

UCGE Reports

Number 20227

**Department of Geomatics Engineering**

**Design of Galileo L1F Receiver Tracking Loops**

(URL: <http://www.geomatics.ucalgary.ca/links/GradTheses.html>)

**by**

**Olivier Julien**

**July 2005**



UNIVERSITY OF CALGARY

Design of Galileo L1F Receiver Tracking Loops

by

Olivier Julien

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES  
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE  
DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF GEOMATICS ENGINEERING

CALGARY, ALBERTA

JULY, 2005

© Olivier Julien 2005

## Abstract

The ever-increasing public demand for location and positioning services has generated a demand for higher performance Global Navigation Satellite Systems (GNSS). The Galileo L1F signal, part of the European contribution to future GNSS, was designed to respond to the above demand in terms of measurement accuracy, tracking robustness and tracking sensitivity. Since the Galileo L1F public signal structure is almost fully specified, this thesis proposes novel tracking loop designs to increase code and phase measurements performance for potential implementation in future receivers.

Using detailed modeling of each error that affects phase tracking, the Galileo L1F signal is shown to result in increased phase tracking robustness under dynamics, providing at least a 5-dB higher sensitivity and delivering more accurate measurements than the GPS C/A signal thanks to the availability of a dataless channel. Further phase measurement accuracy can be achieved through two proposed techniques that combine pilot channel phase tracking with the less robust data channel tracking. These techniques still maintain the same high resistance level towards dynamics as pilot channel-only phase tracking, thus proving to be highly beneficial for precise positioning applications.

A thorough analysis of each critical parameter for Delay Lock Loop (DLL) design shows the advantages of the Binary Offset Carrier (BOC) modulation used by Galileo L1F in terms of intrinsic loop robustness, and thermal noise and multipath mitigation as

compared to GPS C/A code tracking. A new proposed tracking concept offers further multipath mitigation improvement by taking advantage of the potential higher correlation gain offered by the dataless channel, thus improving the measurement accuracy in degraded environments.

Although shown to be extremely beneficial for general code tracking, the BOC modulation is unfortunately also susceptible to yielding biased code measurements that would reduce its attractiveness as a ranging signal. To remove this threat, an innovative acquisition and tracking technique specific to the Galileo L1F signal, referred to as Autocorrelation Side-Peak Correlation Technique (ASPeCT), that provides fully reliable and unbiased code measurements with noise and multipath mitigation equivalent to traditional BOC tracking, is developed and tested. It is consequently an excellent candidate for implementation in future Galileo receivers, especially when measurement reliability is a concern.

## Acknowledgements

There are special people I want to acknowledge because they were extremely important to me during the course of my thesis:

- **Gérard Lachapelle** and **M. Elizabeth Cannon**, my supervisors, for your constant support and guidance over the years. You were essential to my interest in Research. Thank you for your trust and for believing in me.
- **Ma famille. Mes parents** en particulier. Votre amour, votre soutien continu et inaltérable ainsi que l'éducation que vous m'avez apportée sont mes pierres fondatrices sur lesquelles je pourrais toujours m'appuyer. Ma soeurette **Emilie** pour ton dynamisme, ta constante bonne humeur et ton soutien (tu m'as montré l'exemple!). Je pense également beaucoup au **reste de ma famille**, entre France et Catalunya, même si nous avons été séparés par de nombreux kilomètres ces dernières années.
- **Barbarita** because you are soooo special to me. Thank you for everything you brought (bring!) me, for your happiness, your ideas, and your laughs and smiles (and for making me laugh and smile).
- All the friends I met in Calgary. You were essential pieces of my life in Canada (I still count on you for the future!) and made my stay here a great experience. I am

really glad to have met you. I want to specially thank **Moïse, Arnaud, Marina, Oleg, Diep, Leo, Glenn, Jen', Cécile, Junjie, Mark, Akiko, Jussi, Heidi,** and **Nat'**.

- All the members of the **Department of Geomatics Engineering** and particularly the **PLAN group** for the comfortable environment you provided me during this thesis.
- **Christophe Macabiau** that, although always extremely busy, took time to discuss many of my problems and advise me.
- The **Alberta Ingenuity Fund** for their financial support during my thesis.

# Table of Contents

Abstract.....	ii
Acknowledgements.....	iv
Table of Contents.....	vi
List of Tables .....	x
List of Figures and Illustrations .....	xi
List of Symbols and Abbreviations.....	xviii
CHAPTER 1 .....	1
INTRODUCTION .....	1
1.1 Background and Motivation .....	2
1.2 Thesis Objectives.....	8
1.3 Thesis Contributions.....	11
1.4 Thesis Outline.....	12
CHAPTER 2 .....	15
CURRENT GPS C/A SIGNAL STRUCTURE AND INNOVATIONS BROUGHT BY GALILEO L1F SIGNAL .....	15
2.1 GPS C/A Signal Structure.....	16
2.1.1 Brief History .....	16
2.1.2 Signal Structure.....	18
2.1.3 Correlation Properties .....	24
2.2 GPS C/A Signal Limitations.....	32
2.2.1 Tracking Sensitivity .....	32
2.2.2 Tracking Reliability .....	35
2.2.3 Tracking Accuracy.....	36
2.2.4 Tracking Robustness.....	39
2.3 Galileo L1F Signal Structure .....	40
CHAPTER 3 .....	52
SIMULATION TOOLS.....	52
3.1 GNSS Error Sources .....	53
3.1.1 Satellite and Ephemeris Errors.....	53

3.1.2 Satellite and Receiver Oscillator Frequency Noise .....	54
3.1.3 Ionosphere.....	59
3.1.4 Troposphere .....	60
3.1.5 Multipath.....	60
3.1.6 Thermal Noise.....	62
3.1.7 Interference .....	63
3.1.8 Dynamics .....	63
3.2 IF GPS C/A and Galileo L1F Software Signal Generator .....	64
3.2.1 Why an IF Signal Simulator?.....	64
3.2.2 Initial GPS C/A Software Simulator.....	65
3.2.3 Galileo Constellation .....	70
3.2.4 Galileo L1F Waveform Model.....	70
3.2.5 Satellite Oscillator Model .....	72
3.2.6 Multipath Modeling .....	75
3.2.7 Antenna Gain Pattern.....	83
3.2.8 Front-End Design.....	85
3.3 IF Software Receiver Tracking Loops.....	87
3.3.1 General Architecture .....	87
3.3.2 Receiver Oscillator Phase Noise Model.....	89
CHAPTER 4 .....	91
GALILEO L1F CARRIER PHASE TRACKING .....	91
4.1 Generic PLL Structure .....	92
4.2 Phase Discriminators .....	94
4.2.1 Data Channel.....	95
4.2.2 Pilot Channel.....	97
4.3 Sources of Error of a PLL.....	99
4.3.1 Thermal Noise.....	99
4.3.2 Dynamic Stress Error.....	109
4.3.3 Oscillator Frequency Noise.....	111
4.3.4 Oscillator Vibrations .....	115
4.3.5 Conclusion on Phase Tracking Error Modeling.....	116
4.4 Sensitivity Analysis .....	118
4.5 PLL Frequency Error .....	125
4.6 Data/Pilot Implementation .....	126
4.7 Synthesis .....	140
CHAPTER 5 .....	142
GALILEO L1F CODE DELAY TRACKING .....	142
5.1 Generic DLL Structure .....	143
5.1.1 Structure Overview .....	143
5.1.2 Discriminators.....	146
5.1.3 Discriminator Normalization .....	154
5.2 Error Sources .....	161
5.2.1 Thermal Noise.....	162
5.2.2 Multipath.....	185

5.2.3 Dynamic Stress Error .....	188
5.3 DLL Sensitivity Analysis.....	189
5.4 Galileo L1F Multipath Mitigation Technique .....	197
5.4.1 High Resolution Correlator (HRC).....	198
5.4.2 Multipath Mitigation Technique Implementation.....	204
5.4.3 Test of Multipath Mitigation Capability .....	207
5.5 Synthesis .....	214
CHAPTER 6 .....	216
ASPECT: A NEW UNAMBIGUOUS CODE TRACKING TECHNIQUE FOR GALILEO L1F.....	216
6.1 Galileo L1F Ambiguous Acquisition and Tracking .....	217
6.1.1 Acquisition.....	217
6.1.2 Tracking .....	221
6.1.3 Main Indicated Solutions .....	223
6.2 ASPeCT .....	235
6.3 Impact of Thermal Noise and Multipath on ASPeCT .....	246
6.3.1 Thermal Noise.....	246
6.3.2 Multipath.....	257
6.3.3 Conclusions on ASPeCT's Error Mitigation .....	259
6.4 Acquisition Using ASPeCT's Synthesized Correlation Function .....	261
CHAPTER 7 .....	272
CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK.....	272
7.1 Conclusions.....	272
7.2 Recommendations for Future Work .....	276
REFERENCES .....	278
APPENDIX A.....	286
CORRELATOR OUTPUT NOISE POWER AND CORRELATION .....	286
APPENDIX B .....	290
PLL FREQUENCY ERROR VARIANCE .....	290
APPENDIX C .....	294
DOT-PRODUCT DISCRIMINATOR VARIANCE IN THE PRESENCE OF GAUSSIAN NOISE.....	294
C.1 Background .....	294
C.2 Loop Gain.....	296
C.3 Cross-Correlation between $N_1$ and $N_2$ .....	298
C.4 $N_1$ PSD in $f=0$ .....	299
C.5 $N_2$ PSD in $f=0$ .....	302
C.6 Conclusion.....	303

APPENDIX D.....	305
ASPECT DOT-PRODUCT DISCRIMINATOR STANDARD DEVIATION IN THE PRESENCE OF GAUSSIAN NOISE .....	305
D.1 Background .....	305
D.2 Loop Gain .....	309
D.3 $N_4$ PSD in $f=0$ .....	311
C.4 Contribution of the Correlation between $N_1$ and $N_3$ .....	311
D.5 Contribution of the Correlation between $N_2$ and $N_4$ .....	312
D.6 Conclusion .....	313
 APPENDIX E .....	 318
ASPECT EMLP DISCRIMINATOR VARIANCE IN THE PRESENCE OF GAUSSIAN NOISE.....	318
E.1 Background .....	318
E.2 Loop Gain.....	321
E.3 Conclusion.....	323

## List of Tables

Table 2.1 – BPSK(1) and sBOC(1,1) Spectral Characteristics.....	45
Table 2.2 – Galileo L1F Signal Characteristics .....	51
Table 3.1 – Parameters for a Set of Oscillators (Winkel 2003).....	55
Table 3.2 – Ground Reflection Delay for 6 Satellites in View and an Antenna 2 m Above the Ground.....	78
Table 4.1 – Standard Deviation of the Phase Change Rate (rad) during a Coherent Integration for Various Oscillators .....	113
Table 4.2 – 95 % Phase Change Rate (rad) during a Coherent Integration for Various Oscillators .....	113
Table 4.3 – Maximum Coherent Integration Times due to the Oscillator Frequency Noise for Various Oscillators and a Maximum Tolerable Phase Error of $\pi/4$ and $\pi$ .....	114
Table 4.4 - Hard $C/N_0$ Threshold corresponding to the Discriminator Slope at the Origin being Lower than 0.9 (Left Value) and 1 (Right Value) .....	120
Table 4.5 – Tracking Thresholds for the Coherent, DP, Atan and Atan2 Discriminators Assuming a Coherent Integration of 4 ms, and Three Levels of Dynamics (Jerk = 0, 1, 2 g/s) .....	123
Table 4.6 – Tracking Thresholds for the Coherent, DP, Atan and Atan2 Discriminators Assuming a Coherent Integration of 20 ms, and Three Levels of Dynamics (Jerk = 0, 1, 2 g/s) .....	124
Table 4.7 – Phase Tracking Standard Deviation for the Three Combinations Presented for a $C/N_0$ of 30 dB-Hz and a Coherent Integration of 4 ms. ....	138
Table 5.1 – Hard $C/N_0$ Threshold due to the Decrease of the Mean Discriminator Slope at the Origin (for Slope equal to 0.9 on the Left and 1 on the Right) .	181
Table 5.2 – Galileo L1F $C/N_0$ Threshold for Different Coherent Integration Times and using a DP Discriminator with a Correlator Spacing of 0.2 Chips .....	197

## List of Figures and Illustrations

Figure 1.1 - GPS and Galileo Expected Spectral Occupation of the L1 Frequency Band.....	6
Figure 2.1 – Example of GPS C/A Code (Left) Normalized Autocorrelation Function, and (Right) an exploded view about $\pm 10$ Chips .....	20
Figure 2.2 – Power Spectral Densities of the Navigation Data, GPS C/A Spreading Code, and Typical Thermal Noise Assuming a Signal Power of -160 dBW.....	21
Figure 2.3 – Schematic Representation of the Correlation Process in a GNSS Receiver .....	25
Figure 2.4 – GPS C/A Signal Power Loss due to the Front-End Filter Bandwidth.....	30
Figure 2.5 – Impact of the Front-End Filter Bandwidth (One-Sided) on the Spreading Code Autocorrelation Function .....	31
Figure 2.6 – sBOC(1,1) and BPSK(1) Normalized Power Spectral Densities .....	44
Figure 2.7 – sBOC(1,1) and BPSK(1) Signal Power Loss due to the (One-Sided) Front-End Filter Bandwidth.....	46
Figure 2.8 – Impact of the (One-Sided) Front-End Filter Bandwidth on the sBOC(1,1) Autocorrelation Function .....	47
Figure 2.9 – Galileo L1F Signal Generation Architecture.....	50
Figure 3.1 – Allan Standard Deviation for Quartz, TCXO, OCXO, Rubidium and Caesium Oscillators.....	56
Figure 3.2 – Impact of an In-Phase Multipath with a Delay of 0.5 Chips on the Normalized sBOC(1,1) Correlation Function .....	62
Figure 3.3 – Initial IF GPS Simulator Structure (from Dong <i>et al.</i> 2004).....	66
Figure 3.4 - GPS C/A Ionospheric Errors for all Visible Satellites during a 30 Minute Period.....	68
Figure 3.5 – Normalized PSD of the Simulated Galileo L1F Signal and of the Theoretical sBOC(1,1) Envelope .....	72
Figure 3.6 – Theoretical and Simulated Allan Standard Deviation for Standard Caesium and Rubidium Oscillators .....	73
Figure 3.7 – Example of Caesium and Rubidium Oscillator (1575.42 MHz) Phase Error over 4 Hours .....	74
Figure 3.8 – Repartition (Left), and Normalized Power (Right) of 500 Small Reflectors in a 100-Metre Circle around the User.....	77
Figure 3.9 – Ground Reflection Model.....	78
Figure 3.10 – Example of the Obstacle Reflection/Blocking Model.....	79
Figure 3.11 – Far Echo Multipath Test Configuration .....	81

Figure 3.12 – Estimated Signal Amplitude for the Seven Satellites Simulated During the First 1.75 Seconds.....	82
Figure 3.13 – Estimated Signal Amplitude During the Acquisition Process Using a Uniform Unit Gain Antenna (Top Left), a NovAtel GPS 501 Antenna (Top Right), a NovAtel GPS 503 Antenna (Bottom Left), and a NovAtel GPS 600 Antenna (Bottom Right) in Presence of a Strong Multipath Coming from a 10 Degree Elevation .....	84
Figure 3.14 – Simulated Received Signal Power as a Function of Elevation Angle.....	86
Figure 3.15 – High-Level Block Diagram of a GNSS Receiver.....	88
Figure 3.16 – Block Diagram of the GNSS Software Receiver Carrier Generator for the In-Phase Correlation Component .....	89
Figure 3.17 - Theoretical and Simulated Allan Standard Deviation for Standard Caesium and Rubidium Oscillators .....	90
Figure 3.18 - Example of Quartz, TCXO and OCXO Oscillators (1575.42 MHz) Phase Error over 10 (Left) and 1 (Right) Seconds .....	90
Figure 4.1 – Generic PLL Architecture .....	92
Figure 4.2 – Mean DP (Top Left), Atan (Top Right), Coherent (Bottom Left) and Atan2 (Bottom Right) Discriminator Output for 6 C/N <sub>0</sub> (20, 25, 30, 35, 40, and 45 dB-Hz) and a Coherent Integration Time of 4 ms .....	101
Figure 4.3 – Mean Atan (Left) and Atan2 (Right) Discriminator Output for 6 C/N <sub>0</sub> (20, 25, 30, 35, 40 and 45 dB-Hz) and a Coherent Integration Time of 20 ms.....	102
Figure 4.4 - Atan (Left) and Atan2 (Right) Discriminator Slope at the Origin for a Coherent Integration Time of 4 (Left) and 20 (Right) ms .....	103
Figure 4.5 – One-Sided Linear Tracking Domain Width for the Four Discriminators Considered and for a Coherent Integration Time of 4 (Left) and 20 (Right) ms .....	104
Figure 4.6 – Histogram of the Classical (Left) and Extended (Right) Arctangent Discriminators for a C/N <sub>0</sub> of 30 (Top) and 25 (Bottom) for a Coherent Integration Time of 4 ms .....	105
Figure 4.7 – Discriminator Output Standard Deviation for a 4 (Left) and 20 (Right) ms Coherent Integration Time .....	107
Figure 4.8 – PLL Phase Tracking Standard Deviation using One of the Four Discriminators Considered for Integration Times of 4 and 20ms and a 10 Hz Loop Bandwidth.....	108
Figure 4.9 – Comparison of Theory and Simulations for Extended Arctangent Discriminator Phase Tracking Error Standard Deviation for Integration Times of 1 and 20 ms and a 10 Hz Loop Bandwidth .....	109
Figure 4.10 – Arctangent Discriminator Output and Estimated PLL Phase Error Using a 40 dB-Hz Signal and a Quartz Oscillator as the Receiver Oscillator .....	111
Figure 4.11 – Modeled Vibration-Induced Phase Tracking Error Standard Deviation ...	116
Figure 4.12 –PLL Frequency Error Standard Deviation using a Coherent and a DP Discriminator for a Loop Bandwidth of 10 Hz, and a Coherent Integration Time of 4 ms .....	125

