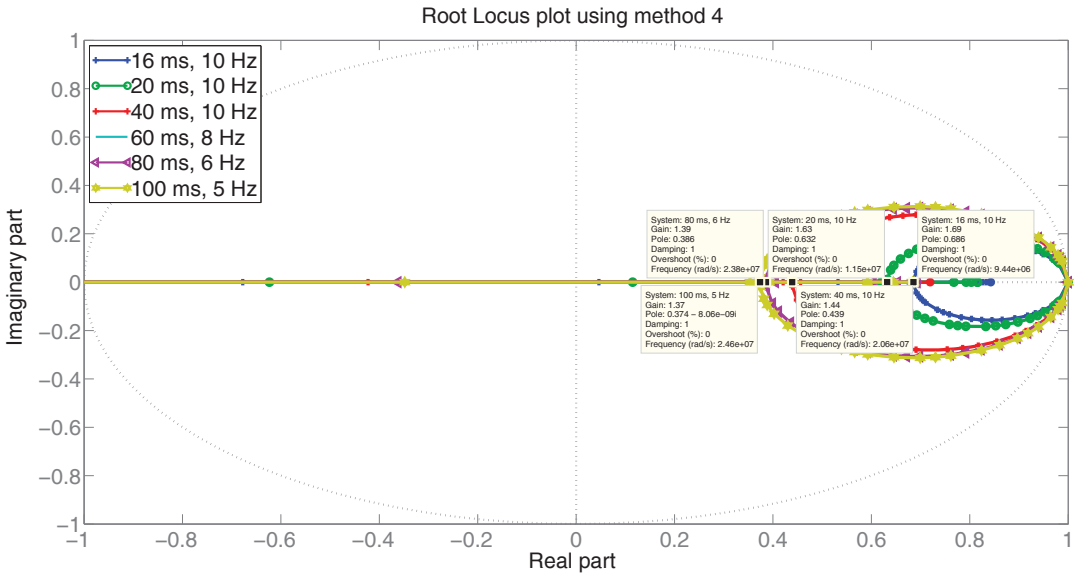


Figure 3.51: Root locus using method 4 for a range of B_L values - for various normalized noise equivalent bandwidth values showing the loop gain threshold beyond which system switches to an overdamped system

the four different methods. After examining Figures 3.48-3.51, the 1st order loop filter designed with method 1 yields roots which move toward the imaginary axis to become two pure complex conjugate poles that will ultimately never drive the transient to zero. The system designed with method 3 on the other hand, although sustaining a lower impact in that sense (the ellipses do not get much near to the imaginary axis), is noted by the change of nature of the designed loop response. In fact, the system designed with method 3 is characterized by an overdamped response for $B_L > 0.4$. The systems designed with method 2 and 4 are less vulnerable to the increase of B_L as they preserve the desired underdamped nature of the loop response for gain values as high as 1.36 and 1.37 respectively. However, method 4 yields a more stable system as its roots are further away from the imaginary axis (ellipses expand less as B_L is increased) as compared to method 2. This is shown better in the following where the poles location for a unity loop gain is plotted in the unit disc.

It is worth noticing that the closed loop system is conditionally stable as the closed loop poles move outside the unit disc after a certain loop gain threshold is reached. This threshold varies according to the designed loop filter methods as well as the normalized

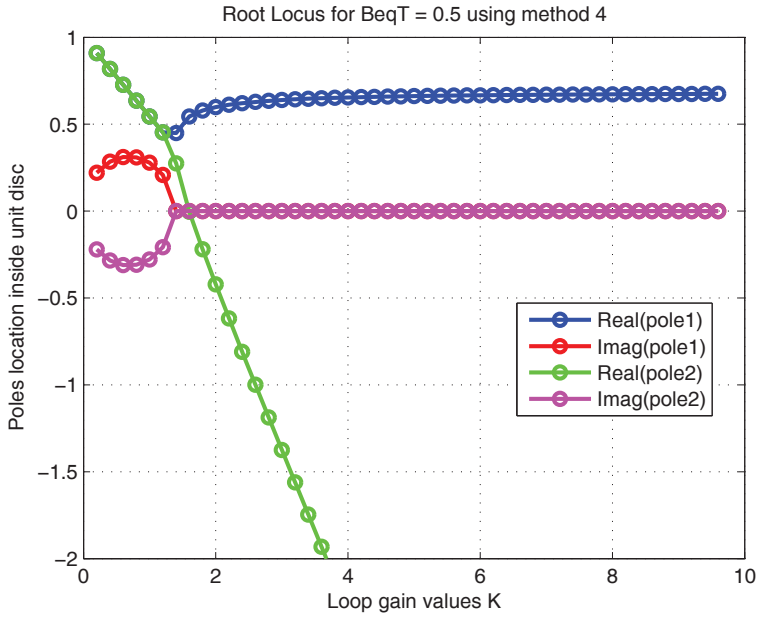


Figure 3.52: System poles as a function of loop gain - using method 4 and a high normalized noise equivalent bandwidth of 0.5 characterized by a 100 ms integration interval.

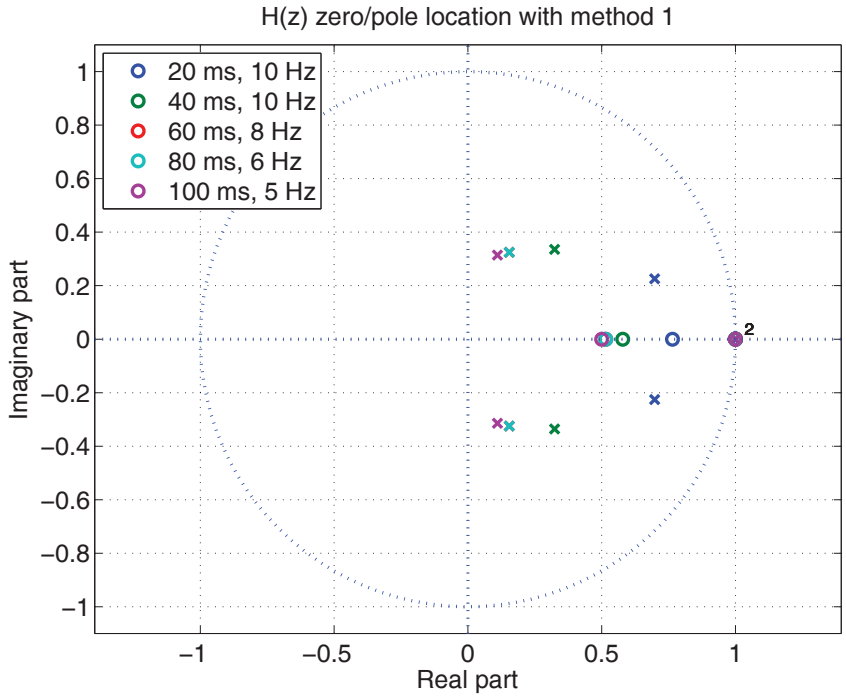
Loop filter	B_L **= 0.16**	**0.2**	**0.4**	**0.48**	**0.48**	**0.5**	**% decrease**
method 1	4.7	3.8	1.8	1.6	1.6	1.5	68
method 2	5.2	4.3	2.5	2.2	2.2	2.1	59
method 3	4.2	3.3	1.5	1.2	1.2	1.15	72
method 4	5.9	4.9	3	2.7	2.7	2.6	56

Table 3.5: Loop gain thresholds in terms of loop instability for a set of normalized noise equivalent bandwidths.

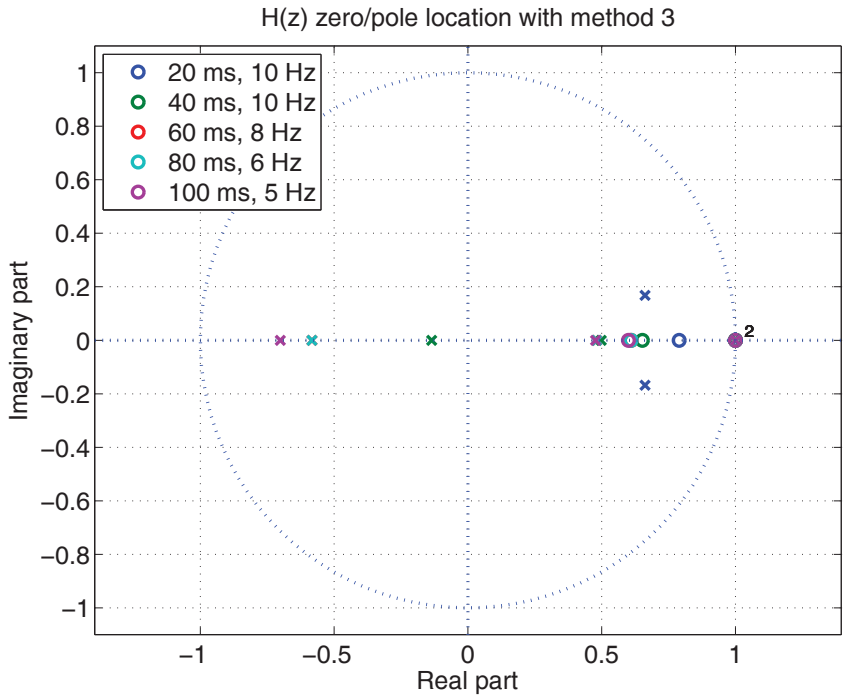
noise equivalent bandwidth B_L as shown in Table 3.5. Again method 3 results in the worst loop stability performance while methods 2 and 4 show a fairly high loop stability gain margin. The superiority of methods 2 and 4 is further supported by looking at the percentage decrease of the corresponding loop gain threshold moving from the lowest to the highest B_L.

Moving the analysis of the closed loop system from a range of loop gain values to a single unity loop gain, it is instructive as well to examine the zeros (depicted by o markers) and poles (depicted by x markers) of the closed loop system transfer function $H(z)$ for different B_L values as shown in Figures 3.53 and 3.54.

Figure 3.53 (b) shows that as B_L varies, the path of the poles of the closed loop system transfer function corresponding to $F_1(z)$ designed according to method 3, changes the system from being underdamped (poles are complex conjugates and the transient is oscillatory) to overdamped (poles are real and no oscillations in transient response). This was expected as was seen previously in Table 3.4 where the overdamped loop gain threshold was less than 1 for $B_L > 0.4$. Moreover, the path of the corresponding poles with the design method 1 as shown in Figure 3.53 (a), move the poles towards a critically damped system where the poles go closer and closer towards a 0 damping ratio (two pure imaginary complex conjugate poles and a transient that never dies out). This happens after a certain threshold value of the normalized noise equivalent bandwidth B_L is crossed due to the fact that beyond such a threshold, continuous to discrete time approximations do not hold and yield a higher B_N as discussed in Section 3.2. On the other hand, the paths of the corresponding poles due to the design of $F_1(z)$ according to method 2 and 4 in Figure 3.54 (a) and (b), demonstrate that the system is under-

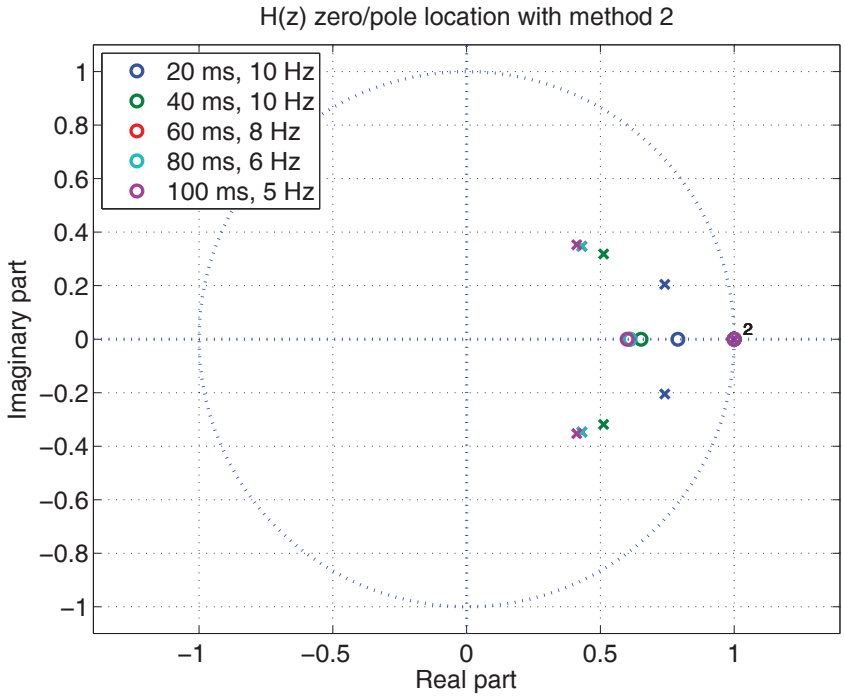


(a)

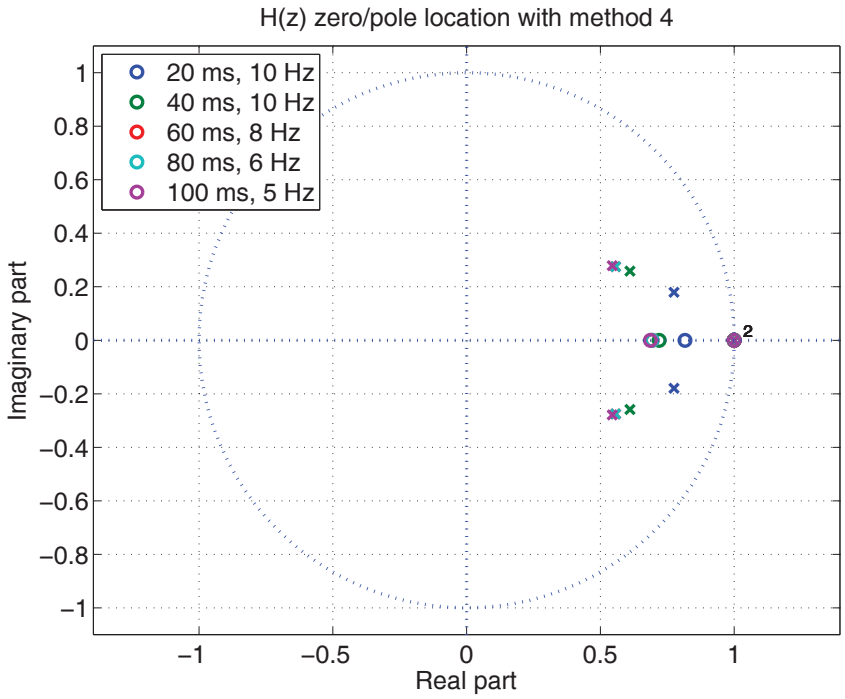


(b)

Figure 3.53: Methods 1 and 3 closed loop system zero pole location for a range of B_L values - using a 1st order loop filter where x markers depict poles and o markers depict zeros.



(a)



(b)

Figure 3.54: Methods 2 and 4 closed loop system zero pole location for a range of B_L values - using a 1st order loop filter where x markers depict poles and o markers depict zeros.

damped for all of the considered normalized noise equivalent bandwidth B_L values for a loop gain of unity.

3.5.3.2 Bode plots

The steady state frequency response of the closed loop system transfer function is analyzed using Bode plots. It is useful to look at the Bode plots to gain a deeper insight into the degree of stability which is characterized by the gain and phase margins of the system [45]. A minimum 6 dB gain margin and somewhere between 30 and 60 degrees phase margin is suggested for the design of systems in general. Figures 3.55 - 3.58 show the frequency response in terms of magnitude and phase of the system transfer function designed with the different four methods presented earlier.

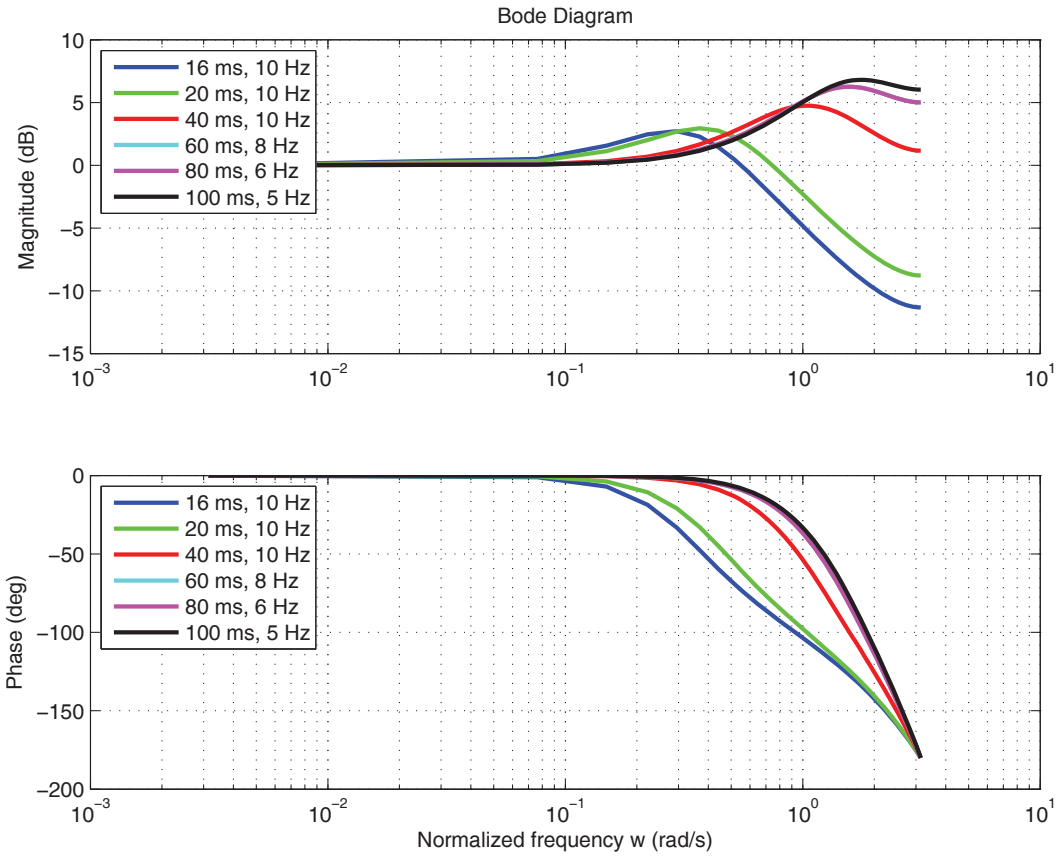


Figure 3.55: Bode plot as a function of B_L for method 1 - showing low pass filter becoming a high pass filter.

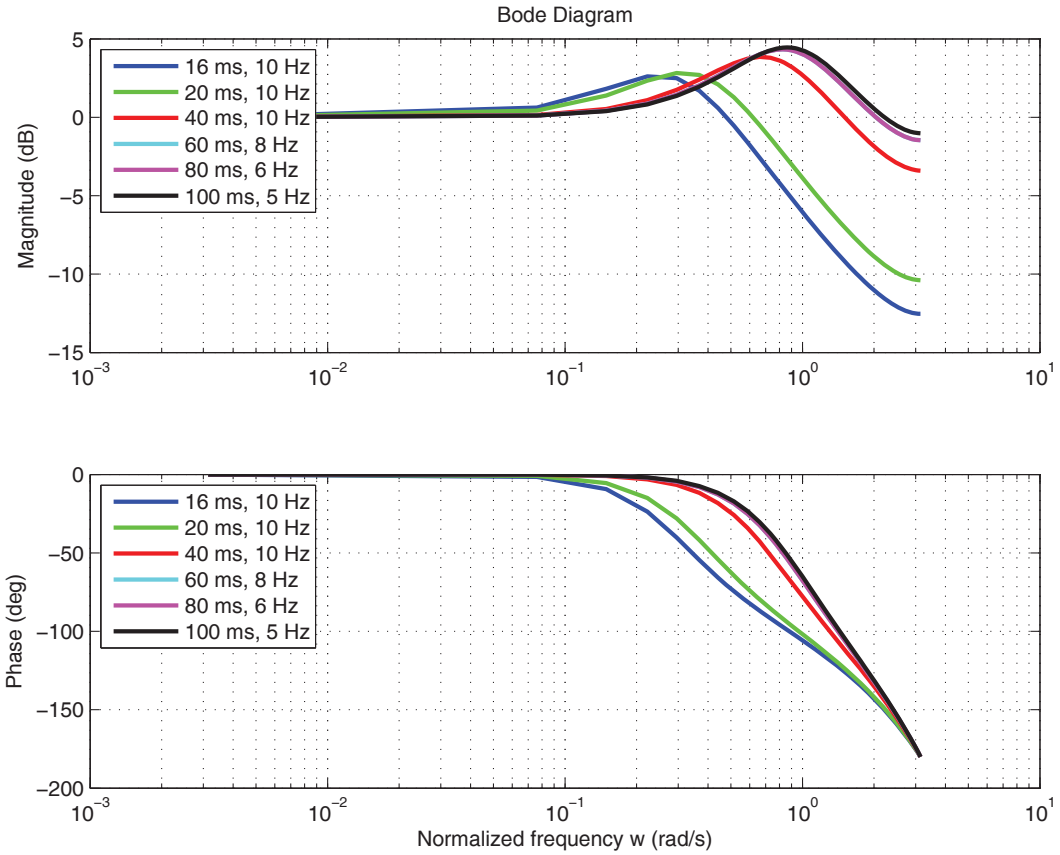


Figure 3.56: Bode plot as a function of B_L for method 2 - showing low pass filter moving its cutoff frequency to higher frequencies.