Figure 4.13 – Schematic Data/Pilot PLL Architecture .....	128
Figure 4.14 – PLL Tracking Error when using the Coherent Discriminator Only, the DP Discriminator Only, and Combined Coherent and DP Discriminators for an Integration Time of 4 ms and a 10 Hz PLL Loop Filter Bandwidth .....	129
Figure 4.15 – DP and Coherent Discriminator Output, Estimated Doppler and Estimated Phase Error using 4 ms Integration Time, a Loop Bandwidth of 10 Hz, a C/N <sub>0</sub> of 45 dB-Hz and an Initial Doppler Error of 6 Hz .....	132
Figure 4.16 – Coherent and DP Discriminators’ Output Standard Deviation for an Integration Time of 4 ms .....	134
Figure 4.17 – Pilot Instantaneous and Filtered (6 epochs) Discriminator Output using 4 ms Integration Time, a C/N <sub>0</sub> of 30 dB-Hz, with an Initial Doppler Offset of 0 Hz (Left), and 6 Hz (Right) .....	136
Figure 4.18 – PLL Tracking Error using the Three Data/Pilot Implementation with a 4 ms Integration Time, a C/N <sub>0</sub> of 30 dB-Hz, and an Initial Doppler Offset of 6 Hz .....	137
Figure 4.19 – Estimated Phase Error using Data/Pilot Combinations 2 and 3 with a C/N <sub>0</sub> of 30 dB-Hz, and a Quartz Oscillator .....	139
Figure 5.1 – Typical Architecture of a GNSS Delay Lock Loop .....	143
Figure 5.2 – EMLP (Left) and DP (Right) Discriminator Output for sBOC(1,1) (Top), and BPSK(1) (Bottom) as a Function of the One-Sided Front-End Filter Bandwidth for a Correlator Spacing of 0.2 Chips .....	150
Figure 5.3 – EMLP (Left) and DP (Right) Discriminator Output for sBOC(1,1) (Top), and BPSK(1) (Bottom) as a Function of the (One-Sided) Front-End Filter Bandwidth for a Correlator Spacing of 0.4 Chips .....	151
Figure 5.4 - Discriminator Output for sBOC(1,1) EMLP (Top Left) and DP (Top Right), and for BPSK(1) EMLP (Bottom Left) and DP (Bottom Right) as a Function of the Correlator Spacing for a 10 MHz One-Sided Front-End Bandwidth .....	153
Figure 5.5 – sBOC(1,1) EMLP (Left) and DP (Right) Discriminator Output for a 0.1 (Top) and 0.2 (Bottom) Chip Correlator Spacing with all the Normalization Considered for a 10 MHz One-Sided Front-End Filter Bandwidth.....	159
Figure 5.6 – BPSK(1) EMLP (Left) and DP (Right) Discriminator Output for a 0.1 (Top) and 0.2 (Bottom) Chip Correlator Spacing with all the Normalization Considered for a 10 MHz One-Sided Front-End Filter Bandwidth.....	160
Figure 5.7 – Variation of the Code Tracking Error Standard Deviation using EMLP (Left) and DP (Right) Discriminators as a Function of the (One-Sided) Front-End Bandwidth and the Correlator Spacing sBOC(1,1) Signal (Top) and BPSK(1) (Bottom) for a C/N <sub>0</sub> of 30 dB-Hz and a 4 ms Integration Time .....	166
Figure 5.8 – Variation of the Code Tracking Error Standard Deviation using EMLP (Left) and DP (Right) Discriminators as a Function of the (One-Sided) Front-End Bandwidth and the Correlator Spacing sBOC(1,1) Signal	

	(Top) and BPSK(1) (Bottom) for a $C/N_0$ of 45 dB-Hz and a 4 ms Integration Time .....	167
Figure 5.9	– Tracking Standard Deviation Improvement between sBOC(1,1) and BPSK(1) for Using EMLP (Left) and DP (Right) Discriminators for two $C/N_0$ (30 (Top) and 45 (Bottom) dB-Hz) for a 4 ms Integration Time .....	168
Figure 5.10	–sBOC(1,1) DP Discriminator using a 3 MHz One-Sided Front-End Filter, a 0.2 Chip Correlator Spacing, and $C/N_0$ values of 45 (Top Left), 40 (Top Right), 35 (Bottom Left), and 30 (Bottom Right) dB-Hz .....	172
Figure 5.11	- sBOC(1,1) DP Discriminator using a 3 MHz One-Sided Front-End Filter, a 0.2 Chip Correlator Spacing, a 45 (Top) and 30 (Bottom) dB-Hz and a 0.4 (Left) and 5 (Right) Chip Discriminator Hard Limiter .....	174
Figure 5.12	– sBOC(1,1) DP Discriminator Slope at the Origin for a 3 MHz One-Sided Front-End Filter, a 4 ms Integration Time, and a 0.2 Chips Correlator Spacing for a 0.4 (Left) and 5 (Right) Chips Discriminator Hard Limiter .....	176
Figure 5.13	– sBOC(1,1) DP Discriminator Slope at the Origin for a 3 MHz One-Sided Front-End Filter, a 20 ms Integration Time, and a 0.2 Chips Correlator Spacing for a 0.4 (Left) and 5 (Right) Chips Discriminator Hard Limiter .....	177
Figure 5.14	– Galileo L1F Receiver DLL Response under Gaussian Noise with a $C/N_0$ of 30 dB-Hz, a 4 ms Integration Time, a 1 Hz Loop Bandwidth With and Without a 0.2 Chips Discriminator Hard Limiter .....	178
Figure 5.15	- sBOC(1,1) DP Discriminator Output Using a 0.4 Chips Hard Limiter, a 3 MHz One-Sided Front-End Filter Bandwidth, a Correlator Spacing Value of 0.2 Chips for a $C/N_0$ of 25 dB-Hz (Left) and 45 dB-Hz (Right) .	179
Figure 5.16	– BPSK(1) DP Discriminator Output Using a 1.2 Chips Hard Limiter, a 3 MHz One-Sided Front-End Filter Bandwidth, a Correlator Spacing Value of 0.2 Chips for a $C/N_0$ of 25 dB-Hz (Left) and 45 dB-Hz (Right) .	179
Figure 5.17	– Code Tracking Error Standard Deviation with sBOC(1,1) Modulation, with a 0.4 Chips Hard Limiter for EMLP (Left) and DP (Right) Normalized Discriminators Using a 3MHz One-Sided Front-End Filter Bandwidth, a 0.2 Chips Correlator Spacing and an Integration Time of 4 ms.....	183
Figure 5.18	– Code Tracking Error Standard Deviation with BPSK(1) Modulation, with a 1.2 Chips Hard Limiter for EMLP (Left) and DP (Right) Normalized Discriminators Using a 3 MHz One-Sided Front-End Filter Bandwidth, a 0.2 Chips Correlator Spacing and an Integration Time of 4 ms.....	183
Figure 5.19	– Simulated, Theoretical and Empirical Code Tracking Error Standard Deviation with Galileo L1F Signal using a 0.4 Chips Hard Limiter, a DP Normalized Discriminators with a 3 MHz One-Sided Front-End Filter Bandwidth, a 0.2 Chips Correlator Spacing and an Integration Time of 4 ms.....	184

Figure 5.20 – Code Tracking Error Envelope for sBOC(1,1) (Top) and BPSK(1) (Bottom) Modulations for One Multipath with an Amplitude Half of the Direct Signal, for a DP Discriminator, Several Front-End Filters, and a Correlator Spacing of 0.2 (Left) and 0.4 (Right) Chips.....	187
Figure 5.21 – BPSK(1) DP Discriminator Output Assuming the Presence of One In-Phase Multipath with a Delay of 0.25 Chips and a Half the Magnitude of the Direct Signal.....	190
Figure 5.22 – Normalized In-Phase Correlation Value For Different Constant Frequency Errors .....	192
Figure 5.23 – Tracking Threshold as a Function of the DLL Loop Bandwidth and the Coherent Integration Time for a BPSK(1) (Left) and sBOC(1,1) (Right) Using a DP Discriminator with a Correlator Spacing of 0.2 Chips.....	195
Figure 5.24 – Example of Local Code Replica for sBOC(1,1) Tracking Using Traditional and HRC Tracking Techniques.....	199
Figure 5.25 – Normalized HRC ( $\delta_{HRC} = 0.05$ Chips) and Traditional Correlation Function for an sBOC(1,1) (Top) and a BPSK(1) (Bottom) Signal .....	200
Figure 5.26 – Traditional and HRC Multipath Envelopes for sBOC(1,1) (Top) and BPSK(1) (Bottom) Signals Using a 12 MHz One-Sided Front-End Filter and a 0.1 Chips Correlator Spacing .....	202
Figure 5.27 – HRC and Traditional Discriminators Output Using a 12 MHz One-Sided Front-End Filter and a 0.1 Chips Correlator Spacing.....	203
Figure 5.28 – Code Delay Residual Estimation for the Three Techniques Tested in the Static Case .....	209
Figure 5.29 – Code Delay Residual Estimation for the Three Technique Tested for in the Static Case (10 Epochs Average and Threshold of 0.15 Chips for the HRC/Traditional Technique).....	211
Figure 5.30 - Code Delay Residual Estimation for the Three Techniques Tested (Left) and Test Output for the Combined Implementation (Right) in the Dynamic Case.....	212
Figure 6.1 – Probability of Detection of the Main and Secondary Peaks of the sBOC(1,1) Signals for 15 and 50 Non-Coherent Summations and a Coherent Integration Time of 4 ms.....	220
Figure 6.2 – Example of False Lock Occurring after False Acquisition on a Secondary Peak.....	223
Figure 6.3 – Normalized Correlation Function between the Incoming sBOC(1,1) Signal and a PRN Local Replica (no Sub-Carrier) Shifted by $f_c$ .....	225
Figure 6.4 – Normalized Power Spectrum Densities of sBOC(1,1) and BPSK(1) shifted by $f_c$ .....	226
Figure 6.5 – Probability that the Magnitude of one of the Secondary Peaks is Higher than the Magnitude of the Main Peak, Assuming Unbiased Tracking.....	230
Figure 6.6 – Probability that the Magnitude of one of the Secondary Peaks is Higher than the Magnitude of the Main Peak, Assuming Unbiased Tracking, for a 20 ms Integration Time, With and Without the Correlated Noise Assumption, and with Different VE – Prompt Separations.....	231

Figure 6.7 - Probability that the Magnitude of the Main Peak is Greater than the Magnitude of the Secondary Peak, Assuming Biased Tracking.....	232
Figure 6.8 - Probability that the Magnitude of one of the Secondary Peaks is Higher than the Magnitude of the Main Peak, Assuming Unbiased Tracking and One Stable Multipath in Phase Opposition with an Amplitude Half of the Direct Signal.....	233
Figure 6.9 – sBOC(1,1)/sBOC(1,1), sBOC(1,1)/PRN Squared Normalized Correlation Functions and the Difference of Both Squared Correlation Functions for 2 (Top Left), 3 (Top Right), 5 (Bottom Left), and 20 (Bottom Right) MHz One-Sided Front-End Filter .....	237
Figure 6.10 - ASPeCT Synthesized Correlation Function For Different Beta Values Using a 2 (Top Left), 3 (Top Right), 5 (Bottom Left), and 20 (Bottom Right) MHz One-Sided Front-End Filter.....	238
Figure 6.11 – Traditional and ASPeCT EMLP (Left) and DP (Right) Discriminator Output for using a 20 (Top) and 3 (Bottom) MHz One-Sided Front-End Filter for a 0.2 Chips Correlator Spacing .....	241
Figure 6.12 – ASPeCT EMLP (Left) and DP (Right) Discriminator Output for different values of Beta using a 3 MHz One-Sided Front-End Filter, and a 0.2 Chips Correlator Spacing.....	242
Figure 6.13 – Traditional and ASPeCT (Beta = 1 and Beta = 1.4) Normalized EMLP (Left) and DP (Right) Discriminator Output .....	244
Figure 6.14 – Code Delay Error for a Sine-BOC(1,1) Signal Tracking Starting with an Initial Code Delay Error of -0.5 Chips (2 Hz DLL), an Integration Time of 1 ms, and for a C/N <sub>0</sub> of 50 dB-Hz for ASPeCT DP (Beta = 1 and 1.4) .....	245
Figure 6.15 – ASPeCT-based DLL Architecture.....	246
Figure 6.16 – EMLP (Left) and DP (Right) Code Tracking Error Standard Deviation using ASPeCT (Beta = 1) and a 30 (Top) and 45 (Bottom) C/N <sub>0</sub> and a 4 ms Coherent Integration Time .....	250
Figure 6.17 - EMLP (Left) and DP (Right) Code Tracking Error Standard Deviation using ASPeCT (Beta = 1.4) and a 30 (Top) and 45 (Bottom) C/N <sub>0</sub> and a 4 ms Coherent Integration Time .....	251
Figure 6.18 – DP and EMLP Code Tracking Error Standard Deviation Degradation Brought from ASPeCT (Beta = 1) Compared to Traditional Tracking for a 30 (Top) and 45 (Bottom) C/N <sub>0</sub> and a 4 ms Coherent Integration Time .....	252
Figure 6.19 - DP and EMLP Code Tracking Error Standard Deviation Degradation Brought from ASPeCT (Beta = 1.4) Compared to Traditional Tracking for a 30 (Top) and 45 (Bottom) C/N <sub>0</sub> and a 4 ms Coherent Integration Time .....	253
Figure 6.20 – Code Tracking Error Standard Deviation for ASPeCT DP and EMLP Discriminators (Beta = 1 on Left and Beta = 1.4 on Right) using a 0.2 Chip Correlator Spacing, a 3 MHz one-sided front-end filter and a 4 ms Coherent Integration Time, and a 1 Hz Loop Bandwidth .....	254
Figure 6.21 – Code Tracking Error Standard Deviation vs. the C/N <sub>0</sub> for the Traditional Sine-BOC(n,n) Tracking and ASPeCT DP (Beta = 1 and	

	Beta = 1.4) with a Correlator Spacing of 0.2 Chips, a Coherent Integration Time of 4 ms, a 3 MHz Front-End Filter Bandwidth, and a 1 Hz Loop Bandwidth.....	255
Figure 6.22	– ASPeCT DP Code Tracking Error Standard Deviation Degradation against Traditional sBOC(1,1) Tracking (DP) vs. $C/N_0$ (Beta = 1 and Beta = 1.4) using a 3 MHz Front-End Filter Bandwidth, a 0.2 Chip Correlator Spacing ,a Coherent Integration Time of 4 ms .....	256
Figure 6.23	– ASPeCT DP implementation Theoretical and Simulated Code Tracking Error using a 4 ms Coherent Integration Time, a 0.2 Chips Correlator Spacing, a 3 MHz Front-End Filter, a 2 Hz Loop Bandwidth and a 0.4 Chips Discriminator Hard Limiter.....	257
Figure 6.24	– Code Tracking Multipath Envelope for Traditional sBOC(1,1), Traditional BPSK(1), and ASPeCT (Beta = 1 and Beta =1.4) for a 0.2 Chips Correlator Spacing, and a 3 MHz One-Sided Front-End Filter.....	259
Figure 6.25	– Synopsis of ASPeCT Acquisition Architecture .....	262
Figure 6.26	– Example of the Output of the Traditional and ASPeCT-based Acquisition Test Criteria for a 30 (Left) and 35 (Right) $C/N_0$ , a 4 ms Coherent integration, and 15 Non-Coherent Integrations (Wide (Top) and Close-up around the Main Peak (Bottom)).....	265
Figure 6.27	– Impact of the Coherent Integration Time and the $C/N_0$ on FOM1 for no (Left) and Maximum (Right) Code Delay/Doppler Uncertainty.....	267
Figure 6.28	– Probability of Detection for Proposed Galileo L1F Signal Assuming a Coherent Integration Time of 4 ms, 15 and 50 Non-Coherent Summations, no Code Delay/Doppler Uncertainty, and a Probability of False Alarm of $10^{-3}$ .....	268
Figure 6.29	– Probability of Detection for Proposed Galileo L1F Signal Assuming a Coherent Integration Time of 10, 20, 30 ms, No Non-Coherent Summations, no Code Delay/Doppler Uncertainty, and a Probability of False Alarm of $10^{-3}$ .....	269
Figure 6.30	- Probability of Detection for Proposed Galileo L1F Signal Assuming a Coherent Integration Time of 4 ms, 15 and 50 Non-Coherent Summations, Code Delay Uncertainty of 0.1 (Left) and 0.2 (Right) Chips, a Doppler Uncertainty of 50 Hz and a Probability of False Alarm of $10^{-3}$ .....	271