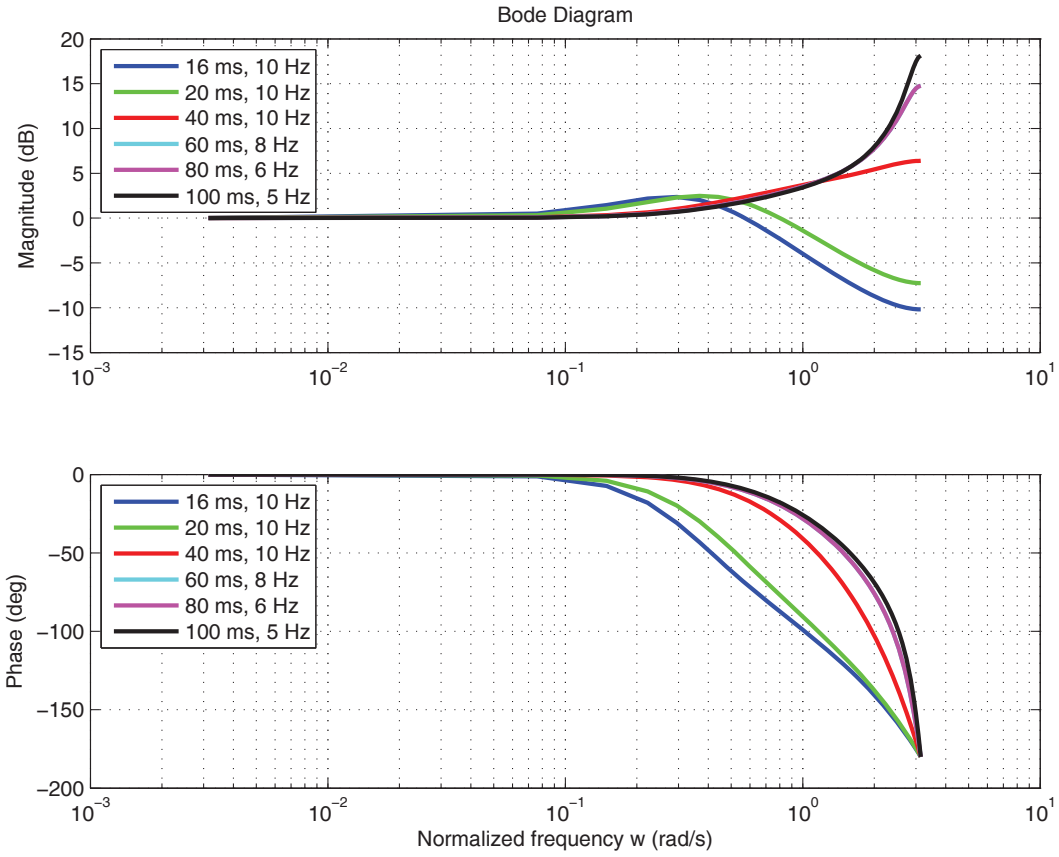


Figure 3.57: Bode plot as a function of B_L for method 3 - showing low pass filter becoming a high pass filter.

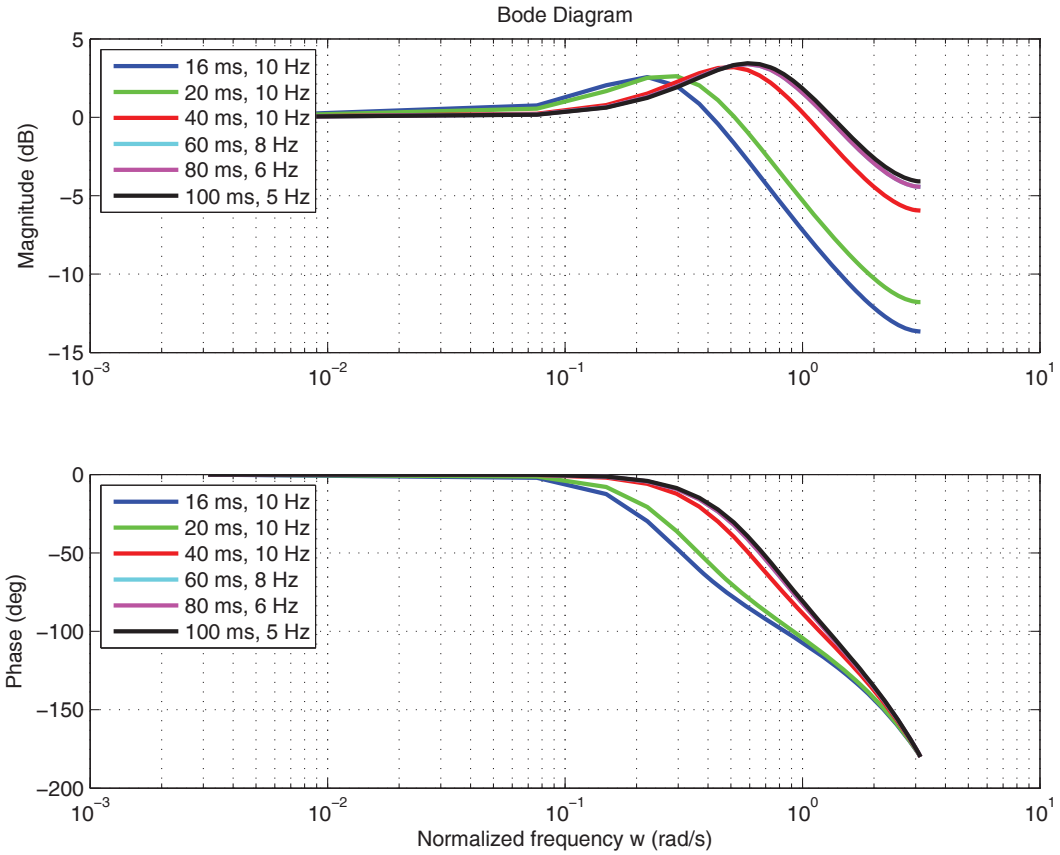


Figure 3.58: Bode plot as a function of B_L for method 4 - showing low pass filter moving its cutoff frequency to higher frequencies but relatively lower than method 2.

Loop filter	B_L **= 0.16**	**0.2**	**0.4**	**0.48**	**0.48**	**0.5**	**% decrease**
method 1	3.68	2.75	0.87	0.56	0.56	0.50	86.41
dB	11.33	8.78	-1.17	-5.01	-5.01	-6.04	
method 2	4.23	3.31	1.48	1.18	1.18	1.12	73.52
dB	12.53	10.39	3.40	1.46	1.46	1.02	
method 3	3.23	2.31	0.48	0.18	0.18	0.12	96.28
dB	10.19	7.26	-6.39	-14.76	-14.76	-18.12	
method 4	4.82	3.88	1.98	1.66	1.66	1.60	66.8
dB	13.65	11.79	5.95	4.43	4.43	4.09	

Table 3.6: Loop gain margins for a set of normalized noise equivalent bandwidths.

Loop filter	B_L **= 0.16**	**0.2**	**0.4**	**0.48**	**0.48**	**0.5**	**% decrease**
method 1	107.20	99.6458	NA	NA	NA	NA	NA
method 2	109.69	104.23	68.36	46.15	46.15	38.73	64.69
method 3	111.01	103.01	NA	NA	NA	NA	NA
method 4	112.48	108.51	88.51	79.51	79.51	77.08	31.47

Table 3.7: Loop phase margins for a set of normalized noise equivalent bandwidths.

It is worth noting that for method 1 and 3, the low pass filter characteristics change into all pass filter characteristics for $B_L > 0.4$ with a higher magnitude for high frequencies. The low pass filter characteristics remain for methods 2 and 4, however, the cutoff frequency is pronouncedly increased especially for method 2, where the high frequencies are only affected by a maximum of -1 dB vs -4 dB for method 4 which offers only a small margin of attenuation. These plots in turn help in identifying the gain and phase margins of the system, i.e. the maximum loop gain and phase fluctuation the system is able to sustain before going to instability. The margins are listed in Tables 3.6 and 3.7 where the gain margin is also expressed in dB and the corresponding percentage decrease in terms of gain/phase margin is computed between the minimum and maximum considered B_L values. It is worth noting that these gain margin values perfectly agree with the loop gain threshold values listed in Table 3.4 with a difference of 1 between the two sets of values indicating the multiplicative gain margin.

Moreover, looking at the gain margin values in dB, it can be concluded that method 4 performs considerably better than method 2 and guarantees the 6 dB gain margin

mentioned earlier for $B_L \leq 0.4$ while at least 4 dB gain margin is guaranteed for $B_L \leq 0.5$. On the other hand, method 2 guarantees the 6 dB gain margin for $B_L = 0.16 - 0.2$ and only 1 dB gain margin for $B_L = 0.5$. Similarly, methods 1 and 3 satisfy the 6 dB gain margin requirement for $B_L = 0.16 - 0.2$ and suffer from a negative gain margin in dB for higher values. Looking at the phase margin values, it is not surprising to note that they are not existent for methods 1 and 3 and $B_L \geq 0.4$ since the corresponding Bode plots show that the system becomes all pass filter for these cases with a gain greater than 0 dB. The phase margins for the remaining cases show that the 30 to 60 degree phase margin requirement is satisfied. It is expected that methods 1 and 3 will not be able to provide stable tracking for $B_L \geq 0.4$ from the considered values while methods 2 and 4 promise relatively good tracking stability.

3.5.3.3 Carrier tracking

Following the scheme of signal simulation described in Section 2.3.4 and after skipping the block implementing the Galileo tiered code signal generation, a pure carrier signal with no Galileo code is generated over 5 seconds with a CNR of 30 dB-Hz and either:

- a constant Doppler frequency of 2 kHz,
- a ramp Doppler frequency of -0.8 Hz/s rate and an initial value of 2 kHz.

Subsequently, the simulated signal is fed into the carrier tracking loop composed of an ATAN2 PD, one of the aforementioned first order loop filters and a frequency NCO which is equivalent to a phase and rate NCO. No loop gain compensation is implemented. An initial frequency error of 3 Hz is set for the range of considered B_L values.

Figures 3.59 to 3.64 look into the PD outputs, phase lock indicators and loop filter carrier frequency estimates showing that for the loop normalized noise equivalent bandwidth B_L values less than 0.2, all four loop filters exhibit similar phase and frequency tracking performance. On the other hand, higher B_L values yield false or loss of tracking for methods 1 and 3, while method 2 and 4 show stable tracking for a B_L as high as 0.5. The results agree with the theoretical analysis performed earlier where the Bode plots have shown that for methods 1 and 3, the closed loop system becomes an all pass filter for $B_L > 0.2$ while the low pass filter characteristics remain for methods

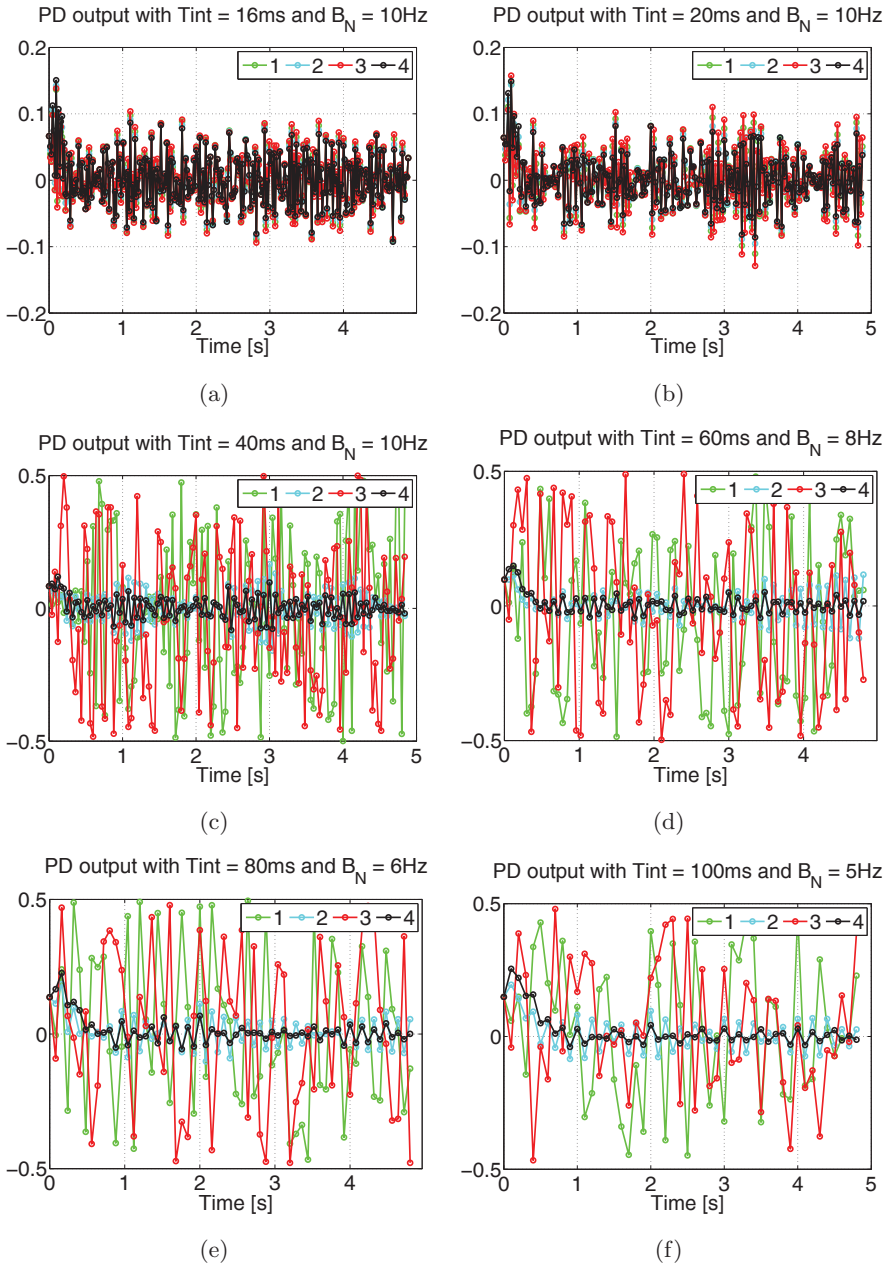


Figure 3.59: PD estimated error with constant carrier frequency estimation - testing different loop filters while tracking a constant 2 kHz Doppler frequency of a signal with a CNR of 30 dB-Hz and using different B_N and PDI values yielding a range of B_L values a) $T_I = 16$ ms and $B_L = 0.16$ b) $T_I = 20$ ms and $B_L = 0.2$ c) $T_I = 40$ ms and $B_L = 0.4$ d) $T_I = 60$ ms and $B_L = 0.48$ e) $T_I = 80$ ms and $B_L = 0.48$ f) $T_I = 100$ ms and $B_L = 0.5$.

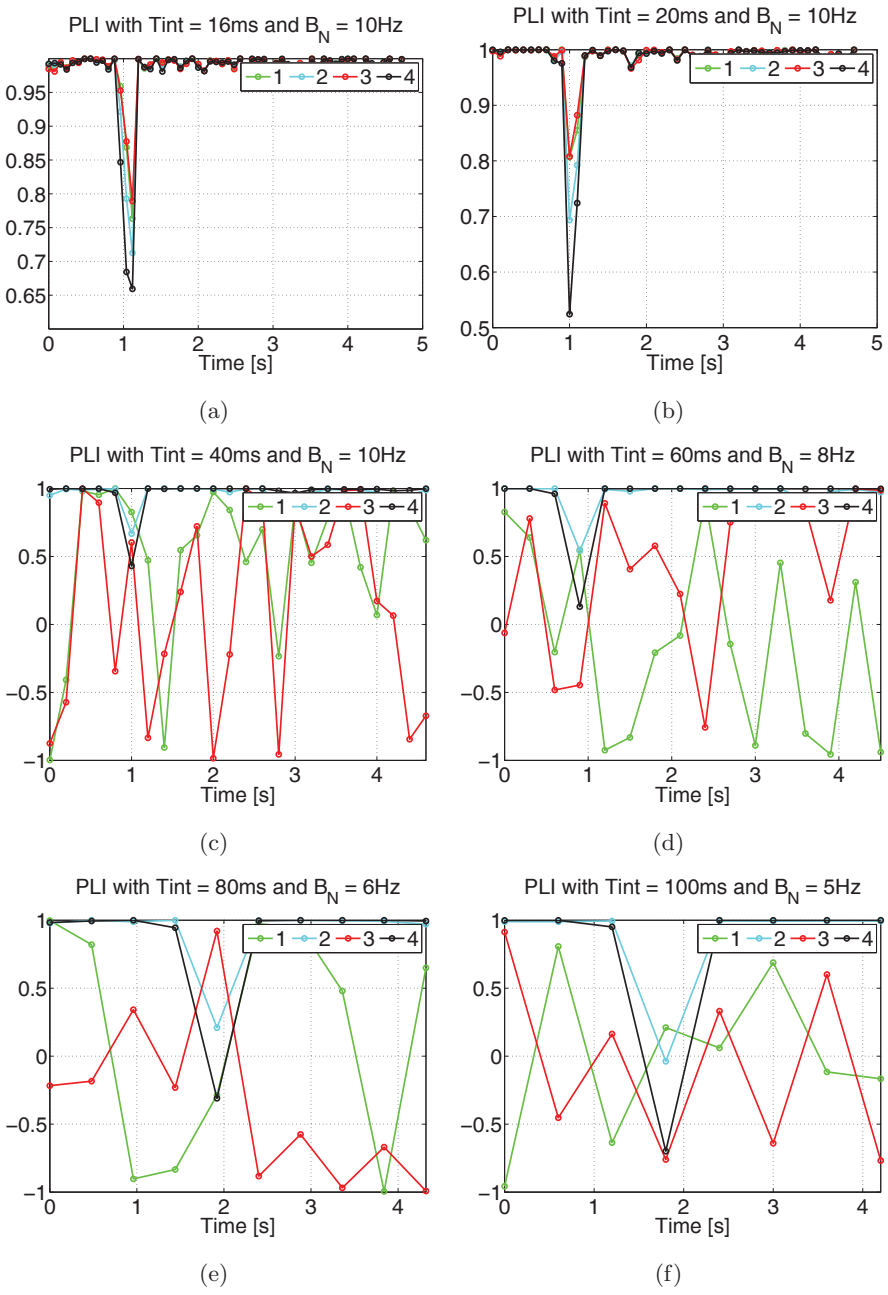


Figure 3.60: PLI with constant carrier frequency estimation - testing different loop filters to track a constant 2 kHz Doppler frequency of a signal with a CNR of 30 dB-Hz and using different B_N and PDI values yielding a range of B_L values a) $T_I = 16$ ms and $B_L = 0.16$ b) $T_I = 20$ ms and $B_L = 0.2$ c) $T_I = 40$ ms and $B_L = 0.4$ d) $T_I = 60$ ms and $B_L = 0.48$ e) $T_I = 80$ ms and $B_L = 0.48$ f) $T_I = 100$ ms and $B_L = 0.5$.

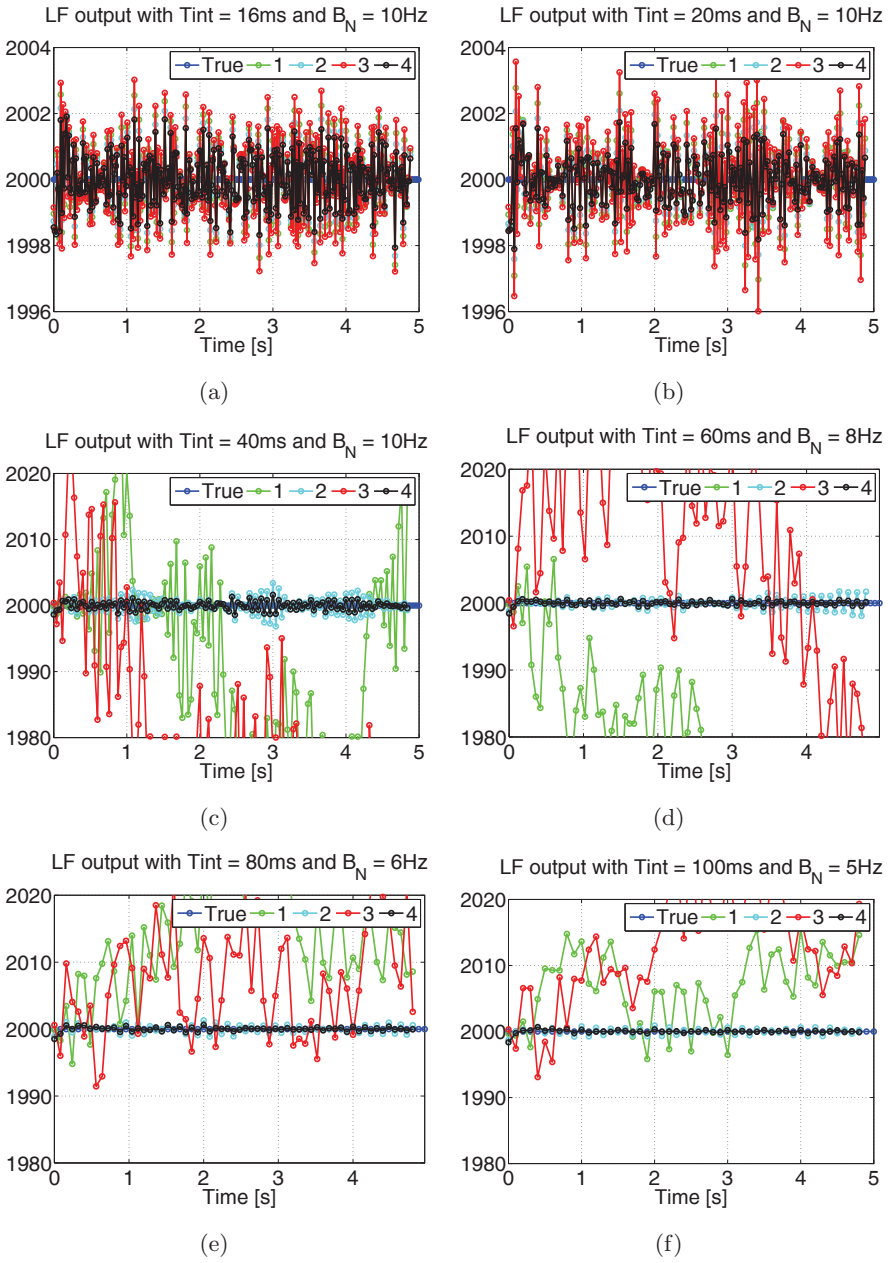


Figure 3.61: Loop filter outputs with constant carrier frequency estimation - testing different loop filters to track a constant 2 kHz Doppler frequency of a signal with a CNR of 30 dB-Hz and using different B_N and PDI values yielding a range of B_L values a) $T_I = 16$ ms and $B_L = 0.16$ b) $T_I = 20$ ms and $B_L = 0.2$ c) $T_I = 40$ ms and $B_L = 0.4$ d) $T_I = 60$ ms and $B_L = 0.48$ e) $T_I = 80$ ms and $B_L = 0.48$ f) $T_I = 100$ ms and $B_L = 0.5$.