# List of Symbols and Abbreviations

## Acronyms

ABAS	Airborne-Based Augmentation System
ADC	Analog-to-Digital Converter
ASPeCT	Autocorrelation Side-Peak Cancellation Technique
BER	Bit Error Rate
BJ	Bump and Jump Technique
BOC	Binary Offset Carrier
BPSK	Binary Phase Shift Keying
C/A	Coarse/Acquisition
CASM	Coherent Adaptive Sub-carrier Modulation
CDDIS	Crustal Dynamics Data Information System institute
CDMA	Code Division Multiple Access
C/N <sub>0</sub>	Carrier-to-Noise PSD ratio
DLL	Delay Lock Loop
DoD	US Department of Defence
DoT	US Department of Transportation
DS-SS	Direct Sequence - Spread Spectrum
EGNOS	European Geostationary Navigational Overlay Service
ESA	European Space Agency
EU	European Union
FEC	Forward Error Correction
FLL	Frequency Lock Loop
FTS	Frequency and Time Standard
GBAS	Ground-Based Augmentation System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSTB	Galileo System Test Bed
HPA	High Performance Amplifier
HRC	High Resolution Correlator code tracking technique
I&D	Integrate and Dump filter
IF	Intermediate Frequency
L1	L1 frequency band centred in 1575.42 MHz
L1-B	Galileo L1F data channel
L1-C	Galileo L1F pilot channel
L1-F	Galileo OS signal on L1
L1-P	Galileo PRS signal on L1

L2	L1 frequency band centred in 1227.6 MHz
LAAS	Local Area Augmentation System
LHCP	Left Hand Circularly Polarized
LNA	Low Noise Amplifier
LOS	Line-Of-Sight
MEO	Medium Elevation Orbit
NRZ	Non-Return to Zero
OCXO	Oven Controlled Crystal Oscillator
OS	Galileo Open Service
P	Precise
PDA	Personal Digital Assistant
PLL	Phase Lock Loop
PRN	Pseudo-Random Noise
PRS	Galileo Public Regulated Service
PSD	Power Spectrum Density
PSK	Phase Shift Keying
QPSK	Quadrature Phase Shift Keying
RHCP	Right Hand Circularly Polarized
SA	Selective Availability
SBAS	Satellite-Based Augmentation System
sBOC	Sine-Phased BOC
SNR	Signal-to-Noise Ratio
SSL	Single Side-Lobe BOC tracking technique
TCXO	Temperature Compensated Crystal Oscillator
WAAS	Wide Area Augmentation System

## Symbols

$R$	Autocorrelation function
$\tilde{R}$	Correlation of the local code with the filtered incoming code
$f_c$	Spreading code rate
$T_c$	Spreading code chip duration
$P$	Signal power
$A$	Signal amplitude
$\tau$	Code group delay
$\hat{\tau}$	Estimated code group delay made by the receiver
$\varepsilon_\tau$	Estimated code delay error
$\phi$	Carrier phase delay
$\hat{\phi}$	Estimated code group delay made by the receiver
$\varepsilon_\phi$	Estimated phase delay error
$\varepsilon_f$	Estimated frequency (Doppler) error
$d$	Binary NRZ materialization of the navigation message

$D$	Sign of the navigation data bit
$c$	Binary NRZ materialization of the spreading code
$f_{L_1}$	L1 carrier frequency (1575.42 MHz)
$t_{GPS}$	GPS reference time
$t_{Sat}$	Satellite oscillator time
$t_{Rx}$	Receiver oscillator time
$\delta t_{Rx}$	Receiver oscillator timing error with respect to GPS time
$\rho$	True satellite – receiver range
$c_l$	Speed of light (299792458 m/s)
$\Delta t_{TC_c}$	Code group delay due to propagation in the atmosphere
$\Delta t_{TC_\phi}$	Carrier phase delay due to propagation in the atmosphere
$\lambda_{L_1}$	L1 carrier wavelength (~19.03 cm)
$f_{Dop}$	Doppler frequency
$T_I$	Coherent integration time
$B_{ID}$	One-sided bandwidth of the Integrate and Dump filter
$LR_I$	Receiver in-phase local replica
$LR_Q$	Receiver quadrature-phase local replica
$I$	In-phase correlation value
$Q$	Quadrature-phase correlation value
$n$	Thermal noise at the receiver antenna level (assumed white and Gaussian)
$N$	PSD of the incoming thermal noise $n$
$N_0$	Constant noise PSD of the thermal noise
$n_I$	In-phase correlation noise (assumed Gaussian)
$P_{n_I}$	Power of the in-phase correlation noise
$n_Q$	Quadrature-phase correlation noise (assumed Gaussian)
$P_{n_Q}$	Power of the in-phase correlation noise
$h$	Front-end filter
$H$	Fourier transform of the front-end filter $h$
$B$	One-sided bandwidth of the front-end filter
$(\tilde{\cdot})$	Filtering operation by the receiver front-end filter
$M$	Number of non-coherent summations
$SL$	Correlation squaring loss
$SNR_{post}$	Post-correlation signal-to-noise-ratio
$SNR_{pre}$	Pre-correlation signal-to-noise-ratio
$CG$	Correlation gain
$\sigma_{LB}$	Cramer-Rao lower bound tracking standard deviation
$\beta_{RMS}$	RMS bandwidth

$\beta_{rect}$	Effective rectangular bandwidth
$f_{max}$	frequency at which the signal PSD is maximum
$\kappa_{S_1 S_2}$	Spectral coefficient separation between signals $S_1$ and $S_2$
$B_L$	DLL filter one-sided bandwidth
$\frac{C}{N_0}$	Carrier-to-noise PSD ratio
$G$	Signal PSD
$\overline{G}$	Signal PSD normalized over $\pm B$
SC	Sub-carrier
$tri\left(\frac{x}{y}\right)$	Triangular function of width $2y$ , centered in $x = 0$ where it has a unity value
TH	Binary NRZ materialization of the Galileo secondary code
$h_{-2}$	Random walk component of the oscillator frequency noise
$h_{-1}$	Flicker component of the oscillator frequency noise
$h_0$	White components of the oscillator frequency noise
$\sigma_A^2$	Allan variance
$\Delta$	True time interval
$K_B$	Boltzmann constant (= -228.6 dBW/K/Hz)
$T_{sys}$	System noise temperature
$f_{IF}$	IF frequency
$\varphi_0$	Initial carrier phase
MP	Multipath
$\delta t_{Iono}^{L1}$	Delay due to ionosphere on the L1 frequency
$\delta t_{Tropo}$	Delay due to troposphere
$\delta t_{Sat}$	Satellite clock error
$\delta t_{eph}$	Ephemeris error
$t_p$	Propagation delay
$P_0$	Average maximum power received from near echoes
$\delta d$	Near echo delay
$\alpha$	Decay of near echo power with delay
$a_e$	Near echo amplitude
$h_{PLL}$	PLL equivalent loop filter
$h_{DLL}$	DLL equivalent loop filter
$\alpha$	Absolute value of the slope of the spreading sequence autocorrelation main peak

## CHAPTER 1

### INTRODUCTION

The *Global Positioning System* (GPS) has been extensively studied and used by specialized communities dealing with positioning, navigation and timing. The GPS's ability to provide relatively accurate positioning under all weather conditions has contributed to its widespread use as a professional tool (i.e. civil aviation, surveying, etc.) as well as for leisure activities (i.e. car navigation, pedestrian navigation, sailing, etc...). In recent years, the miniaturization of electronic components and improvements in power management have allowed the integration of GPS chips into small autonomous devices such as handheld GPS receivers, *Personal Digital Assistants* (PDAs), and cellular phones, increasing the speed of its assimilation by the general public. In order to capitalize on this massive rising demand, and to cope with civil and military expectations in terms of performance, several projects were launched to give birth to a second generation of *Global Navigation Satellite Systems* (GNSSs) in the 1990's (Kovach & Van Dyke, 1997). This effort led to two major GNSS decisions: the modernization of the current US GPS (known as GPS I), and the independent European effort to create its own GNSS, known as Galileo. These two systems are now being finalized and are expected to be available to the public by the end of the decade.

## 1.1 Background and Motivation

The GNSS modernization process began with the development of augmentation systems to complement GPS I. The function of these external systems is to provide additional information in order to reduce the uncertainties in the final positioning or navigation solutions while improving availability and reliability. They are essentially meant for specialized applications such as civil aviation or marine applications, and can be categorized as *Satellite-Based Augmentation Systems (SBAS)*, *Ground-Based Augmentation Systems (GBAS)* or *Airborne-Based Augmentation Systems (ABAS)*. The *US Wide Area Augmentation System (WAAS)* (Enge & Van Dierendonck 1997), *Local Area Augmentation System (LAAS)* (Bradley *et al.* 2000), and the *European Geostationary Navigational Overlay Service (EGNOS)* (Toran-Marti & Ventura-Traveset 2004) are well-known examples of such systems complementing GPS. Although effective, they are very expensive to build and the overall performance is constrained by inherent factors limiting the capability of GPS I (Kovach & Van Dyke 1997).

Consequently, the experience gained during the GPS I design and exploitation initiated a desire to enhance the US-operated GPS and improve overall system performance. The early success encountered by GPS combined with a substantial potential for growth in positioning and timing applications in the civilian market (Onidi 2001) are the major reasons for a world-wide modernization effort. In the 1990's, The US *Department of Defence (DoD)* and *Department of Transportation (DoT)* started a GPS modernization process, called GPS II and GPS III (Kovach & Van Dyke 1997; US Office of Science and Technology Policy 2004). The *European Union (EU)* and the *European Space Agency*

(ESA) decided to launch their own GNSS, known as Galileo (European Commission 1999; PriceWaterHouseCoopers 2001). Finally, Russia decided to re-activate its GLONASS program (GPS World 2005).

GPS II and Galileo are currently of particular interest owing to their pending availability, and because their launch will see the realization of a large number of modifications intended to address the main weaknesses of GPS I, as well as to enhance its overall performance. These improvements have been obtained through changes at all levels of the respective navigation systems: the space segment, the user segment, and the ground segment. For example, the number of signals to be transmitted by each GNSS satellite has significantly increased, as compared to GPS I. Indeed, GPS II and Galileo satellites will each broadcast three civilian signals (Lachapelle 2004a) designed for frequency diversity and ionosphere estimation purposes. This means that any GNSS user can have free access to a minimum of six different signals, but also that the receivers will have to be adapted to these changes.

Of all the proposed signals, the Galileo L1F signal is of particular interest for several reasons:

- It is designed for mass-market users. A mass-market signal means that every Galileo user will have access to this signal, and that it should be the target signal for most of the autonomous and leisure-oriented applications such as cellular phones and PDAs, implying a very large potential market. Consequently, it is Galileo's direct counterpart to the current GPS I civil signal. In order to appeal to a wide-range of civilian uses, and personal navigation in particular, the signal

design should be such that it does not require an immense amount of processing power for tracking while still providing optimal positioning capabilities in degraded signal environments. The significance of these factors in terms of system cost have spurred much research into the receiver design adaptations necessary to take full advantage of this signal.

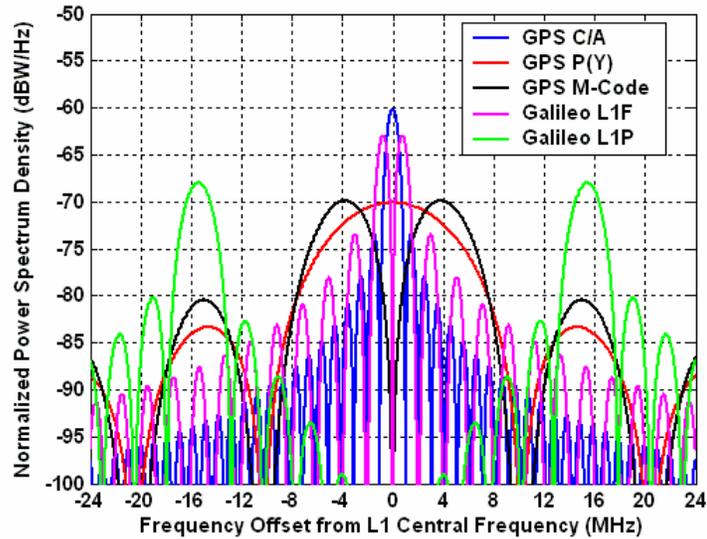
- It is important to identify and assess any improvement for mass-market users brought about by the use of the Galileo L1F signal compared to what is currently available to them through the GPS C/A signal.
- The Galileo L1F signal uses new modulations that are also used by other GPS II and Galileo signals (Hein *et al.* 2002). Consequently, knowledge gained from research and analysis with this particular signal can be transposed to amelioration of other signal profiles.

As previously mentioned, the Galileo L1F signal presents many structural innovations in its design. In order to quantify how these changes will affect the users, it seems natural to analyze their effects on the three types of measurements realized by a GNSS receiver: pseudorange, carrier-phase and Doppler. Since standalone GNSS positioning performance is partially based on the accuracy of its measurements, seeking ways to minimize the magnitude of the associated errors is extremely worthwhile. Indeed, the design and study of future signals also calls for researching innovative solutions that would solve inherent system design problems, or that would allow the systems' performance to surpass initial expectations. Thus, a thorough study of the architecture of

the two main tracking loops, namely the *Delay Lock Loop* (DLL) and the *Phase Lock Loop* (PLL), becomes particularly valuable.

Two innovations in the Galileo L1F signal are of central interest for tracking performance as they lead to substantial improvement in tracking: (1) the use of a *Binary Offset Carrier* (BOC) modulation; and (2) the presence of a dataless (or pilot) channel.

GPS I signals employ *Phase Shift Keying* (PSK) modulation. By comparison, BOC modulation was chosen as the chief candidate for several future navigation signals for several purposes. Its split spectrum property allows the reduction of spectral overlapping with other PSK signals and, thus, lessens the potential for interference with the GPS legacy signals that have their energy around the carrier frequency. Another implication for non-overlapping signals is the possibility for the US military to jam civil signals without losing the military signals. Of the several BOC families available, a relevant choice of this modulation can drastically limit inter- and intra-system interference (Betz 2002; Betz & Goldenstein 2002). Figure 1.1 shows the current expected spectral occupancy of the L1 frequency band. The GPS legacy signals, the GPS C/A and P(Y), employ PSK modulation; GPS M-code, Galileo L1F and Galileo L1P signals use the BOC modulation. It seems clear that the spectral overlap with PSK signals is reduced significantly through the use of BOC signals. Moreover, BOC modulation has the very interesting tracking property of outperforming an equivalent *Binary Phase Shift Keying* (BPSK) modulation in respect of resistance to thermal noise, narrow-band interference rejection, and multipath mitigation (Betz 2002).



**Figure 1.1 - GPS and Galileo Expected Spectral Occupation of the L1 Frequency Band**

However, despite these advantages, some problems remain with the use of BOC modulation. BOC signal tracking shows potential false tracking lock points (Betz 2002, Julien *et al.* 2004c). This stems from the BOC autocorrelation function, which is characterized by multiple side-peaks with non-negligible magnitude. Undergoing a false lock produces biased measurements and, hence, an unacceptable result for a system that aims to provide an accurate navigation solution. Consequently, solutions have to be found to minimize this bias threat in order to be able to use BOC signals in a GNSS. Many studies on the BOC bias (or ambiguity) tracking problem have been published, e.g. Fine & Wilson (1999), Martin *et al.* (2003), Lin *et al.* (2003), and Ward (2004). However, these methods either have reliability issues, or they significantly degrade tracking performance. In addition, they all try to resolve the tracking ambiguity problem in the same fashion for all BOC families. In reality, however, there exist different BOC families with characteristics defined by the spectral separation and width of the side-lobes. The

GPS M-code and Galileo L1P signals in Figure 1.1 are examples of two different BOC families. Although the families may be derived in a similar way, a single resolution to the ambiguity tracking problem may not be optimal for all signals. An unambiguous tracking technique dedicated to a certain BOC family might provide a more effective solution in the same way that some tracking techniques are optimal for BPSK modulations but not for other types of modulation.

The second innovation of interest in this thesis regarding the Galileo L1F signal structure is the presence of a dataless channel in quadrature-phase with a conventional data channel that carries the navigation data message (Hein *et al.* 2004). In the GPS legacy signals, only a data channel is present, which limits the integration time on the correlation process in a typical standalone GPS receiver (it should be mentioned that *Aided GPS* (AGPS) receivers were developed and designed to strip off the navigation bits based on an auxiliary communication channel that delivers the navigation message, but at additional financial cost). This leads to two main drawbacks: (1) it limits the mitigation of thermal noise on the correlation values; (2) it significantly degrades carrier-phase tracking performance due to their necessity to be insensitive to data bit transitions, which are equivalent to a  $180^\circ$  phase shift, as explained by Ward (1996) and Van Dierendonck (1997). Consequently, the presence of a pilot channel implies a potentially great benefit in terms of tracking performance for standalone GNSS receivers. It will clearly have a significant impact on the code delay, carrier phase and/or frequency tracking loops due to the use of cleaner correlation values. Not only does this translate to an improvement in accuracy, but also to increased tracking sensitivity and robustness.

It is understandable that being able to track a very low-powered signal is an asset for a GNSS, and even more so when it is intended to support mass-market applications. For GPS I, which was meant to work in a ‘clear sky’-type of environment, this is a major drawback. The efficient use of a pilot channel in low-signal-power locations, (i.e. where obstacles impede the GNSS signals’ *Line Of Sight* (LOS)) to improve tracking is a challenging problem. One must first understand all of the parameters involved in the use of long coherent integrations. Design of the receiver tracking loops is incomplete without an assessment of its tracking capability. In addition to providing better tracking performance, the presence of the data and pilot channels in quadrature creates an opportunity to combine them in attempts to further improve the accuracy of the resulting measurements (Hegarty 1999).

These two innovations in the Galileo L1F signal design constitute the core of this thesis. Seeking an understanding of their importance and the search for an optimal way of using them was the motivating factor for this work.

## **1.2 Thesis Objectives**

The aim of this thesis is the conception of Galileo L1F receiver tracking loops which incorporate the two innovations discussed in Section 1.1 above. The task of reaching this objective has been divided into the following research goals:

1. **To understand all the important parameters in the design of the various tracking loops.** The DLL and PLL are usually interconnected; however, an independent study of each in turn allows a good appreciation of:
  - The impact of the main sources of error; and
  - The critical parameters and their effect on the tracking performance.
  
2. **To assess and quantify the advantage of the new Galileo L1F signal modulation on the receiver tracking loops.** This goal focuses on quantifying how the BOC modulation and the pilot channel might improve the tracking process using traditional tracking architectures. This work will be based on the understanding gained in the first goal. It aims at demonstrating the advantages of the Galileo L1F signal over the only public signal currently available, the GPS I legacy civil signal.
  
3. **To find an innovative and close-to-optimal solution to the BOC tracking bias threat.** The Galileo L1F signal uses a particular BOC modulation family, also known as a Manchester-encoded ranging signal. The purpose here is to design a DLL that will provide a measurement accuracy close to traditional BOC tracking (and thus better than GPS C/A tracking), while completely removing the tracking bias threat. Since the signal of interest is the Galileo L1F, the solution presented will be relevant to this particular signal, even if it does not work for all BOC families. To find an efficient solution, a new technique was developed according to several constraints:
  - A tracking architecture close to current tracking techniques in order to be easily incorporated in any receiver;
  - A total reliability (no false lock possible); and

- A close-to-optimal tracking accuracy.

**4. To find an optimal way to use the information from the data and pilot channels.**

The research conducted to fulfill the second goal provides an excellent overview of the advantage of pilot channel tracking over data pilot tracking. The potential exists to achieve even better tracking performance by combining both channels. Prospective data/pilot tracking algorithms will be studied and compared for PLL implementations.

**5. To test the new algorithms using a Galileo L1F software receiver and signal simulator.**

It is important to confirm, through simulations, the results obtained theoretically in this thesis. Since no Galileo signals are currently being transmitted, it was necessary to build simulation tools. Recent advances in computer speed made it possible to modify existing software packages, or to create new ones, in order to realize relevant and versatile simulation tools. This goal was to develop two complementary software tools:

- **A software Galileo L1F signal generator.** It is important in this case to model as accurately as possible the signal transmission channel in order to produce a signal reflecting the ‘real world’; and
- **A set of software Galileo L1F receiver tracking loops.** As this thesis is limited to a study of tracking performance, only the tracking loops were implemented for data post-processing.

The implementation of these two complementary software tools allows the testing of both the traditional tracking techniques, as well as the newly developed methods for a fair and complete comparison.

The organization of the thesis is determined by the inter-relationship of these five goals, as it will be outlined in Section 1.4.

### **1.3 Thesis Contributions**

The major contributions of this thesis are enumerated as follow:

- A detailed analysis of the critical parameters involved in the design of a PLL and a DLL,
- A detailed analysis of the benefits brought by the availability of a pilot channel for code and carrier phase tracking,
- The design and test of a new carrier phase tracking technique for noise mitigation that combines the data and pilot channels,
- The design and test of a new code tracking technique for multipath mitigation based on the use of the pilot channel only,
- The design and test of a new code tracking technique that cancels the ambiguity threat associated with Galileo L1F modulation while maintaining excellent tracking performances,
- The design of a new acquisition technique that cancels the risks of ambiguous acquisition due to Galileo L1F spreading sequence autocorrelation side-peaks.

## 1.4 Thesis Outline

Since this thesis focuses on the tracking loops of a Galileo L1 civil receiver, and in order to provide a comprehensive view of the research realized, this document has been structured in the following way.

Following a survey of the field and an introduction to its semantics and the scope of the thesis in Chapter 1, Chapter 2 gives an overview of the innovations brought by the Galileo L1F as compared to the GPS C/A signal. It starts with a brief overview of the current GPS mass-market signal, and explains its main limitations with respect to current civil expectations in terms of positioning and accuracy. It then describes the main innovations brought by Galileo L1F over GPS I to have a solid understanding of the future Galileo mass-market signal. Finally, a detailed description of the Galileo L1F model used in this thesis is presented.

Chapter 3 describes the software tools used throughout this thesis to test algorithms and techniques. The main GNSS sources of error are first explained. The Galileo L1F signal software generator is then described with detailed explanations of the models implemented for the various error sources. The tracking loops and their different stages are then briefly outlined.

Chapter 4 consists of an in-depth investigation of the Galileo L1F receiver PLL. The general architecture of a PLL and its main sources of error are first described. The difference between the PLL architectures for data and pilot channels is emphasized. The improvement in terms of sensitivity realized through use of the pilot channel is quantified

in comparison to the use of the data channel alone. Finally, several phase tracking architectures using both the data and pilot channels for enhanced noise mitigation are proposed and studied.

Chapter 5 is a thorough study of the Galileo L1F receiver DLL. Beginning with a description of the general architecture of a typical DLL and its main sources of error, the emphasis is placed on the benefits brought by BOC signal tracking, as compared to BPSK signal tracking methods employed in the current GPS civil signal. The impact of the pilot channel on code delay tracking is also considered. A new robust multipath mitigation technique, using the pilot channel only, and based on the complementarities of two different code tracking techniques is finally proposed.

Chapter 6 introduces the code tracking ambiguity threat inherent to BOC tracking. It also introduces two existing techniques designed to cope with this problem, also illustrating their weaknesses. An innovative bias-free BOC tracking technique referred to as the *Autocorrelation Side-Peak Cancellation Technique* (ASPeCT) is then introduced. This new technique is first described in great theoretical detail. Its critical parameters and resistance to the main sources of error are thoroughly investigated, followed by testing with the aid of simulations. A possible implementation, derived from ASPeCT, for unbiased acquisition schemes (corresponding to an acquisition on the main Galileo L1F autocorrelation peak by opposition to potential acquisition on its inherent side-peaks) is also considered.

Finally, Chapter 7 draws conclusions from this research and makes recommendations for future work.

## CHAPTER 2

### CURRENT GPS C/A SIGNAL STRUCTURE AND INNOVATIONS BROUGHT BY GALILEO L1F SIGNAL

This second chapter first presents an overview of the only civil GNSS signal currently available to public users, namely the GPS *Coarse/Acquisition* (C/A) signal. Since this thesis focuses on signal tracking, only the signal structure and its main characteristics are presented; more thorough discussions of general GPS operations and applications can be found in Ward (1996) and Misra & Enge (2001). Because the GPS C/A signal was not originally intended to be a precise mass-market signal, and since it is a pioneer signal, the current intensity of product development and increasing user demand present a wide range of challenges. These limitations are discussed in this thesis from a pure tracking point of view since, as mentioned in Chapter 1, the focus is put on the measurement domain. The proposed Galileo mass-market signal on the L1 band, Galileo L1F, is then described. It shows how this signal was designed to take advantage of the knowledge learned from the GPS C/A signal, and why it is expected to provide better tracking measurements than what is currently available. Finally, the Galileo L1F signal model examined in the rest of this thesis is introduced.

## 2.1 GPS C/A Signal Structure

In terms of overall contribution to the success of GPS and its navigation capability, the GPS C/A signal has made GPS what it is today: a field in complete technological and organizational transformation to meet an ever-increasing public and private demand. However, the GPS C/A code was not originally meant for the widespread penetration and public success it now encounters.

### 2.1.1 Brief History

The development of GPS started in 1973 as a navigation system project dedicated to support military applications (Parkinson 1997). In essence, the GPS I system was designed around the idea of measuring the distance between the user and *Medium Elevation Orbit* (MEO) satellites with known locations. Multi-lateration is then used to retrieve the user position. From a signal point of view, GPS was designed to fulfill several military requirements summarized by Spilker (1997a):

1. Tolerance to signals from other GPS satellites sharing the same frequency band; i.e. multiple access capability;
2. Tolerance to a certain level of multipath interference;
3. Tolerance to reasonable levels of unintentional or intentional interference jamming, or spoofing by a signal designed to mimic the GPS signal; and
4. Signal sufficiently low in power in order to avoid interference with terrestrial microwave line-of-sight communication.

To fulfill these requirements, GPS I signals use a *Direct Sequence Spread Spectrum* (DS-SS) technique, and are based on *Code Division Multiple Access* (CDMA) principles to distinguish signals coming from different satellites (Peterson *et al.* 1995; Spilker 1997b). GPS I signals use a PSK modulation with rectangular *Non-Return to Zero* (NRZ) spreading codes. This permits a simple signal structure that also possesses a constant envelope and, hence, is well suited to the *High Performance Amplifiers* (HPA) used by GPS satellites to transmit the signals.

Three signals were designed to meet the specifications of the system. Two high performance signals, with long spreading codes and a fast code rate were designed. These codes are referred to as *Precise* (P) spreading codes. They are broadcast at two different carrier frequencies: L1 (1575.42 MHz) and L2 (1227.6 MHz) to offer frequency diversity (to counter possible signal jamming) and ionosphere modeling (since the ionosphere is dispersive, as will be explained in Section 3.1.3). These precise codes are encrypted by a classified Y code, and, as such, are restricted to authorized users.

A third signal, with lower performance, was transmitted on the L1 frequency only and was left unencrypted. It was intended to be used mainly for fast acquisition purposes since its spreading code length is much shorter and its spreading code rate ten times slower than that of the P(Y) code. Following fast acquisition using this signal, military users could use the precise signals, hence the origin of its name, C/A code for Coarse/Acquisition. It was left freely available to civilian users; however, a voluntary degradation of the satellite broadcast ephemeris, referred to as *Selective Availability* (SA) was instituted to reduce the positioning accuracy obtained by civil receivers to a

horizontal positioning accuracy of 75 metres (95%). Authorized users could overcome SA and obtain optimal system accuracy (Lachapelle 2004a). This had the effect of limiting civilian interest for GPS utilization. However, innovative techniques, such as measurement double-differencing, reduced the effectiveness of SA, and gradually made GPS one of the main positioning and navigation tools for an increasing number of applications. Finally, a US presidential decision was made in May 2000 to turn SA off (Clinton 2000), making GPS a tremendously interesting civil navigation and timing tool with horizontal positioning accuracy at the metre level or better in differential mode. The design of the GPS C/A signal has had a great impact on the current notoriety of GNSS. A thorough description of this structure is given in the following section.

### **2.1.2 Signal Structure**

The GPS C/A signal is *Right Hand Circular Polarized* (RHCP) and uses a BPSK modulation. The carrier, at L1, is modulated by a navigation message and a *Pseudo-Random Noise* (PRN) code, both using rectangular NRZ symbols. The carrier wavelength is approximately 19 cm.

The navigation message contains all of the information relevant for precise positioning and timing as described by Spilker (1997c), and has a data rate of 50 Hz. This relatively low rate compared to typical communication devices allows an acceptable *Bit Error Rate* (BER) for common *Signal-to-Noise Ratios* (SNR) (Van Dierendonck 1997).

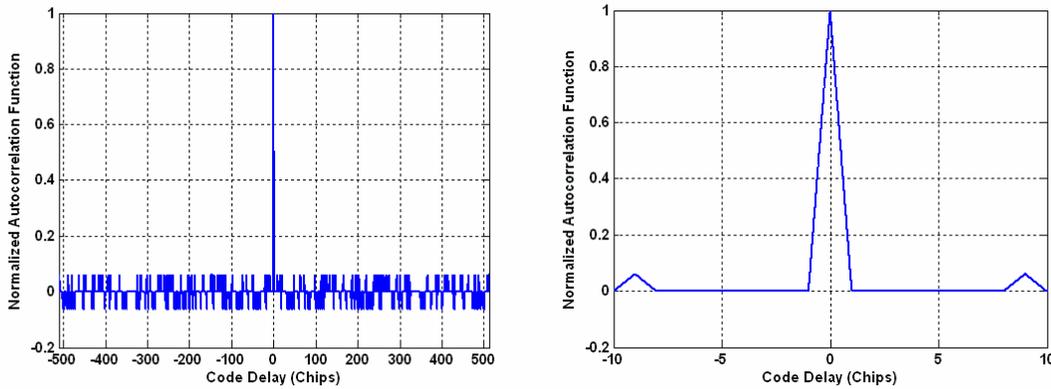
The role of the PRN code is to spread the signal over a wide frequency bandwidth. The GPS C/A signal uses part of a 1023-bit Gold code family described in Gold (1967) and

Spilker (1997b). This code family was chosen for its good correlation properties, which provide a minimum isolation of 21.6 dB between the autocorrelation main peak and the autocorrelation and cross-correlation (with codes from other satellites) side peaks (Spilker 1997b). The spreading code rate used is 1.023 MHz, i.e. more than 20,000 times faster than the data bit train. To simplify the notations used herein, a BPSK( $n$ ) will refer to a BPSK modulation with a PRN code rate equal to  $n \times 1.023$  MHz. Consequently, GPS C/A modulation will be referred to as BPSK(1). An example of a typical normalized Gold code autocorrelation function used by GPS C/A signal is given in Figure 2.1.

The central part of the C/A code autocorrelation function is of major importance since it significantly conditions the tracking process, as explained in Spilker (1997b) and studied in Chapters 5 herein. When modeling the GPS C/A code, it is usually assumed that the spreading code chips are independent. This implies that the code autocorrelation function can be approximated by the autocorrelation of a spreading code chip, as explained in Holmes (1982). This approximation of the autocorrelation function is a perfect triangle and is given by:

$$R_P(x) = \begin{cases} 1 - |x| & \text{if } |x| \leq 1 \text{ chip} \\ 0 & \text{if } |x| > 1 \text{ chip} \end{cases} \quad (2.1)$$

Figure 2.1 shows that it is a good approximation. However, when designing a receiver, due to the potential presence of side-peaks that can be located at  $\pm 1$  chips, the potential deviation of certain C/A code autocorrelation functions from this approximation must be considered (Simsy & Sleewaegen 2004).



**Figure 2.1 – Example of GPS C/A Code (Left) Normalized Autocorrelation Function, and (Right) an exploded view about  $\pm 10$  Chips**

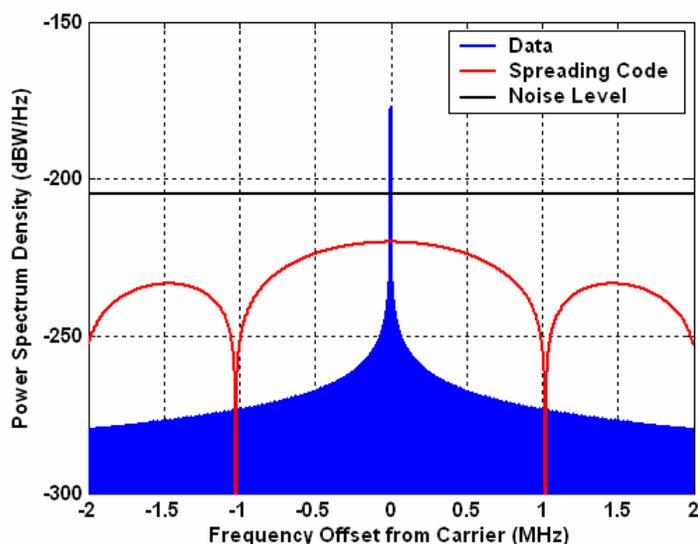
The *Power Spectral Density* (PSD) of the GPS C/A signal is a peak spectrum. Its derivation is given by Macabiau (2001). The presence of the PRN spreading code creates this peak spectrum which envelope is a sinc function corresponding to the PSD of a spreading code chip symbol. Each of the PSD spectral lines is separated by an increment equal to the ratio between the chip rate and the spreading code length, or 1 KHz in the case of the GPS C/A signal. However, in order to facilitate analysis herein, the signal PSD is usually modeled as its PSD envelope. Consequently, the normalized GPS C/A PSD is usually modeled as:

$$G_{C/A}(f) = T_c \left( \frac{\sin\left(\pi \frac{f}{f_c}\right)}{\pi \frac{f}{f_c}} \right)^2 \quad (2.2)$$

where  $f_c$  is the spreading code rate, and

$$T_c = \frac{1}{f_c} \text{ is the spreading code chip duration.}$$

In order not to interfere with existing communication systems, the power level of the GPS signal was set very low. The specified minimum received power, using a 0 dBic antenna with RHCP, is -160 dBW (Spilker 1997b), although the actual received power seems to be around -154 dBW and -157 dBW according to Hudnut *et al.* (2004). In addition to this characteristically low signal power, the spreading of the data bits over a very large spectral bandwidth brings the signal PSD under the usual thermal noise PSD level, which can be approximated at about -205 dBW/Hz under normal conditions (Van Dierendonck 1997) (see Figure 2.2). This means that GPS C/A signals cannot be detected, searching the L1 band, using classical tools such as an oscilloscope. It also implies that they should not interfere with terrestrial communication systems.