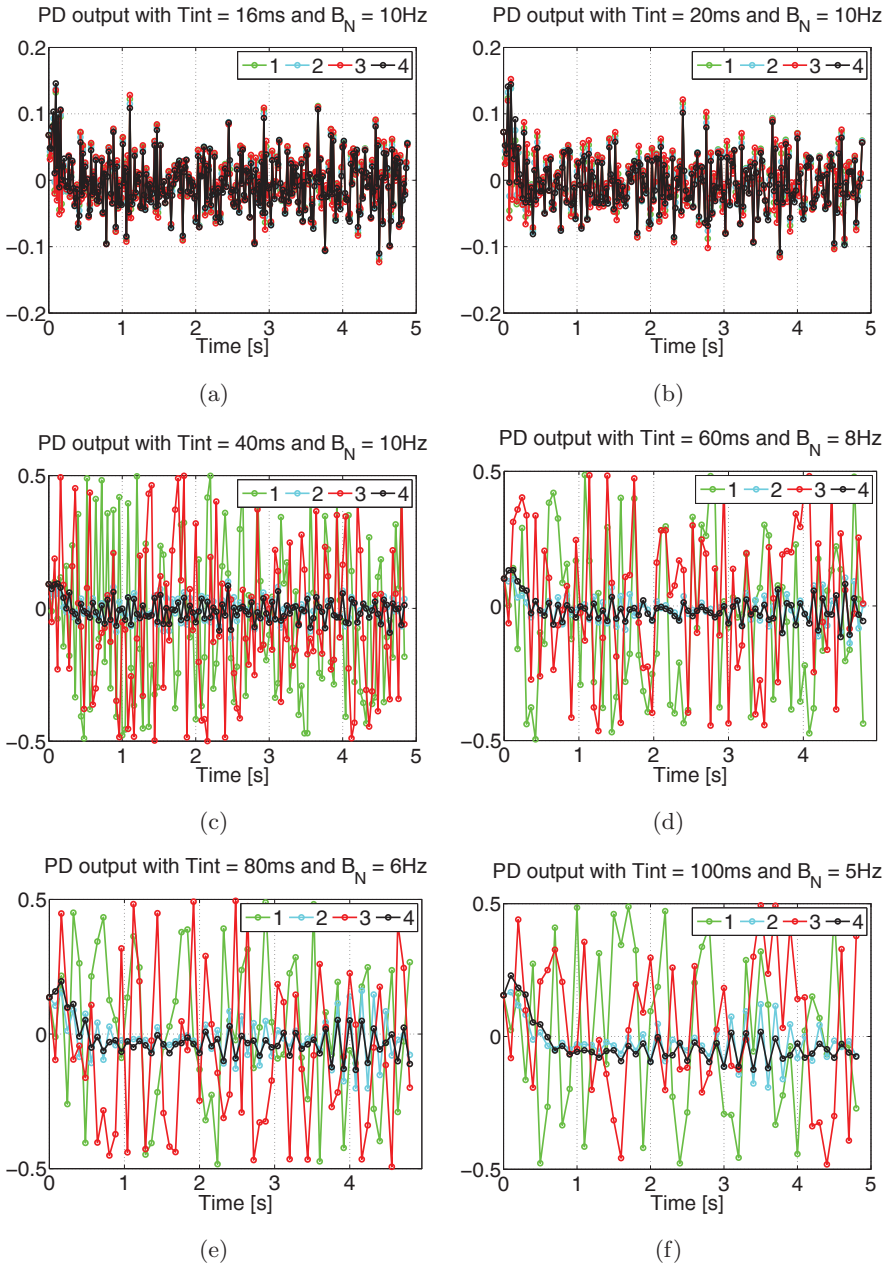


Figure 3.62: PD estimated error with a ramp carrier frequency estimation - testing different loop filters while tracking a variable Doppler frequency of a signal with an initial value of 2 kHz and a Doppler rate of -0.8 Hz/s, a CNR of 30 dB-Hz and using different B_N and PDI values yielding a range of B_L values a) $T_I = 16$ ms and $B_L = 0.16$ b) $T_I = 20$ ms and $B_L = 0.2$ c) $T_I = 40$ ms and $B_L = 0.4$ d) $T_I = 60$ ms and $B_L = 0.48$ e) $T_I = 80$ ms and $B_L = 0.48$ f) $T_I = 100$ ms and $B_L = 0.5$.

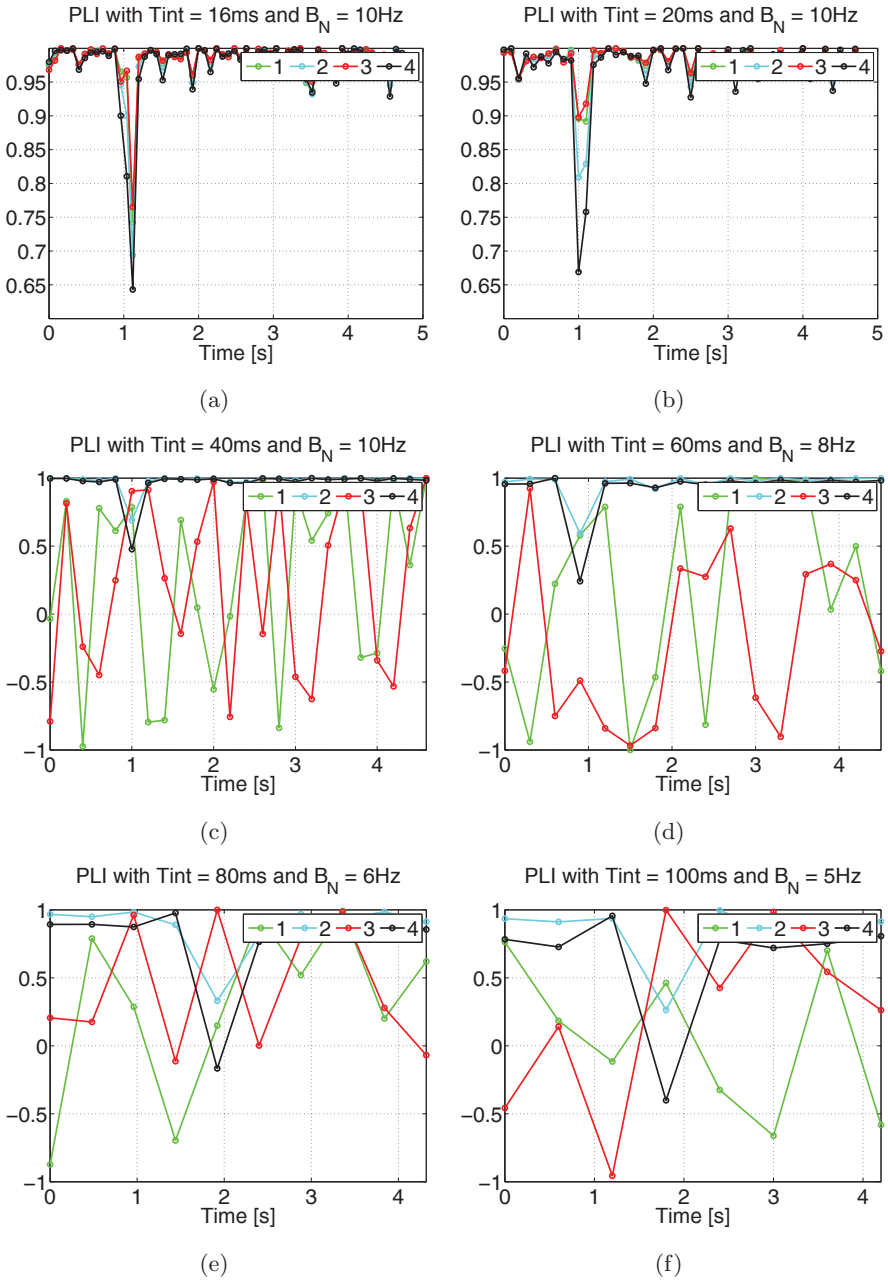


Figure 3.63: PLI with a ramp carrier frequency estimation - testing different loop filters while tracking a variable Doppler frequency of a signal with an initial value of 2 kHz and a Doppler rate of -0.8 Hz/s, a CNR of 30 dB-Hz and using different B_N and PDI values yielding a range of B_L values a) T_I = 16 ms and B_L = 0.16 b) T_I = 20 ms and B_L = 0.2 c) T_I = 40 ms and B_L = 0.4 d) T_I = 60 ms and B_L = 0.48 e) T_I = 80 ms and B_L = 0.48 f) T_I = 100 ms and B_L = 0.5.