**Figure 2.2 – Power Spectral Densities of the Navigation Data, GPS C/A Spreading Code, and Typical Thermal Noise Assuming a Signal Power of -160 dBW**

From the above description, and taking into account the propagation time, the GPS C/A signal, coming from a particular satellite, can then be modeled at the receiver antenna level as:

$$s_{C/A}(t) = \sqrt{2P_{C/A}} d(t - \tau_{C/A}(t)) c_{C/A}(t - \tau_{C/A}(t)) \cos(2\pi f_{L_1} t + \phi_{C/A}(t)) \quad (2.3)$$

where  $P_{C/A}$  is the power of the received GPS C/A signal with  $P_{C/A} = \frac{A_{C/A}^2}{2}$ ,

$A_{C/A}$  is the amplitude of the received GPS C/A signal,

$d_{C/A}$  is the binary NRZ materialization of the data bit train,

$c_{C/A}$  is the binary NRZ materialization of the C/A code,

$\tau_{C/A}$  is the time-varying code group delay due to the propagation (in seconds),

$\phi_{C/A}$  is the time-varying carrier phase delay due to the propagation (in radians), and

$f_{L_1}$  is the L1 carrier frequency.

Since GPS positioning is based on the measurement of the satellite – receiver propagation time, it is important to define each term in Equation (2.3) with respect to a common reference time. GPS time ( $t_{GPS}$ ) is defined herein as the GPS reference time. It is the time with respect to which the actual propagation time should be calculated in order to unify the measurements and, thereby, produce a relevant position solution. However, the satellites and the receiver have their own *Frequency and Time Standards* (FTS) as manifested by their respective oscillators, which provide  $t_{Sat}$  and  $t_{Rx}$  respectively. The tracking loops' measurements are made inside the receiver and, consequently, they are made with respect to the receiver time. Accordingly, in the frame of this thesis, the time of interest is the receiver time, and  $t = t_{Rx}$  in Equation (2.3), unless otherwise stated.

The code and carrier phase delay can be decomposed as:

$$\tau_{C/A} = \frac{\rho}{c_l} + (t_{Rx} - t_{Sat}) + \Delta t_{TC_c} \quad (2.4)$$

and

$$\phi_{C/A} = 2\pi f_{L_1} \left( \frac{\rho}{c_l} + (t_{Rx} - t_{Sat}) + \Delta t_{TC_p} \right) \quad (2.5)$$

where  $\rho$  is the true distance between the satellite and the receiver antennas,

$c_l$  is the speed of light,

$\Delta t_{TC_c}$  represents the part of the code group delay due to propagation in the atmosphere, and

$\Delta t_{TC_p}$  represents the part of the carrier phase delay due to propagation in the atmosphere.

It must be noted from Equations (2.3), (2.4) and (2.5) that the code and carrier phase delays are differentiated. This is mainly due to the fact that the ionosphere, as will be seen in Section 3.1.3., has an opposite effect on  $\Delta t_{TC_c}$  and  $\Delta t_{TC_p}$ . Indeed, signal propagation through the ionosphere induces a phase advance and a code delay of the same magnitude (Klobuchar 1997). It is then necessary to differentiate them.

It is also important to understand that the phase group delay  $\phi_{C/A}$  incorporates the Doppler effect due to the signal propagation time variation between the satellite and receiver. This Doppler effect can thus be modeled as:

$$f_{Dop} = \frac{1}{2\pi} \frac{d\phi_{C/A}}{dt} \quad (2.6)$$

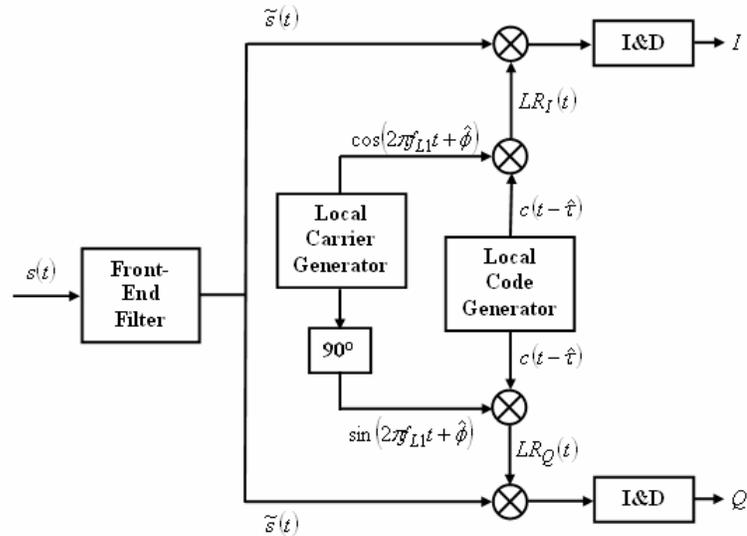
Finally, when it comes to assessing the performance of the tracking loops, one has to understand that this performance is measured by comparing the code and phase delays tracked by the receiver and the actual code and phase delays at the entrance of the tracking loops. Consequently, a constant (or slowly varying compared to the tracking loop response time) bias inherent to the propagation channel, for instance - although considered as an error in the final position solution - cannot be considered as an error made by the tracking loops.

Now that a detailed description of the GPS C/A signal structure has been given, it is instructive, in order to understand its performance, to study the first step involved in signal processing in a conventional receiver: the correlation of the incoming signal with its locally-generated replica. This technique and its main properties are explained in the next section.

### **2.1.3 Correlation Properties**

The correlation process is based on the multiplication of a locally-generated replica of the received PRN code with the received signal. Three factors have a significant impact on this correlation process: (1) the presence of external disturbances (i.e. thermal noise, multipath, etc...); (2) the presence of the carrier; and (3) the front-end filter that prevents the receiver from receiving the full signal spectrum. The correlation process in a GPS receiver is schematized in Figure 2.3, although not taking into account the *Analog-to-*

*Digital Converter* (ADC) effect. The notations in Figure 2.3 follow the notations used later on for the correlation value computation. For the sake of clarity, the reference to the C/A code through the subscript “C/A” is abandoned; the notations can be understood as the representation of any signal following the classical DS-SS CDMA model.



**Figure 2.3 – Schematic Representation of the Correlation Process in a GNSS Receiver**

The thermal noise will be considered, in this section, as the only external disturbance. This is motivated by the fact that, unlike other disturbances such as multipath (that will be studied in Chapter 5) or interference, it is a phenomenon always present. Thermal noise is usually represented as a white Gaussian noise  $n$  with a constant PSD equal to

$\frac{N_0}{2}$  dBW/Hz. More details are given in Section 3.1.6. The received signal at the antenna

level can then be modeled as:

$$s(t) = s_{\text{useful}}(t) + n(t) \quad (2.7)$$

In a traditional GPS receiver, the incoming signal is first filtered and down-converted to an *Intermediate Frequency* (IF) by the front-end filter (leading to  $\tilde{s}$ ). Following Figure 2.3, the GPS receiver first generates an in-phase and quadra-phase local replica of the incoming signal that can be modeled as:

$$LR_Q(t) = c(t - \hat{\tau}) \cos(2\pi f_{IF} t + \hat{\phi}) \quad (2.8)$$

$$LR_Q(t) = c(t - \hat{\tau}) \sin(2\pi f_{IF} t + \hat{\phi}) \quad (2.9)$$

where  $\hat{\tau}$  is the receiver's estimation of the incoming signal group delay, and

$\hat{\phi}$  is the carrier phase delay estimated by the receiver.

The in-phase and quadra-phase code replicas are then multiplied by the received signal, which has passed through the front-end filter. This product is then processed by an *Integrate and Dump* (I&D) filter that realizes the correlation. This can be modeled as:

$$I = \frac{1}{T_I} \int_0^{T_I} \tilde{s}(t) LR_I(t) dt \quad (2.10)$$

$$Q = \frac{1}{T_I} \int_0^{T_I} \tilde{s}(t) LR_Q(t) dt \quad (2.11)$$

where  $T_I$  is the coherent integration time, and

$(\tilde{\bullet})$  represents the filtering operation by the front-end filter.

Assuming that the change in code delay and phase delay is small during the integration time  $T_I$ , that the receiver front-end filter  $h$  has a one-sided bandwidth of  $B$  Hz, and that

the correlation process is realized within one data bit, the  $I$  and  $Q$  values can be approximated by (Holmes 2000):

$$I = \sqrt{\frac{P}{2}} \tilde{R}(\varepsilon_\tau) D \frac{\sin(\pi \varepsilon_f T_I)}{\pi \varepsilon_f T_I} \cos(\varepsilon_\phi) + n_I \quad (2.12)$$

$$Q = \sqrt{\frac{P}{2}} \tilde{R}(\varepsilon_\tau) D \frac{\sin(\pi \varepsilon_f T_I)}{\pi \varepsilon_f T_I} \sin(\varepsilon_\phi) + n_Q \quad (2.13)$$

where  $\tilde{R}$  is the correlation of the local spreading code with the filtered incoming spreading code,

$D$  is the sign of the navigation data bit,

$\varepsilon_\tau$  is the code group delay error ( $\varepsilon_\tau = \tau - \hat{\tau}$ ),

$\varepsilon_\phi$  is the carrier phase delay error ( $\varepsilon_\phi = \phi - \hat{\phi}$ ),

$\varepsilon_f$  is the frequency (Doppler) error ( $\varepsilon_f = f - \hat{f}$ ), and

$n_I$  and  $n_Q$  are independent Gaussian noises with equal power.

It can be seen that the signal power is split between the  $I$  and  $Q$  components, according to the estimated phase error.

Modeling the I&D filter as a simple integrator filter with a one-sided bandwidth of

$B_{ID} = \frac{1}{T_I}$ , and assuming that the front-end filter  $h$  has a one-sided bandwidth  $B$

significantly greater than  $B_{ID}$ , Appendix A shows that the noise components have the following power and autocorrelation function, as confirmed by Bastide (2004):

$$P_{n_I} = P_{n_Q} = \frac{N_0}{4T_I} \int_{-\infty}^{+\infty} G(f) |H(f)|^2 df \quad (2.14)$$

$$R_{n_I}(x) = R_{n_Q}(x) = \frac{N_0}{4T_I} \int_{-\infty}^{+\infty} G(f) |H(f)|^2 e^{-2i\pi fx} df \quad (2.15)$$

and

$$R_{n_I n_Q}(x) = 0 \quad (2.16)$$

where  $N$  is the PSD of the incoming noise  $n$ , and

$H$  is the Fourier transform of the front-end filter  $h$ .

The filtered autocorrelation function can be modeled by:

$$\tilde{R}(x) = \int_{-\infty}^{\infty} G(f) H(f) e^{2i\pi fx} df \quad (2.17)$$

Assuming that the front-end filter is rectangular with a one-sided bandwidth of  $B$  Hz,

and a perfect carrier wipe-off operation is realized ( $\varepsilon_\phi = 0$  and  $\varepsilon_f = 0$ ), it is interesting

to note that the post-correlation SNR, taken on the  $I$  component, is equal to:

$$SNR_{post} = \frac{2PT_I \tilde{R}^2(\varepsilon_\tau)}{N_0 \tilde{R}(0)} \quad (2.18)$$

The pre-correlation SNR, defined with the signal used before the correlation process, is

given by:

$$SNR_{pre} = \frac{P\tilde{R}(0)}{N_0 B} \quad (2.19)$$

where  $\tilde{R}(0)$  can be considered as the power loss due to the front-end filter.

Consequently, the correlation gain  $CG$  can be defined as the ratio between the post-correlation SNR and the pre-correlation SNR, and is given by:

$$CG = \frac{SNR_{post}}{SNR_{pre}} = \frac{2\tilde{R}^2(\varepsilon_\tau)T_I B}{\tilde{R}^2(0)} \quad (2.20)$$

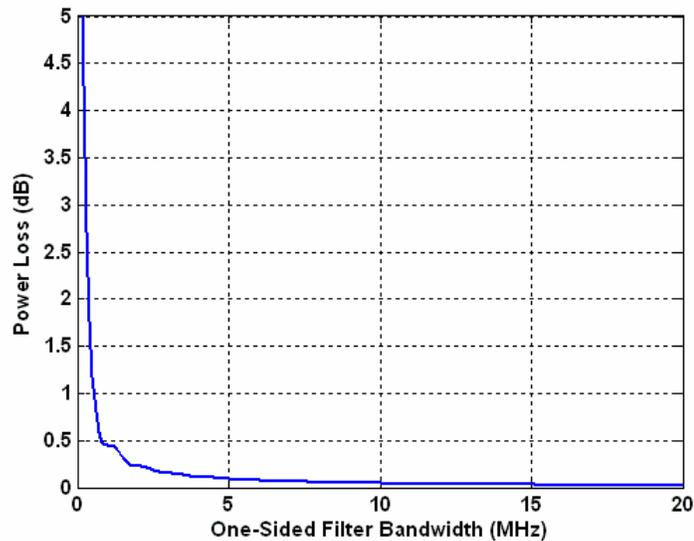
Assuming perfect code delay estimation,  $CG$  can be written as (confirmed by Macabiau (2001)):

$$CG = 2T_I B \quad (2.21)$$

Several conclusions can be drawn from the results shown above. First of all, if all of the parameters are accurately estimated, it is easy to see in Equation (2.12) that the value of the navigation data will fully appear on the  $I$  correlation component. The ability to extract the data bit sign is then dependent upon the post-correlation SNR given in Equation (2.18). The higher the SNR, the easier the data navigation bit can be read. It is clear that a longer integration time will proportionally increase the correlation gain. In a more general way, it is understandable that a high post-correlation SNR will provide a signal with less noise and will ease any processing of the correlation values.

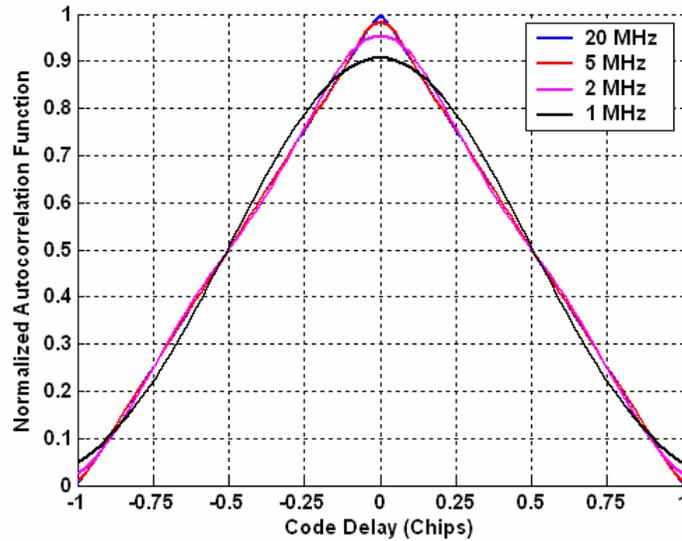
The choice of the front-end filter bandwidth will also play a role in the magnitude of the signal power processed in the receiver, since a narrow filter will cut out the outside lobes of the GPS C/A spectrum. Consequently, a large front-end filter bandwidth allows reception of more of the useful signal, as shown in Figure 2.4 for a rectangular front-end filter that has a unity gain within  $\pm B$  Hz. It is interesting to note that, for a filter that

would just filter the main lobe of the useful signal spectrum ( $B = 1.023$  MHz), the signal power loss would equal 0.45 dB, as confirmed by Van Dierendonck (1997).



**Figure 2.4 – GPS C/A Signal Power Loss due to the Front-End Filter Bandwidth**

The front-end filter bandwidth also has an impact on the shape of the autocorrelation function. Figure 2.5, obtained using Equation (2.17), shows that a narrow filter can significantly change the shape of the spreading code autocorrelation function, as compared to its ideal shape, and seems to round off the main peak. This means that a receiver architecture built assuming an ideal autocorrelation shape might undergo a performance degradation with a narrow-band filter, as will be seen in Chapters 5 and 6.



**Figure 2.5 – Impact of the Front-End Filter Bandwidth (One-Sided) on the Spreading Code Autocorrelation Function**

It is also extremely important to understand that the noise components of the correlation of the incoming signal with different delayed replicas of the locally-generated spreading code are correlated according to Equation (2.15). This is of major importance for the signal tracking performance in Gaussian noise, since GPS C/A signal code delay tracking is based on a code delay discriminator obtained from the combination of early and late correlation values, as will be seen in Chapters 5.

It should be noted that the correlation results shown are essentially valid for the tracking of other GNSS signals, unless otherwise stated. Consequently, these results (Equations (2.8) to (2.21)) will also be used throughout this thesis for the Galileo L1F signal.

Now that the GPS C/A signal has been addressed, and its main correlation properties analyzed, it is important to understand what aspects can receive improvements. Indeed, as previously mentioned, the increasing user demand in terms of positioning calls for a

general consolidation of the system and, in particular, higher reliability, greater sensitivity, better accuracy, and increased robustness.

## **2.2 GPS C/A Signal Limitations**

The desire by the public and professionals to use the GPS C/A signal under ever more challenging environments has shed light on some of the limitations of the signal. From a signal tracking perspective, the increased user expectation can be summarized in four main points: tracking sensitivity, tracking reliability, tracking accuracy, and tracking robustness.

### **2.2.1 Tracking Sensitivity**

Tracking sensitivity is defined herein as the minimum pre-correlation SNR that insures the correct tracking of the signal. It is understandable that, with the use of GPS chips in cellular phones and PDAs, the provision of positioning solutions, or at least measurement information, would greatly enhance GPS users' experience in degraded environments such as in buildings or shaded areas where the direct GPS signals might be degraded by more than 20 dB, as shown by MacGougan *et al.* (2002), Klukas *et al.* (2004) or Watson (2005).

One way to improve the tracking sensitivity of a receiver is to increase the post-correlation SNR as much as possible, since it is the correlation values that will be processed in the receiver tracking loops. As seen in Equation (2.18), the main parameter

used to increase the post-correlation SNR is the coherent integration duration  $T_I$ . However, the presence of the data bit transitions on the GPS C/A signal imposes a hard limitation on the maximum possible coherent integration time. This results in the impossibility of raising the post-correlation SNR value as high as needed; understandably, this will limit the receiver's tracking sensitivity (as already mentioned in the introductory chapter, AGPS receivers were designed to overcome this limitation, but with the cost of an additional data link). However, even if the received signal contained no navigation data, it is important to note that several constraints arise for extremely long coherent integration times, such as the code and phase delay variation during a coherent integration. This will be studied in Chapters 4 and 5. However, it would still allow a significant correlation gain, as explained in detail by Watson (2005).

To overcome this problem of potential limited coherent integration duration, it is possible to use non-coherent summations to obtain a correlation gain greater than that obtained through the maximum allowed coherent integration time. This technique is based on the summation of successive squared correlation values and can be represented by the following equation:

$$A = \sqrt{\sum_{k=0}^M X_k^2} \quad (2.22)$$

where  $X_k$  represents the successive correlation values obtained over a coherent integration time of  $T_I$ , and

$M$  is the number of non-coherent summations.

This process allows realization of a further correlation gain, but is susceptible to squaring losses  $SL$  due to the squaring of the noise component in Equations (2.12) and (2.13), as described by Chansarkar & Garin (2000) and MacGougan (2003). Owing to this phenomenon, the correlation gain due to non-coherent summations cannot be as efficient as an equivalent coherent integration time of  $MT_I$ . Moreover, the efficiency of the non-coherent summations is dependent upon the post-correlation SNR of each correlation value. Indeed, the lower the post-correlation SNR, the more pronounced is the associated squaring loss. The correlation gain using non-coherent summations is then given by:

$$CG_{nc} = \frac{2MT_I B}{SL(SNR_{post})} \quad (2.23)$$

where  $SL$  is given in Lachapelle (2004a).

It must be noticed that the use of non-coherent summations with limited coherent integration durations is potentially less sensitive to code and phase delay variations since the coherent integration is made over a shorter duration. If synchronized with the data bit train this technique also becomes insensitive to data bit transitions.

This method is mainly used for signal acquisition (e.g. Chansarkar & Garin 2000; Bastide *et al.* 2002), but it can also be used for code tracking (e.g. Ward 1996). This will be further discussed in Chapters 5.

Of course, an easier way to increase the post-correlation SNR would be to increase the signal power at the satellite end; however, this entails a heavier payload and,

consequently, a significant increase in the price of the satellite launch. Furthermore, an inordinately high signal power would induce problems of interference with existing terrestrial communication systems. For this reason, the power of satellite signals is subject to International Telecommunication Union regulations.

The receiver tracking sensitivity can also be separated between the sensitivity of each set of tracking loops. The PLL has to deal with the data bit transition in a particular way, since the transition implies a  $180^\circ$  phase shift. The design of the carrier phase and frequency loops then has to be insensitive to these phase shifts to maintain the same level of efficiency. This means that a classical pure PLL cannot be used, thus inducing an inherent loss in the tracking sensitivity, as it will be shown in Chapters 4.

The DLL is usually based on the combination of early and late correlation values, as will be demonstrated in Chapters 5. The net effect of this is that the noise correlation of each correlation output is also important to the resulting combined noise. Consequently, the choice of a relevant modulation that would limit the noise in the DLL code delay discriminator would have a beneficial effect on DLL tracking sensitivity.

### **2.2.2 Tracking Reliability**

Tracking reliability is defined herein as the ability of the tracking loops to track the incoming signal only using the signal spreading code autocorrelation main peak. From a tracking point of view, the main source of unreliability in tracking loops is the tracking of cross-correlation peaks, or auto-correlation side-peaks. As mentioned above, the GPS C/A code provides a minimum isolation of 21.6 dB against such secondary peaks.

Although representing a very comfortable margin in an open-sky environment, this value seems rather limited when considering more challenging environments. Indeed, the diversification of GNSS use as a navigation and service tool begets a desire to try to implement it in wide spread devices such as cars, cellular phones and PDAs. This implies that the receiver should be able to accommodate tracking signals passing through materials while other signals coming from other satellites might arrive unobstructed, such as in urban canyon areas and indoors. An example of the power loss experienced by a signal travelling through buildings is given by Klukas *et al.* (2004) and Lachapelle (2004b); the effects of such losses can easily reach the GPS C/A cross-correlation isolation level - indicating that, in degraded environments, there is a high likelihood of tracking cross-correlation peaks, as demonstrated experimentally by MacGougan (2003). Several methods, such as SafeTrack™ by NovAtel, have been designed to mitigate such a threat, but systematic removal of any chance of such an event at its origin remains the only comprehensive option. Consequently, the use of codes, for instance, with higher cross-correlation isolation capabilities would enhance the receiver's tracking reliability. This can be achieved through the use of longer codes, such as those chosen for the future GPS L5 (Van Dierendonck & Spilker 1999), or developing more efficient codes (Pratt 2002).

### **2.2.3 Tracking Accuracy**

Tracking accuracy is defined herein as the accuracy of the measurements obtained from the tracking loop considering the different sources of errors inherent to signal tracking,

and assuming reliable tracking of the direct signal. It is obviously greatly conditioned by the disturbances affecting the tracking process. The well-known high-level sources of tracking errors are: (1) thermal noise; (2) multipath; and (3) interference.

The ability of the tracking loop to mitigate thermal noise can be assessed using the Cramer-Rao lower bound tracking standard deviation in presence of white Gaussian noise. It is given by Betz (2001) as:

$$\sigma_{LB} = \frac{1}{2\pi\beta_{RMS}} \sqrt{\frac{B_L}{\lambda \frac{C}{N_0}}} \quad (2.24)$$

with

$$\beta_{RMS} = \sqrt{\int_{-B}^B f^2 \overline{G}(f) df} \quad (2.25)$$

where  $\beta_{RMS}$  is the RMS bandwidth of the signal,

$B_L$  is the DLL filter one-sided bandwidth (it will be defined in Chapter 4),

$\frac{C}{N_0}$  is the carrier-to-noise PSD ratio, and is equal to  $B \times SNR_{pre}$ , and

$\overline{G}(f) = \frac{G(f)}{\int_{-B}^{+B} G(f) df}$  is the signal PSD normalized over  $\pm B$  Hz.

As a consequence, from Equations (2.24) and (2.25), it is easy to understand that it is better, in terms of mitigation of the thermal noise, to have a significant part of the signal PSD away from the carrier frequency. BPSK modulation does not seem optimal in this

sense since it groups the main part of its power around the carrier frequency, as shown in Figure 2.2.

The resistance to narrow-band interference is usually assessed through the effective rectangular bandwidth  $\beta_{rect}$ . It represents the bandwidth of an equivalent signal with a PSD that would occupy the same area and the same maximum value as the test signal. The greater  $\beta_{rect}$ , the better the resistance to narrow-band interference.  $\beta_{rect}$  is given by Betz (2001):

$$\beta_{rect} = \frac{\int_{-B}^B G(f)df}{G(f_{max})} \quad (2.26)$$

where  $f_{max}$  is the frequency at which the signal PSD is maximum.

As in the case of the Cramer-Rao tracking lower bound standard deviation, the signal modulation plays a major role in the ability of the signal to mitigate narrow-band interference. Once again, it is interesting to note that BPSK modulation, with the main part of its energy located relatively close to the carrier frequency, might not be the optimal choice in this respect. This is understandable since the wider the signal PSD, the less the narrow-band interference will affect a significant part of the useful signal. As an example, an increase in the spreading code chip rate would widen the main lobe of the signal (such as for the GPS P(Y) signal), and would as a result improve the signal resistance to narrow-band interference.

The increase of the spreading code chip rate would also result in reduced susceptibility to multipath. This has been shown in numerous publications comparing the impact of multipath on tracking accuracy for the GPS C/A signal and the GPS P(Y) signal or other modulations, as shown by Braasch (1997), Betz (2001), and Hegarty *et al.* (2004). It is mainly due to the fact that shorter chip duration implies a reduced effect for longer multipath since only multipath with a delay shorter than 1 code chip will have a significant impact on tracking.

#### **2.2.4 Tracking Robustness**

Tracking robustness is defined herein as the ability of the tracking loop to maintain lock in all conditions. The main parameters affecting tracking robustness is the loop stability region, or the pull-in region, and the tracking accuracy. Usually, the wider the main peak of the autocorrelation function is, the greater the loop stability will be. Consequently, the long spreading code chip duration of the GPS C/A signal provides a tracking very resistant to high level of errors or short loss of lock. On the other hand, the GPS C/A signal does not provide an excellent mitigation of certain tracking errors, as already discussed it will be seen in Chapter 5.

Now that the main limitations of the GPS C/A signal in relation to the three tracking parameters considered have been analyzed, it is instructive to study the signal structure of Galileo L1F in order to understand how it performs with respect to these limitations.

## 2.3 Galileo L1F Signal Structure

As discussed above, the Galileo L1F signal has not yet been entirely defined. However, the main characteristics of the signal structure have been decided upon, and only a few parameters remain undetermined. The following description of the Galileo L1F signal is based on an early document released by the Galileo Joint Undertaking (2005).

Galileo L1 signals have an RHCP polarization, which facilitates the use of a same antenna for a combined GPS C/A / Galileo L1F receiver. Two signals will be broadcast on the L1 frequency band by Galileo:

- Galileo L1F, which is an *Open Service* (OS) signal, free of charge, and available to any user possessing a suitable receiver. This signal is the focus of this thesis. Along with the traditional navigation message, it also contains an integrity message and encrypted commercial data that will be available only to authorized users (Galileo Joint Undertaking 2005).
- Galileo L1P is a *Public Regulated Service* (PRS) signal. It contains encrypted ranging codes, and will be available only for regulated or critical applications by European and sponsoring state members.

Since the focus of this thesis is on the Galileo L1 mass-market signal, only the Galileo L1F signal will be thoroughly described herein. Its specified minimum received signal power with an isotropic antenna of 0 dBi gain is -157 dBW, or 3 dB greater than the current GPS C/A signal specification. This entails a higher pre-correlation SNR at the receiver antenna level. Galileo L1F is composed of two channels:

- A data channel (L1-B) that carries the Galileo navigation message, and
- A pilot channel (L1-C) that does not possess a navigation data bit.

The total power of the Galileo L1F signal is equally distributed between the data and pilot channels. As a consequence, taken separately, Galileo L1-B and L1-C will have the same minimum received power as the currently specified GPS C/A signal. However, if grouped in joint processing, the resulting higher signal power would allow a more accurate tracking.

The presence of a pilot channel allows the realization of longer coherent integrations and more robust phase tracking through the use of a pure PLL. As a result, it is meant to provide a robust tracking scheme on which to rely in case of low received signal power. The study of the actual benefit of the pilot channel on signal tracking will constitute a significant part of this thesis.

The data bit rate on Galileo L1-B has been set to 125 bps. This allows for the transmission of more data than GPS in the same time, enhancing the communication capacity of this signal. However, a higher data rate means that the BER will increase, since the energy carried by the signal during a given data bit will be reduced. To limit this impact, the navigation message is encoded using a *Forward Error Correction* (FEC) code. It corresponds to a Viterbi convolutional encoding with a constraint length of 7, and a coding rate of  $\frac{1}{2}$ . The effect of this FEC encoding is to offer a more reliable recovery of the data bit train. Its positive effect on the BER value is shown by Bastide *et al.* (2003). Another consequence of the FEC code is that the data channel will be actually

transmitting the encoded symbols at a 250 symbols per second (sps) rate to maintain a navigation message rate of 125 bps. Galileo L1F receivers will then have to possess a Viterbi decoder to access the navigation message. It is interesting to note that, because of the FEC encoding, the user will not have direct access to the navigation data, but will have to wait until the decoding is performed.

Galileo L1-B and L1-C channels are broadcast in quadrature-phase. It can then be seen as a *Quadrature Phase Shift Keying* (QPSK) signal with rectangular NRZ spreading symbols. The spreading codes used on each of the channels have a length of 4,092 chips, which is four times greater than their GPS C/A code counterparts. Even if the codes have not yet been released, as of April 26<sup>th</sup> 2005, this should allow for a better isolation against cross-correlation peaks by several dBs (e.g. if random sequences are used, an increase of the code length by a factor of 4 would result in a 6 dB decrease in the RMS side-peak level) and, hence, improvement in tracking reliability. Each channel uses its own code in order to lower the intra-signal interference.

The code chipping rate has been set to 1.023 Mcps, which is the same as for the GPS C/A signal. This means that a full code cycle lasts 4 ms and, as a result, a minimum coherent integration time of 4 ms is needed to avoid a partial correlation value. It also means that the length of a data symbol is the same as the length of a code cycle. Consequently, it is not possible to do a coherent integration of more than 4 ms on the data channel. Therefore this suggests that, in the absence of previous knowledge of the data bit train, the coherent integration time on the data channel is constrained to exactly 4 ms for optimal use.

The modulation chosen for both Galileo L1-B and L1-C channels is a *sine-phased*  $BOC(1,1)$ , referred to as sBOC(1,1). A slight optimization of this modulation for use by Galileo L1F is currently under examination (as of April 25<sup>th</sup> 2005).

A BOC signal is obtained through the product of an NRZ spreading PRN code with a synchronized square wave sub-carrier. This square wave can either be sine- or cosine-phased, which leads to different signal characteristics. They are referred to as sine-BOC and cosine-BOC, respectively (Hein *et al.* 2004). In the navigation community, a BOC signal is characterized by its spreading code frequency  $f_c$  ( $m \times 1.023$  MHz), and its sub-carrier frequency  $f_s$  ( $n \times 1.023$  MHz) with  $n$  and  $m$  being constrained to:

- positive integers,
- $n \geq m$ , and
- the ratio  $k = \frac{2n}{m}$  being a positive integer.

The sine sub-carrier can be modeled as:

$$SC(t) = \text{sgn}(\sin(2\pi f_s t)) \quad (2.27)$$

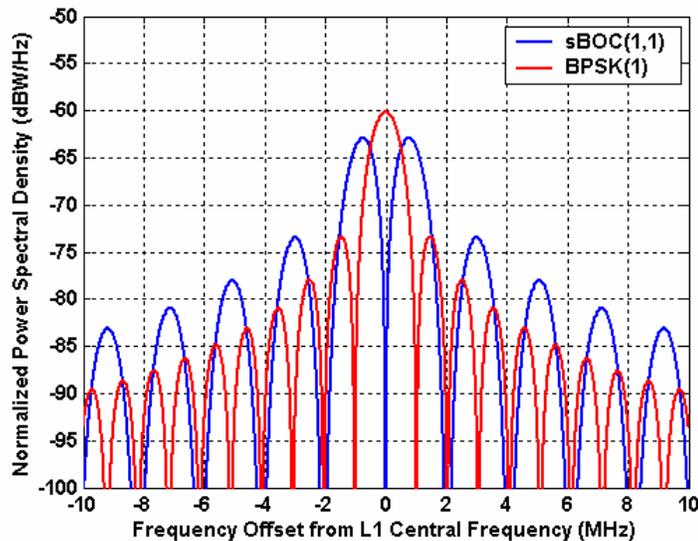
Each family defined by these two parameters is referred to as a BOC(n,m) modulation and has its specific spectral characteristics (Betz 2002, Rebeyrol *et al.* 2005). In order to avoid confusion, the BOC waveform will be referred to as spreading sequence (and not spreading code since it might be confused with the PRN code only).

The sBOC(1,1) modulation is part of the sBOC(n,n) family, where the length of one sub-carrier period equals one PRN chip duration. Its impact on code tracking will be

thoroughly investigated in Chapter 5. However, in order to have a high-level idea of its impact on signal tracking, a few details are given herein. sBOC(n,n) modulation splits the usual BPSK(n) spectrum into two symmetric side-lobes centered at  $\pm f_s$  MHz around the carrier frequency. This allows for a wider spectral occupancy. The sBOC(n,n) PSD envelope is given by Betz (2001):

$$G_{sBOC(1,1)}(f) = f_c \left( \frac{\sin\left(\frac{\pi f}{2f_c}\right) \sin\left(\frac{\pi f}{f_c}\right)}{\pi f \cos\left(\frac{\pi f}{2f_c}\right)} \right)^2 \quad (2.28)$$

The sBOC(1,1) PSD envelope is shown in Figure 2.6 along with the BPSK(1) PSD envelope that represents the GPS C/A code modulation.