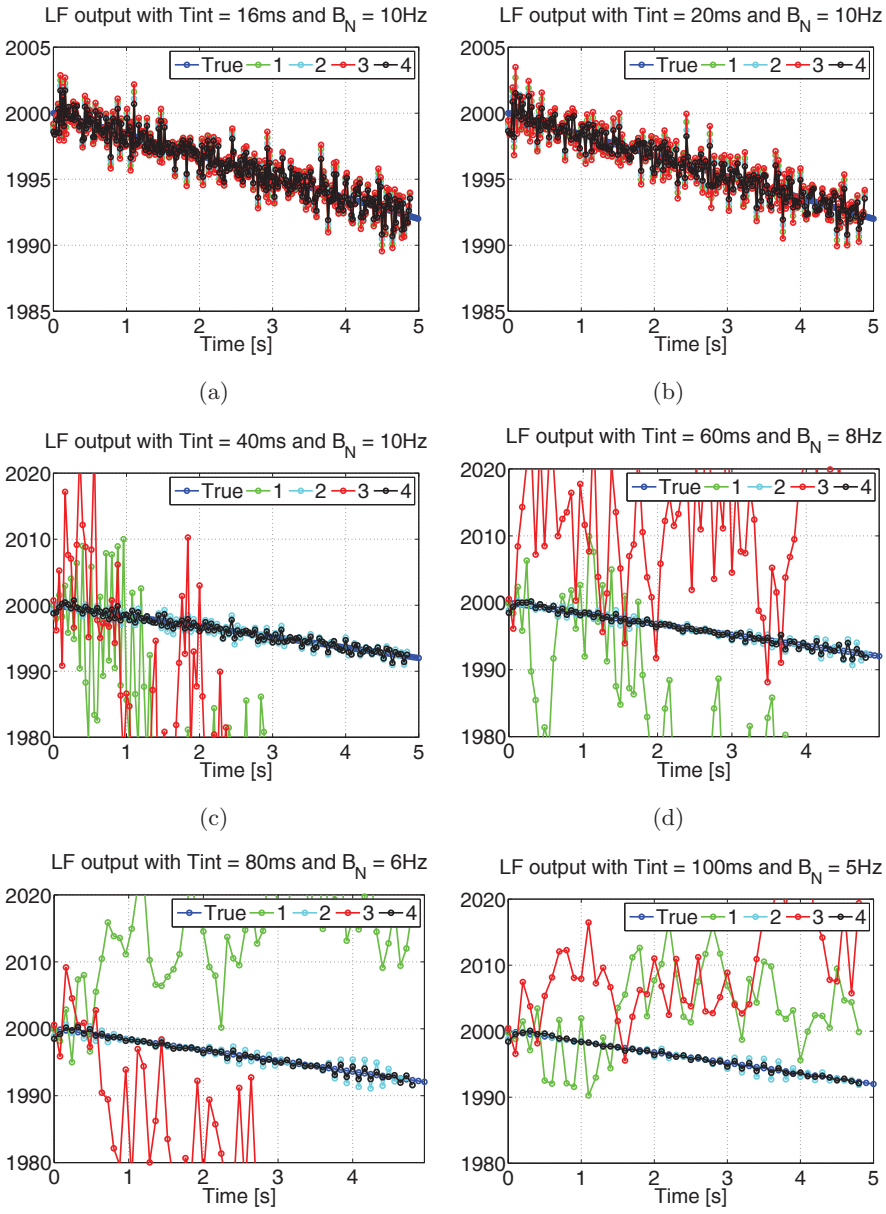


Figure 3.64: Loop filter outputs with a ramp carrier frequency estimation - testing different loop filters while tracking a variable Doppler frequency of a signal with an initial value of 2 kHz and a Doppler rate of -0.8 Hz/s, a CNR of 30 dB-Hz and using different B_N and PDI values yielding a range of B_L values a) $T_I = 16$ ms and $B_L = 0.16$ b) $T_I = 20$ ms and $B_L = 0.2$ c) $T_I = 40$ ms and $B_L = 0.4$ d) $T_I = 60$ ms and $B_L = 0.48$ e) $T_I = 80$ ms and $B_L = 0.48$ f) $T_I = 100$ ms and $B_L = 0.5$.

2 and 4. Root locus plots on the other hand have shown the progress of the nature of the loop response where the initially designed underdamped response moves towards critically damped and overdamped for methods 1 and 3 respectively, while complex conjugate poles characteristic of an underdamped response hold for high B_L values using methods 2 and 4.
It is worth mentioning that, following the Nyquist sampling bound $B_N < 1/(2T_I)$ as explained in Section 3.4.2.1, the noise equivalent bandwidth design parameter has been bounded with the increase of the integration interval, such that $B_N = 10$ Hz is chosen for integration intervals less than 40 ms. On the other hand, PDI values of 60 80 and 100 ms were used with a $B_N = 8$, 6 and 5 Hz.

3.5.3.4 Code tracking

Following the scheme of signal simulation described in Section 2.3.4 and after skipping the blocks implementing the carrier signal generation, the band-pass modulation and LPF, pure Galileo E1 and E5a spreading code signals with no carrier are generated over 5 seconds with a CNR of 45 dB-Hz and either the effect of:

- a constant Doppler frequency carrier of 2 kHz on the raw code rate,
- a ramp Doppler frequency of -0.8 Hz/s rate and an initial value of 2 kHz on the raw code rate.

Subsequently, the simulated signal is fed into the code tracking loop made up by a code PD, all of the aforementioned first order loop filters and a frequency NCO which is equivalent to a phase and rate NCO. The PD is selected as either the standard EMLE PD for Galileo E5a or the 4 taps VEMLE PD as described in Section 3.4.4.4 for the Galileo E1 signal.

Moreover, an initial code phase error of a few samples and an almost zero initial code rate error is applied while testing all PDI values. The reason behind this choice is because the Doppler frequency is scaled by the signal carrier frequency and hence results in a very low contribution in the spreading code rate as compared to the Doppler frequency contribution in the carrier frequency. Similarly to the carrier tracking case, the code tracking loop is governed by the Nyquist sampling bound $B_N < 1/(2T_I)$ as

explained in Section 3.4.2.1. Subsequently, the noise equivalent bandwidth design parameter has been set to $B_N = 5$ Hz in general, with the exception of $T_I = 60$ ms case where $B_N = 3.33$ Hz to yield $B_L = 0.2$ one of the considered values in the theoretical analysis put forth in Sections 3.5.3.1 and 3.5.3.2.

Figures 3.65 to 3.70 are the result of tracking the aforementioned simulated Galileo E1 signals and look into PD outputs, code lock indicators represented by the CNR and incremental loop filter code rate estimates. The incremental code rate means the additional code rate value on top of the nominal Galileo E1/E5 code rate values. These results show that for B_L values less than 0.2, all four loop filters exhibit similar code rate tracking performance. On the other hand, higher B_L values of 0.4 and 0.5 yield false or loss of tracking for methods 1 and 3, while method 2 and 4 show stable tracking for a B_L as high as 0.5. The results agree with the theoretical analysis performed earlier where the Bode plots have shown that for methods 1 and 3, the closed loop system becomes an all pass filter for $B_L > 0.2$ while the low pass filter characteristics remain for methods 2 and 4. Root locus plots on the other hand have shown the progress of the nature of the loop response where the initially designed underdamped response moves towards critically damped and overdamped for methods 1 and 3 respectively. However, complex conjugate poles which characterize an underdamped response still hold for high B_L values using methods 2 and 4.

Similarly, Figures 3.71 to 3.76 are the result of tracking the aforementioned simulated Galileo E5a signals. The same conclusions can be drawn for methods 1 and 3 as before while method 2 loses lock as soon as $B_L > 0.2$ and method 4 loses lock for $B_L \geq 0.5$. As mentioned before, method 2 and 4 yield a stable closed loop system with an underdamped response for B_L values as high as 0.5, however their frequency response curve is characterized by low attenuation of high frequencies and is therefore vulnerable to high frequency noise. It is interesting to notice that Galileo E1 and E5a signal tracking performance is different in this case where E1 tracking is possible for a B_L value as high as 0.5 while E5a tracking is limited by B_L value of 0.2 and 0.5 for methods 2 and 4 respectively. However, although the Loop Filter (LF) and NCO used in the tracking of each of these frequencies are identical, the PD is quite different due to the inherent nature of Galileo E1 and E5a/b signals. For Galileo E5a/b signals, a two

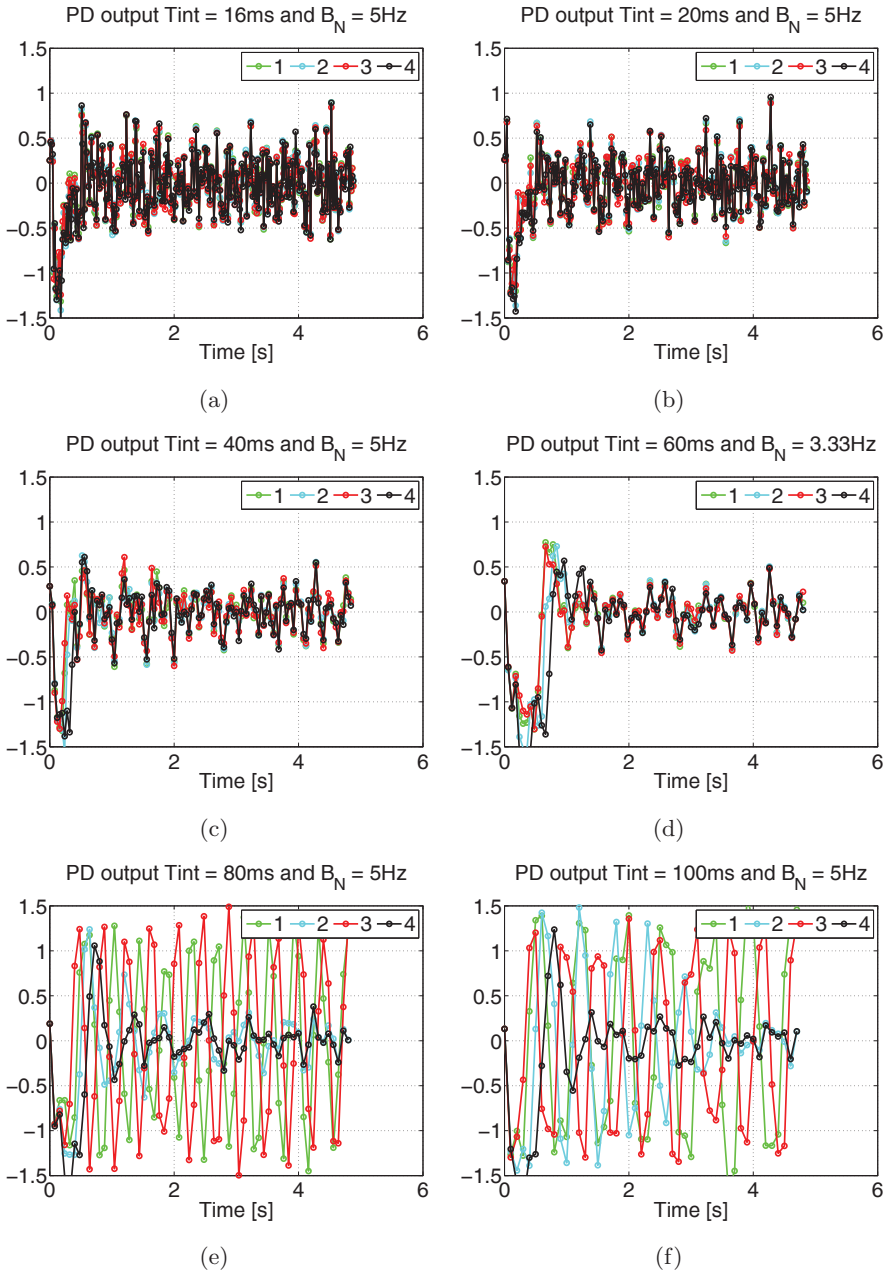


Figure 3.65: Four taps VEMLE PD output in chips with a constant code rate input on Galileo E1 band - testing different loop filters to track the effect of a constant 2 kHz Doppler frequency on the Galileo E1 code rate with a CNR of 45 dB-Hz and using different B_N and PDI values yielding a range of B_L values a) $T_I = 16$ ms and $B_L = 0.08$ b) $T_I = 20$ ms and $B_L = 0.1$ c) $T_I = 40$ ms and $B_L = 0.2$ d) $T_I = 60$ ms and $B_L = 0.2$ e) $T_I = 80$ ms and $B_L = 0.4$ f) $T_I = 100$ ms and $B_L = 0.5$.