**Figure 2.6 – sBOC(1,1) and BPSK(1) Normalized Power Spectral Densities**

The sBOC(1,1) PSD has its side lobes on the zeros of the GPS C/A code PSD. As a consequence, it is well suited to have good spectral separation properties from the GPS

C/A signal. This is important to avoid inter-system interference. The efficiency of the spectral separation is usually assessed using a spectral separation coefficient defined in Betz (2001) as:

$$\kappa_{S_1 S_2} = \int_{-B}^B \overline{G_{S_1}}(f) G_{S_2}(f) df \quad (2.29)$$

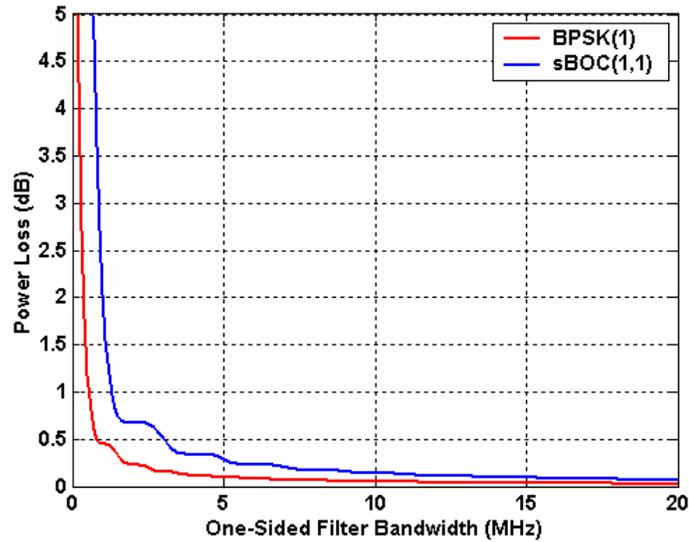
Table 2.1 summarizes the spectral characteristics of the BPSK(1) and sBOC(1,1) signals, assuming a receiver with a one-sided front-end filter of 20 MHz. It can be seen that sBOC(1,1) modulation brings a better inherent tracking noise, a better inherent narrow-band interference mitigation, as well as a good separation from the GPS C/A legacy signal as compared to the use of a BPSK(1) for Galileo L1F.

**Table 2.1 – BPSK(1) and sBOC(1,1) Spectral Characteristics**

	<b>BPSK(1)</b>	<b>sBOC(1,1)</b>
<b>Maximum PSD (dBW/Hz)</b>	<b>-60.1</b>	<b>-62.9</b>
<b>RMS Bandwidth (MHz)</b>	<b>1.1</b>	<b>2.5</b>
<b>Equivalent Rectangular Bandwidth (MHz)</b>	<b>1.0</b>	<b>1.3</b>
<b>Spectral Separation Coefficient with BPSK(1) (dB/Hz)</b>	<b>-61.8</b>	<b>-67.9</b>
<b>Spectral Separation Coefficient with sBOC(1,1) (dB/Hz)</b>	<b>-67.9</b>	<b>-64.8</b>

It can be seen from Figure 2.6 that a receiver tracking a Galileo L1F signal will require a wider minimum front-end filter than if tracking the GPS C/A code. This is one of the compensations for having better inherent tracking properties, which usually requires a wider frequency occupation. It also means that the minimum sampling rate fulfilling

Nyquist criteria will be higher than for a GPS C/A receiver. This implies higher power consumption and will reduce the autonomy of devices using this signal compared to using GPS C/A code. The wider sBOC(1,1) spectral occupation also implies that for an equivalent front-end filter bandwidth, there will be a higher power loss than for a BPSK(1) signal, as shown in Figure 2.7. As an example, for a 3 MHz (one-sided) front-end filter, the power loss is 0.15 and 0.5 dB for a BPSK(1) and a sBOC(1,1) signal, respectively.



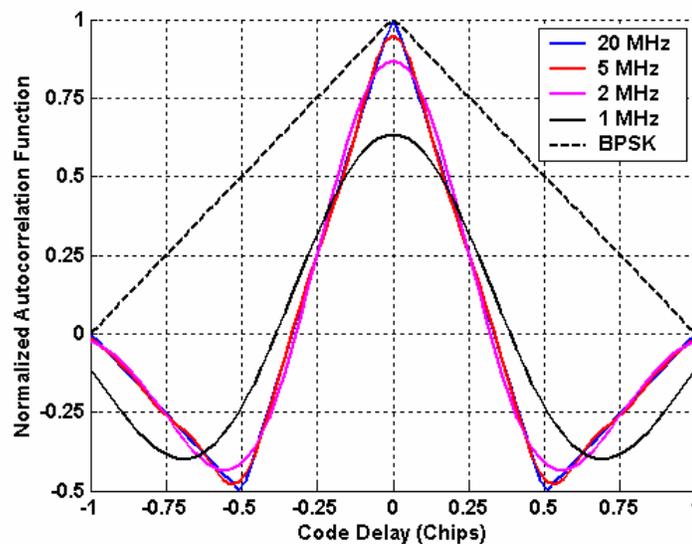
**Figure 2.7 – sBOC(1,1) and BPSK(1) Signal Power Loss due to the (One-Sided) Front-End Filter Bandwidth**

The theoretical autocorrelation of an sBOC(n,n) signal possesses three peaks in the region  $\pm 1$  chip, and is given by Julien *et al.* (2004c):

$$R_B(x) = \text{tri}\left(\frac{x}{1}\right) - \frac{1}{2} \left( \text{tri}\left(\frac{x - \frac{1}{2}}{\frac{1}{2}}\right) + \text{tri}\left(\frac{x + \frac{1}{2}}{\frac{1}{2}}\right) \right) \quad \text{for} \quad x \leq 1 \text{ chip} \quad (2.30)$$

where  $\text{tri}\left(\frac{x}{y}\right)$  is the triangular function of width  $2y$ , centered in  $x=0$  where it has a unity value.

Using Equation (2.17), it is easy to analyze the impact of the front-end filter bandwidth on the shape of the sBOC(1,1) autocorrelation function. It is shown in Figure 2.8 along with the theoretical BPSK(1) autocorrelation function.



**Figure 2.8 – Impact of the (One-Sided) Front-End Filter Bandwidth on the sBOC(1,1) Autocorrelation Function**

Due to its wider spectral occupation, the sBOC(1,1) autocorrelation function's main peak is sharper than that of the BPSK(1). It can also be seen that the front-end filter bandwidth rounds the main autocorrelation peak off. It also moves the location of the sBOC(1,1) side-peaks away from their theoretical location at  $\pm 0.5$  chips.

The Galileo L1F pilot channel might also possess a secondary code, also known as a synchronization sequence, which would be common to all Galileo satellites. This

secondary code is added modulo-2 to all the various spreading codes. Its length has been set to 25 bits, and the duration of one of its bit equals the duration of one spreading code cycle, or 4 ms. Although its exact sequence is still not publicly available, secondary codes in general have several purposes. The first one is to help to achieve data synchronization, as explained by Bastide (2004) in the case of GPS L5 signals. Indeed, once the acquisition on the spreading code has been achieved, and the code delay and Doppler estimation have converged, it is then possible to acquire the synchronization sequence, using its auto-correlation properties. Since the secondary code (possibly present on the pilot channel) is synchronized with the data bit transitions (on the data channel), it is easy to achieve data bit synchronization. To be efficient, the synchronization sequence is chosen to have an autocorrelation function providing good side-peak isolation. The secondary code that has been chosen for GPS L5 on its pilot channel is 20 bits long and provides a 7 dB side-peak isolation (Bastide 2004). Because of a similar secondary code length, the Galileo L1-C synchronization sequence is expected to exhibit approximately the same autocorrelation properties. Such a method for bit synchronization exhibits better behaviour in Gaussian noise than the usual histogram method (Van Dierendonck 1997).

It must be emphasized that, in order to take full advantage of the synchronization sequence and its autocorrelation function, the receiver must coherently integrate over the full secondary code sequence, which means 100 ms in the case of Galileo L1-C. As will be seen in upcoming chapters, this might create some problems in certain unstable conditions. It is also important to underline that, because of the presence of the

synchronization sequence in the pilot channel, no coherent integration longer than the secondary code bit duration is possible until synchronization is achieved. This might be a problem as far as acquisition is concerned since only non-coherent summations can then be used, with its corresponding squaring loss.

Another important property associated with the secondary code is that it also increases the signal resistance to narrow-band interference. Indeed, the use of the synchronization sequence superficially creates a spreading code whose length is the product of the length of the Galileo L1-C spreading code and the synchronization sequence length. This has spectral consequences since it means that apparent code length has been multiplied by 25, while the chip rate has remained the same. Thus, the number of spectral lines constituting the signal PSD is multiplied by the secondary code length and, since the same energy is carried by a greater number of spectral lines, this means that each one of them carries less energy. Consequently, if one spectral line is affected by narrow-band interference, it will have a lower impact on tracking performance.

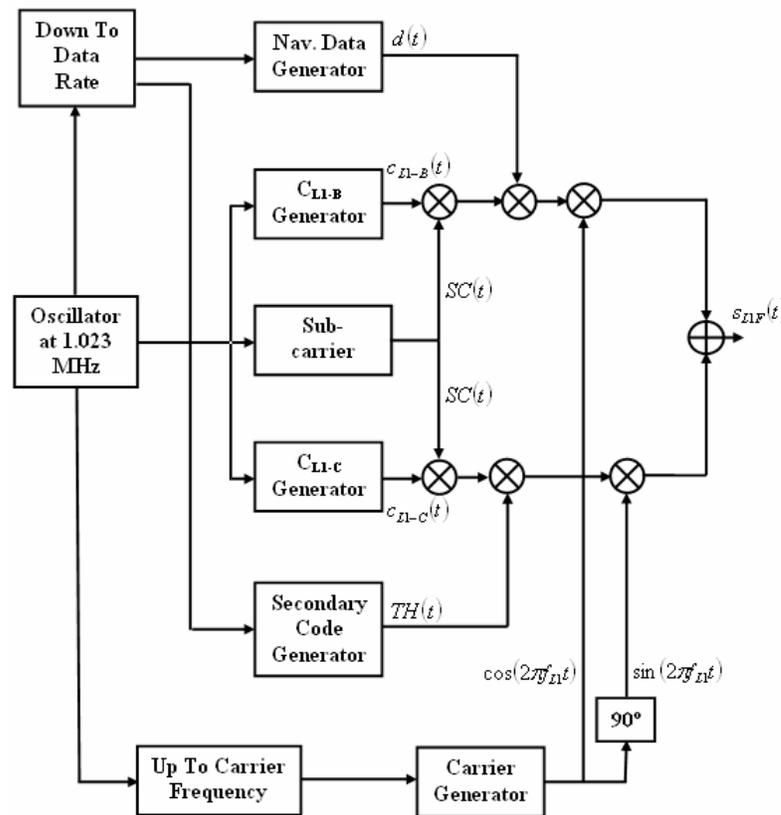
To conclude, Galileo L1F signal can be modeled as:

$$S_{L1F} = \sqrt{2P} \begin{pmatrix} c_{L1-B}(t-\tau)SC(t-\tau)d(t-\tau)\cos(2\pi f_{L1}t + \phi) \\ -c_{L1-C}(t-\tau)SC(t-\tau)TH(t-\tau)\sin(2\pi f_{L1}t + \phi) \end{pmatrix} \quad (2.31)$$

where  $TH$  represents the binary NRZ materialization of the secondary code on the pilot channel.

It is important to mention that, since the data and pilot components are synchronized, the code and phase delays are the same for both channels. The architecture of the signal

generation is shown in Figure 2.9. A summary of Galileo L1F signal characteristics is given in Table 2.2.



**Figure 2.9 – Galileo L1F Signal Generation Architecture**

As previously mentioned, the actual Galileo L1 signal actually consists of three components: the data and pilot channel of Galileo L1F and the PRS Galileo L1P signal. In reality, they will all be multiplexed in a *Coherent Adaptive Sub-carrier Modulation* (CASM), or tricode hexaphase modulation described by Raghavan *et al.* (1997), and Dafesh *et al.* (1999). However, the model used in Equation (2.31) still applies to a specific study of Galileo L1F and will be used throughout this thesis.

**Table 2.2 – Galileo L1F Signal Characteristics**

	<b>Galileo L1F</b>	
	<b>L1B</b>	<b>L1C</b>
<b>Code Length (Chips)</b>	<b>4096</b>	<b>4096</b>
<b>Code Chipping Rate (Mcps)</b>	<b>1.023</b>	<b>1.023</b>
<b>Modulation</b>	<b>sBOC(1,1)</b>	<b>sBOC(1,1)</b>
<b>Navigation Data</b>	<b>Yes</b>	<b>No</b>
<b>Data Rate (sps)</b>	<b>250</b>	<b>-</b>
<b>Data Encoding</b>	<b>Viterbi Convolutional (7,1/2)</b>	<b>-</b>
<b>Secondary Code</b>	<b>-</b>	<b>Yes</b>
<b>Secondary Code Length (Bits)</b>	<b>N/A</b>	<b>25</b>

## **CHAPTER 3**

### **SIMULATION TOOLS**

An investigation of the tracking loop performance can be realized theoretically. Since Galileo L1F signals are not transmitted operationally as yet, simulations present a workable means of confirming these results. This chapter introduces the simulation tools that will be used for this task, namely a software signal simulator and the tracking loops of a software receiver. The idea was to realize complementary tools that would represent the simulation and processing of the navigation data at the IF level. This chapter starts with a description of the major error sources the GNSS final position and commenting on its actual impact on signal tracking. The elaboration of a Galileo L1F signal simulator is then described. This tool aims at representing as precisely as possible a test signal arriving in the receiver tracking loops. Following this, the various models used to mimic a typical GNSS transmission channel are thoroughly explored. Finally, the architecture of the software receiver tracking loops is introduced.

## **3.1 GNSS Error Sources**

Before modeling the main GNSS error sources, it is instructive to become familiar with the nature of these errors in general. This will assist in assessing the necessity to model them to study the behaviour of the tracking loops, as intended by this thesis.

### **3.1.1 Satellite and Ephemeris Errors**

Satellite and ephemeris errors result from a mismatch between the actual location of the satellite and the predicted satellite position as broadcast in the navigation message. For GPS satellites, the prediction of the satellite position is realized for a minimum of 8 hours (three uploads per day). It is understandable that, over time, the accuracy of the prediction will decrease, due to the forces applied on the satellite. The standard deviation of the range error due to ephemeris inaccuracy is 2.6 metres according to Lachapelle (2004a) for current GPS satellites.

However, this type of error is seen as a bias by the tracking loops since, at the time the signal enters the receiver front-end, this error is already present. Consequently it will be tracked as part of the LOS signal propagation time. Indeed, with respect to error generation, the main concern for the tracking loops is a change in the signal characteristics during coherent integration times, or during a period of time that will be smaller than the tracking loop response, typically a few seconds to a few tens of seconds. Olynik (2003) showed that the satellite orbital error was correlated over periods of time greater than the receiver tracking loop filter response; i.e., greater than 90% for time

intervals lower than 5 minutes. This means that the change in the actual satellite-receiver range due to the satellite ephemeris error will be tracked without error by the tracking loops. A momentary range variation would be seen as signal dynamics by the receiver, in which case, it leads to errors similar in nature to those produced by high signal dynamics, as described in Section 3.1.8.

### 3.1.2 Satellite and Receiver Oscillator Frequency Noise

As already mentioned, GNSS relies on accurate timing to be able to provide accurate measurements. Yet, oscillators in satellites and receivers are not perfect, and they can provide only a limited FTS. This means that at both ends of the transmission channel, the satellite and the receiver should either be close to the common GPS reference time, or should assess their offset with respect to it. Oscillator timing error originates from the deviation of the oscillator from its nominal frequency. This is often described as the oscillator frequency noise. A more detailed treatment can be found in Sullivan *et al.* (1990) or Allan *et al.* (1997). The oscillator frequency noise can usually be modeled through three main components: random walk, Flicker and white frequency noise. The oscillator frequency noise PSD can then be written as (Winkel 2003):

$$S_{Osc\_Err}(f) = \frac{h_{-2}}{2f^2} + \frac{h_{-1}}{2f} + \frac{h_0}{2} \quad (3.1)$$

where  $h_{-2}$ ,  $h_{-1}$ ,  $h_0$  represent the random walk, Flicker and white components of the oscillator frequency noise respectively.