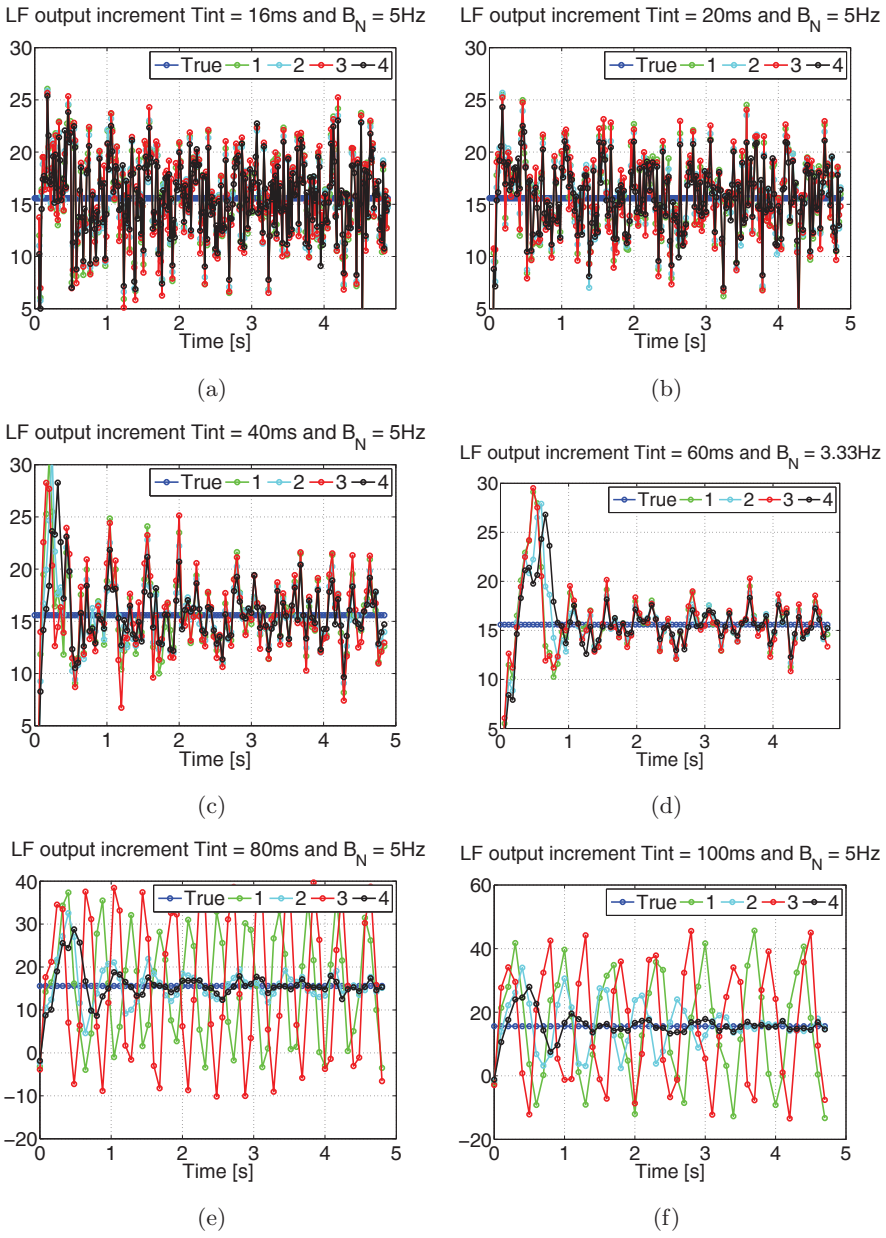


Figure 3.66: Loop filter constant code rate estimation on Galileo E1 band - testing different loop filters to track the effect of a constant 2 kHz Doppler frequency on the Galileo E1 code rate with a CNR of 45 dB-Hz and using different B_N and PDI values yielding a range of B_L values a) $T_I = 16$ ms and $B_L = 0.08$ b) $T_I = 20$ ms and $B_L = 0.1$ c) $T_I = 40$ ms and $B_L = 0.2$ d) $T_I = 60$ ms and $B_L = 0.2$ e) $T_I = 80$ ms and $B_L = 0.4$ f) $T_I = 100$ ms and $B_L = 0.5$.

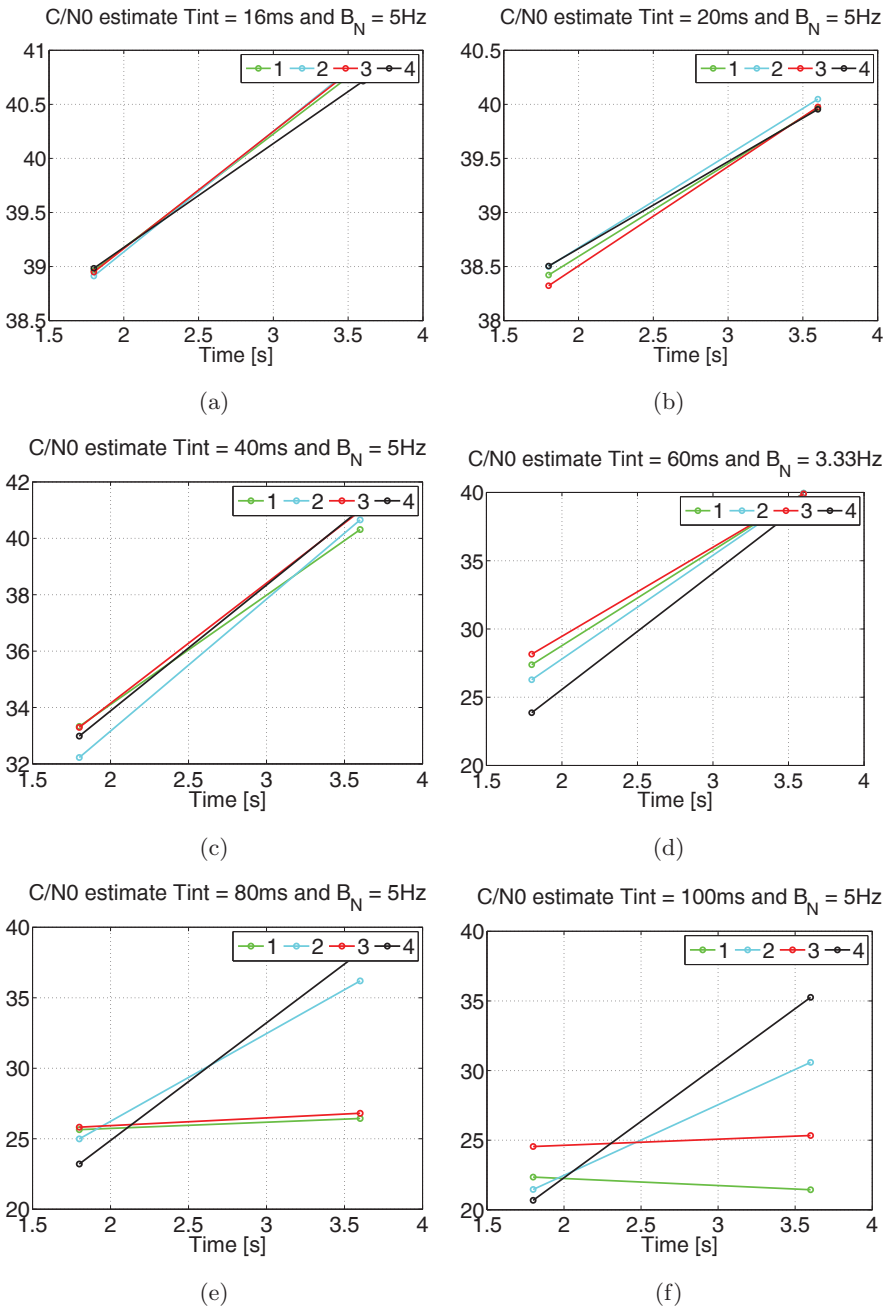


Figure 3.67: CNR estimate in dB-Hz with a constant code rate input on Galileo E1 band - testing different loop filters to track the effect of a constant 2 kHz Doppler frequency on the Galileo E1 code rate with a CNR of 45 dB-Hz and using different B_N and PDI values yielding a range of B_L values a) $T_I = 16$ ms and $B_L = 0.08$ b) $T_I = 20$ ms and $B_L = 0.1$ c) $T_I = 40$ ms and $B_L = 0.2$ d) $T_I = 60$ ms and $B_L = 0.2$ e) $T_I = 80$ ms and $B_L = 0.4$ f) $T_I = 100$ ms and $B_L = 0.5$.

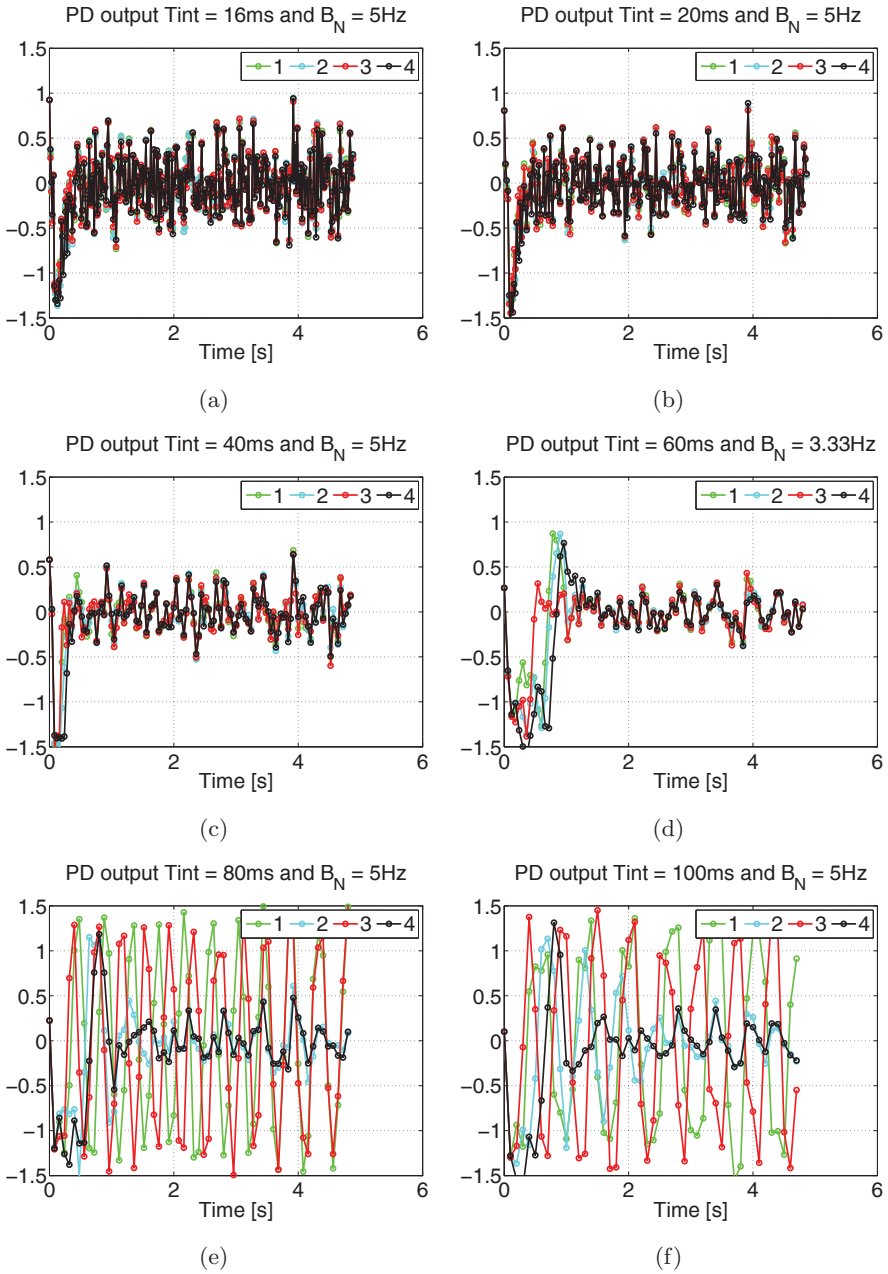


Figure 3.68: Four taps VEMLE PD output in chips with a ramp code rate input on Galileo E1 band - testing different loop filters to track the effect of a variable Doppler frequency on the Galileo E1 code rate with a CNR of 45 dB-Hz and using different B_N and PDI values yielding a range of B_L values a) $T_I = 16$ ms and $B_L = 0.08$ b) $T_I = 20$ ms and $B_L = 0.1$ c) $T_I = 40$ ms and $B_L = 0.2$ d) $T_I = 60$ ms and $B_L = 0.2$ e) $T_I = 80$ ms and $B_L = 0.4$ f) $T_I = 100$ ms and $B_L = 0.5$.

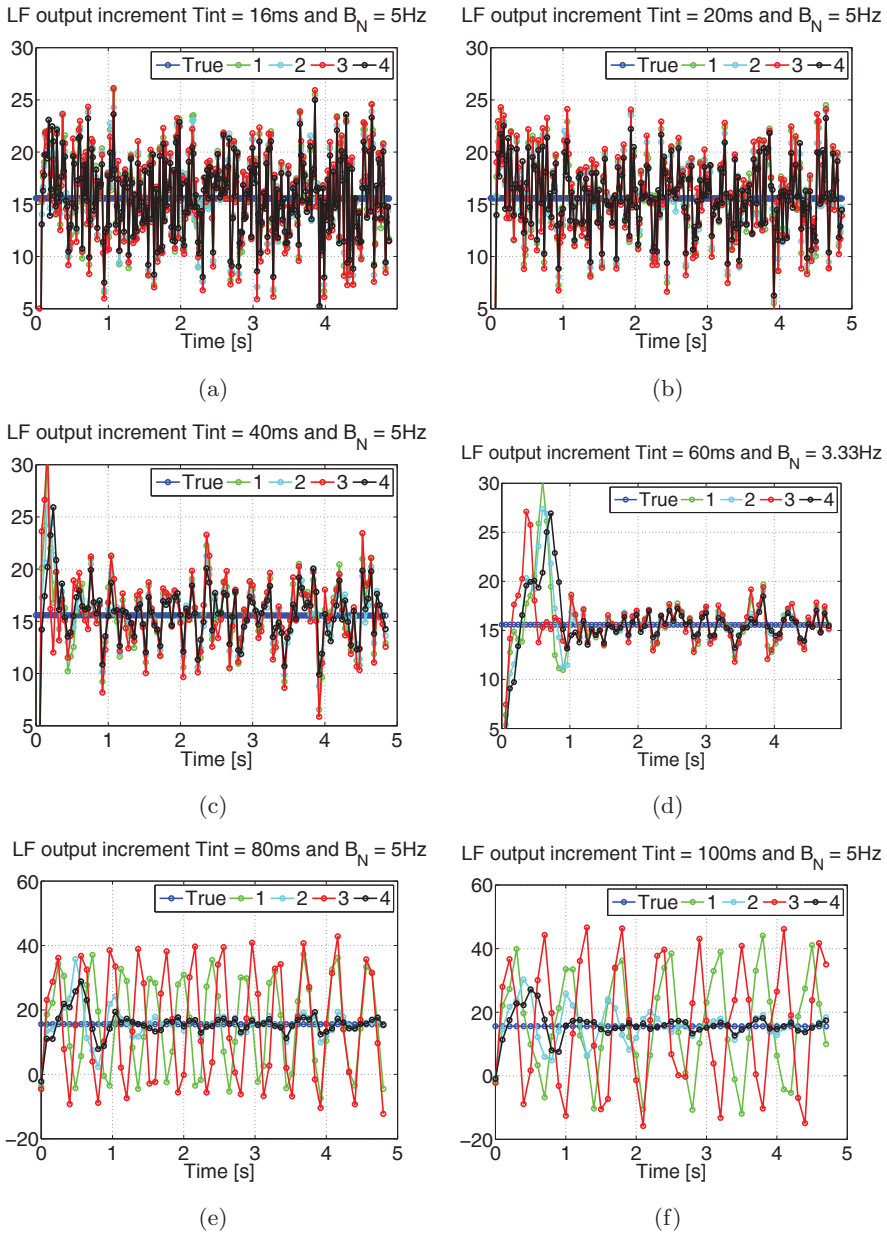


Figure 3.69: Loop filter ramp code rate estimation on Galileo E1 band - testing different loop filters to track the effect of a variable Doppler frequency on the Galileo E1 code rate with a CNR of 45 dB-Hz and using different B_N and PDI values yielding a range of B_L values a) $T_I = 16$ ms and $B_L = 0.08$ b) $T_I = 20$ ms and $B_L = 0.1$ c) $T_I = 40$ ms and $B_L = 0.2$ d) $T_I = 60$ ms and $B_L = 0.2$ e) $T_I = 80$ ms and $B_L = 0.4$ f) $T_I = 100$ ms and $B_L = 0.5$.

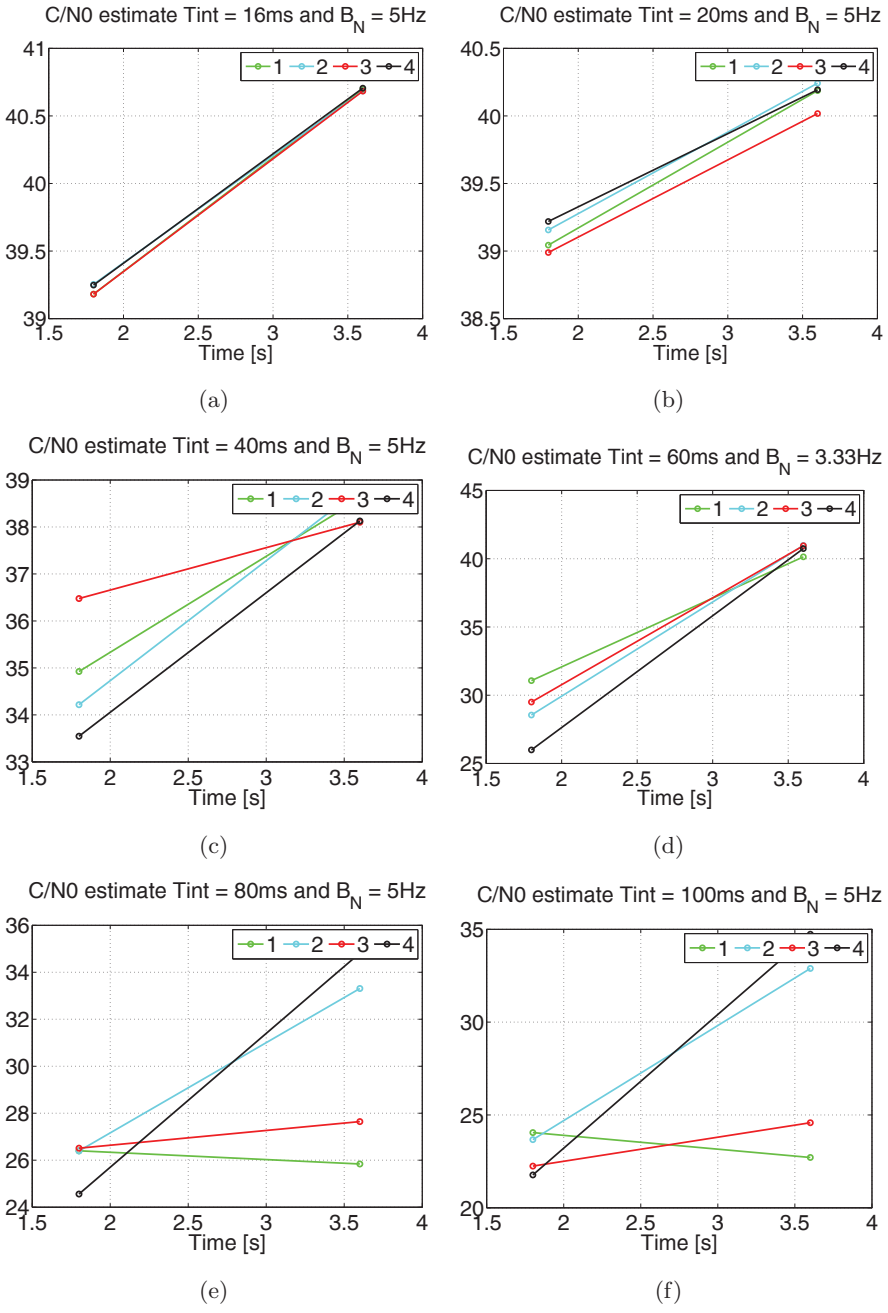


Figure 3.70: CNR estimate in dB-Hz with a ramp code rate input on Galileo E1 band - testing different loop filters to track the effect of a variable Doppler frequency on the Galileo E1 code rate with a CNR of 45 dB-Hz and and using different B_N and PDI values yielding a range of B_L values a) $T_I = 16$ ms and $B_L = 0.08$ b) $T_I = 20$ ms and $B_L = 0.1$ c) $T_I = 40$ ms and $B_L = 0.2$ d) $T_I = 60$ ms and $B_L = 0.2$ e) $T_I = 80$ ms and $B_L = 0.4$ f) $T_I = 100$ ms and $B_L = 0.5$.