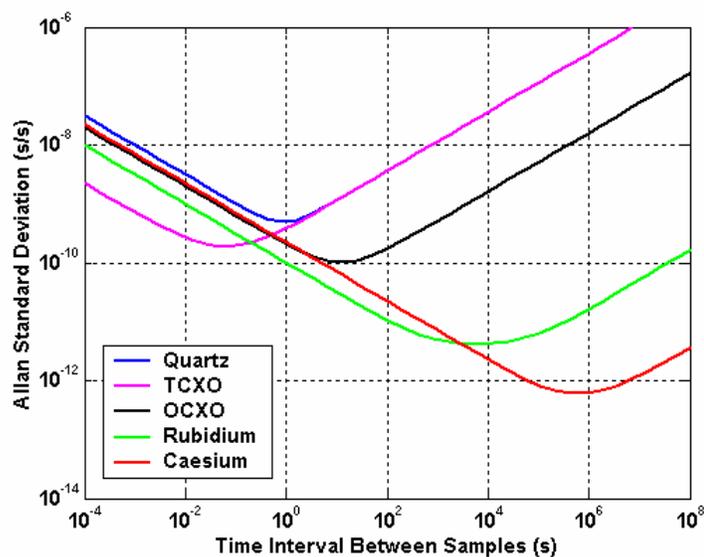
The deviation of the oscillator frequency from its nominal value translates into a phase error that will impair the proper transmission (satellite oscillator error) and processing (receiver oscillator error) of the signal. The resulting oscillator timing error is usually represented through the Allan variance, which represents half of the root mean square of the timing error between two adjacent samples separated by a true time interval  $\Delta$ . The three components of the oscillator frequency noise will have a different effect on the Allan variance. The Allan variance of an oscillator is given by Winkel (2003):

$$\sigma_A^2(\Delta) = \frac{h_0}{2\Delta} + 2\ln(2)h_{-1} + \frac{2\pi^2}{3}\Delta h_{-2} \quad (3.2)$$

Winkel (2003) gives a set of  $h_{-2}$ ,  $h_{-1}$ ,  $h_0$  values representing five different categories of oscillators: Quartz, Temperature Compensated Crystal Oscillator (TCXO), Oven Controlled Crystal Oscillator (OCXO), Rubidium and Caesium. These values are reproduced in Table 3.1, and their resulting Allan standard deviations are shown in Figure 3.1. The three components of the frequency noise can be observed: the negative slope corresponds to the dominance of the white component, the flat portion corresponds to the preponderance of Flicker frequency noise, while the positive slope, over a larger time interval, represents the random walk component.

**Table 3.1 – Parameters for a Set of Oscillators (Winkel 2003)**

	Oscillator Parametres		
	$h_0$ (s)	$h_{-1}$	$h_{-2}$ (Hz)
<b>Quartz</b>	$2e^{-19}$	$7e^{-21}$	$2e^{-20}$
<b>TCXO</b>	$1e^{-21}$	$1e^{-20}$	$2e^{-20}$
<b>OCXO</b>	$8e^{-20}$	$2e^{-21}$	$4e^{-23}$
<b>Rubidium</b>	$2e^{-20}$	$7e^{-24}$	$1e^{-29}$
<b>Caesium</b>	$1e^{-19}$	$1e^{-25}$	$2e^{-23}$



**Figure 3.1 – Allan Standard Deviation for Quartz, TCXO, OCXO, Rubidium and Caesium Oscillators**

Figure 3.1 shows that some oscillators can be well suited to providing short-term stability, while others are preferable when stability is required over the long term. In a GNSS satellite, long-term stability is important since it is desirable that the oscillator timing error modeling parameters transmitted in the navigation message remain valid for as long as possible to avoid the necessity of frequent uploading of these parameters. This is the reason why GNSS satellites typically employ Rubidium or Caesium oscillators which are known for their long-term stability. In a GNSS receiver, short- and medium-term stability (over periods of one millisecond to one second) is of major interest, since (as will be seen in subsequent chapters) the tracking process behaves better when the signal phase is stable during coherent and non-coherent integrations. From Figure 3.1, for instance, it seems that for this purpose a TCXO would exhibit better performance than a Rubidium or Caesium oscillator for integrations shorter than a few tens of milliseconds. Moreover, for commercial receivers, the cost of the oscillator is important, and a high

quality oscillator with excellent long-term stability characteristics would be prohibitively expensive, especially for mass-market applications. It is important to note that each oscillator is unique in its particular application; the results given here are simply meant to convey the importance of oscillator quality and not to deal with the complexities of actual oscillator operation.

The oscillator phase error may also originate from vibrations. Any motion of the transmitter or the receiver will obviously cause movement of the associated oscillator. In signals terms, this will translate into an oscillator frequency noise. This is usually referred to as external phase noise, as opposed to internal phase noise caused by inherent oscillator frequency instability (Irsigler & Eissfeller 2002). This vibration-induced phase error is closely tied to the g-sensitivity of the oscillator, given the relationship between piezoelectric accelerometers and the presence of vibrations (Van Dierendonck 1997). Typical examples of the PSD of these sorts of vibrations are given in Irsigler & Eissfeller (2002) and Hegarty (1997).

During signal transmission, the satellite oscillator error will result in a deviation of the carrier from its intended nominal frequency, and will result in a signal that can be modeled as:

$$c(t_{GPS} + \delta t_{Sat}) \cos(2\pi f_{L_1}(t_{GPS} + \delta t_{Sat})) \quad (3.3)$$

where  $\delta t_{Sat}$  is the timing error, with respect to GPS time, due to the satellite oscillator frequency noise.

Equation (3.3) shows that the resulting phase error is dependent upon the nominal oscillator frequency. Consequently, an oscillator with a lower nominal frequency will exhibit a lower phase noise.

At the receiver level, the oscillator phase noise will impact the local replica generation in the exact same way as shown in Equation (3.3). As a consequence, it will affect both the code and carrier wipe-off, and Equations (2.8) and (2.9) can be re-written as:

$$LR_I(t) = c(t_{GPS} + \delta t_{Rx}) \cos(2\pi f_{L_1} (t_{GPS} + \delta t_{Rx})) \quad (3.4)$$

$$LR_Q(t) = c(t_{GPS} + \delta t_{Rx}) \sin(2\pi f_{L_1} (t_{GPS} + \delta t_{Rx})) \quad (3.5)$$

where  $\delta t_{Rx}$  is the timing error, with respect to GPS time, due to the receiver oscillator frequency noise.

Due to the short carrier wavelength, the oscillator phase error can have a significant impact on carrier-phase tracking. However, this effect will be negligible on the code delay error budget since the chip length is far longer. The tracking loop measurements are made with respect to the receiver time and, as a result, it must be understood that the error source for the tracking loops is not the deviation of the oscillators from the GPS reference time, but the phase variations during signal tracking. Indeed, this impacts the carrier wipe-off during coherent integration, and will often be faster than the tracking loop time response. In Section 2.1.3, it has been seen that the assumption used to obtain the correlation output values assumed that the phase error would not change significantly during the coherent integration - an assumption that may not always be valid due to the particular oscillator characteristics.

### 3.1.3 Ionosphere

The ionosphere extends approximately from 50 to 1000 km above the Earth surface and is composed of free ions and electrons. These electrons directly impact the signal propagation properties of this layer. The ionosphere has an opposite effect on the code and carrier phase portions of the signal. It delays the code and advances the carrier phase by an equal amount, as discussed previously. The typical measurement error coming from the ionospheric effect is around 7 m for a satellite at the zenith, but it is greatly dependent upon the nature of the ionosphere activity, and the elevation of the satellite (Lachapelle 2004a). Signals coming from low-elevation satellites will undergo a higher ionospheric error, since they effectively transit a thicker ionospheric layer due to the low angle of incidence. During ionospheric storms, the signal can be delayed by tens of metres. During a normal day, the ionospheric error is usually at its maximum around 2 p.m. (local time) and at its lowest during night time. It is interesting to note that the ionosphere is a dispersive medium, and consequently will impact signals differently based on the frequency of transmission.

In general, the ionospheric delay changes very smoothly, and will not impact tracking performance, since it will be interpreted by the tracking loops as a part of the signal propagation time. The slow change in the ionospheric error, translating in a propagation time variation, will be easily tracked by the loops without bias. Olynik (2003) showed the autocorrelation function of the ionospheric error over 30-minute periods for high and normal ionospheric activity - illustrating that, even during periods of high ionospheric activity, the correlation over 5 minutes is around 90%. However, this does not mean that

sudden changes of the signal transmission time do not occur. Indeed, in extreme cases, high ionospheric storms or ionospheric scintillation can create very unstable propagation time changes that could be assimilated to high dynamics, and can potentially lead to loss of phase lock as shown by Coster *et al.* (2003) and Morrissey *et al.* (2002).

### **3.1.4 Troposphere**

The troposphere extends to 70 km over the Earth's surface and includes most of the water vapour. As distinct from the ionosphere, the troposphere is non-dispersive. It induces the same error (delay) on both signal code and phase. The tropospheric error is typically 2.5 m for a satellite at the zenith, and 9 m for a satellite with an elevation of 15 degrees (Lachapelle 2004a). Simple models can usually estimate the tropospheric error to within 20 centimetres (Lachapelle 2004a).

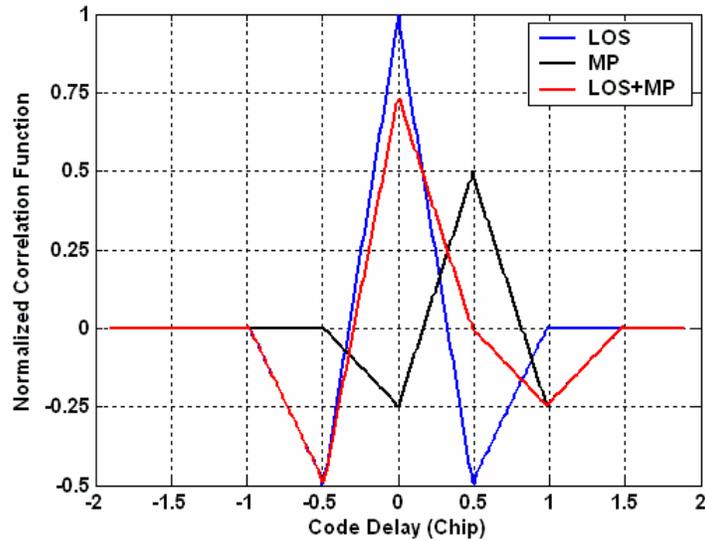
Similar to the ionospheric error, the tropospheric delay usually varies slowly with time, and will be seen by the tracking loops as a part of the signal propagation path. Olynik (2003) showed that its time correlation is higher than for the ionosphere (90% for a 10-minute time interval). Unless there is very high tropospheric activity, its impact on the signal propagation time change will not affect the behaviour of the tracking loops.

### **3.1.5 Multipath**

Multipath is known as one of the major sources of error in GNSS signal tracking. A multipath signal is a delayed version of the incoming signal that enters the receiver front-end and mixes with the direct LOS signal (Braasch 1997). It usually originates from the

reflection or diffraction of the direct signal on objects in the proximity of the signal's trajectory. It can be either diffuse or specular. Specular multipath originates from large and smooth surfaces. It is usually possible to apply Snell's law of reflection in such cases. This type of multipath can be very stable, and the resulting power loss with respect to the direct signal can be very small. Diffuse multipath is due to relatively rough surfaces that reflect the incoming signal in many different directions. It usually results in unstable multipath with a large power loss in the direction of the user antenna. In general, the nature of the reflective surface will greatly impact the multipath delay, phase, and amplitude.

At the receiver level, the correlation process is linear and, as a consequence, the multipath effect will be determined by its correlation with the local signal replica that will superpose the correlation of the local replica with the LOS signal. As seen in Section 2.1.3, the autocorrelation functions of the GPS C/A or Galileo L1F have very low values for code delays greater than 1 chip due to the nature of their PRN spreading codes. Consequently, the impact of multipath is usually significant only when its delay is smaller than 1 code chip. The superposition of the direct signal and multipath signals with the local replica is illustrated in Figure 3.2 for an in-phase multipath (same phase as the LOS signal) with an amplitude equal to half of the LOS signal amplitude. The distortion of the resulting correlation peak will impact tracking performance.



**Figure 3.2 – Impact of an In-Phase Multipath with a Delay of 0.5 Chips on the Normalized sBOC(1,1) Correlation Function**

The impact of multipath on signal tracking depends upon the tracking technology used and on the type of signal modulation. As an example, for the GPS C/A code, the code-tracking error can reach a magnitude of 150 metres for a multipath that has a magnitude that is half of the direct signal (Lachapelle 2004a). For phase-tracking, the error is bounded by one quarter of the signal wavelength. The impact of multipath on code tracking will be investigated in greater details in Chapters 5.

### 3.1.6 Thermal Noise

Thermal noise corresponds to the ambient noise present at the receiver antenna level. It is assumed to be white and Gaussian with a PSD equal to:

$$N_0 = K_B T_{sys} \quad (3.6)$$

where  $K_B$  is the Boltzmann constant ( $= -228.6$  dBW/K/Hz), and

$T_{sys}$  is the system noise temperature, dependent upon the front-end architecture (filters and *Low Noise Amplifiers* (LNA)), and defined through the Friis formula.

For a typical receiver, it is common to assume that  $N_0 = -205$  dBW/Hz.

### **3.1.7 Interference**

Any source of interference that spectrally overlaps GPS or Galileo signals will impact their tracking since it will enter the receiver front-end and will not be filtered out. The impact of a particular interference event on signal tracking greatly depends on the type of interference (e.g., Jiang 2004).

### **3.1.8 Dynamics**

Changes in the signal dynamics have an impact on signal tracking performance since tracking loops are normally designed to operate in response to a certain kind of dynamics. Moreover, it is possible that, due to the time response of the tracking loops' filter, high dynamics can result in degraded tracking performance or even a loss of lock. This will be further described in Chapter 4 and 5 for the PLL and the DLL.

Every phenomenon that tends to modify the signal propagation time will be seen by the receiver as signal dynamics, and will therefore imply a change in the code delay, carrier phase and Doppler values.

Now that the major error sources have been presented, the realization of the IF GPS C/A and Galileo L1F software signal generator can be presented.

## **3.2 IF GPS C/A and Galileo L1F Software Signal Generator**

### **3.2.1 Why an IF Signal Simulator?**

There are two ways to confirm theoretical results in practice: (1) to implement the technique studied and realize tests in the ‘real world’; and (2) to simulate the environment to test for the method. In the context of this thesis, Galileo L1F, the signal of interest, is not yet transmitted by any satellite; the first two Galileo test satellites, part of the *Galileo System Test Bed* (GSTB) (GSTB-V2/A and GSTB-V2/B), will be launched by the end of 2005 (Rooney *et al.* 2004). Methods developed theoretically at this point cannot wait on the launch of the first Galileo satellite in space for testing and verification. It is more appropriate to use a signal simulator that would faithfully replicate a typical GNSS environment.

On this basis, the choice of the signal generator design is then very important. Two main trends are currently competing: hardware and software signal simulators. Hardware simulators are expensive, and might not be as easy to configure as expected; however, they can provide robust data for real-time or *post hoc* processing (post-processing). The problem currently faced by hardware receivers is that the parameters of the Galileo L1F signal have not been fully defined as yet and, consequently, no commercial hardware receivers are available on the market with the ability to simulate Galileo L1F.

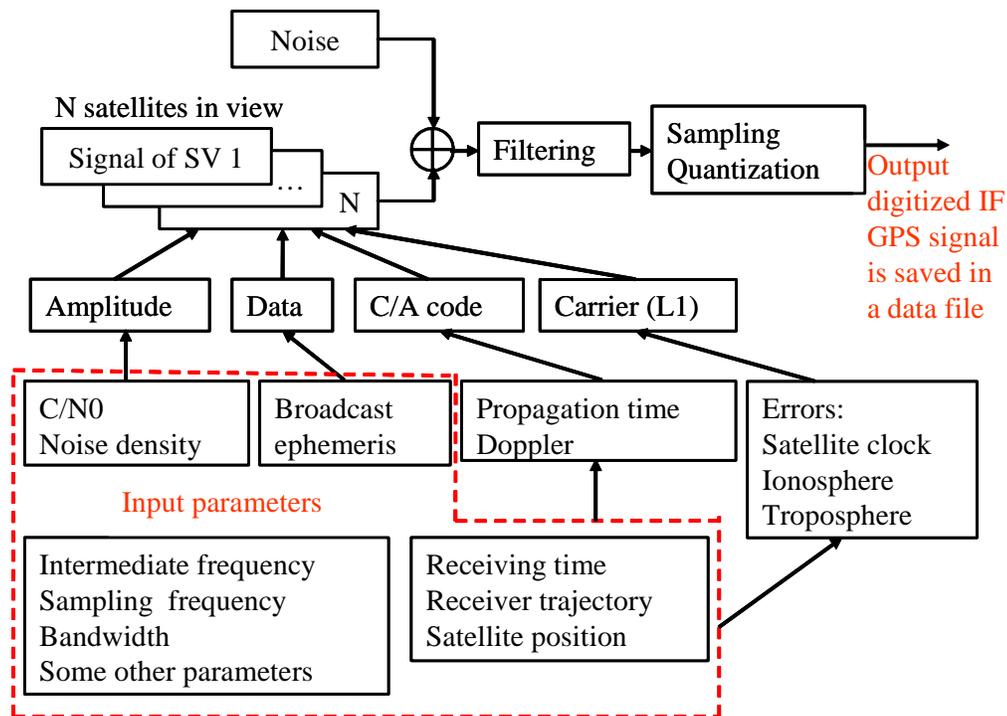
The development of a software signal simulator seems, in this context, a better solution for testing the methods developed in this thesis. Indeed, it provides, when well coded and structured, an extremely versatile tool that can be upgraded or modified very easily. Moreover, it can be integrated with a software receiver that takes IF data as input. Thus, its comparatively lower cost, versatility and adaptability were the principal reasons for developing an IF GNSS software signal generator.

The main concern in this case is to have a simulator that accurately reproduces the satellite–receiver propagation channel. This does not mean that the simulator has to completely reproduce all the sources of error, but it should efficiently model their key parameters. These error sources were presented and their relevance for tracking loop analysis described in the previous section.

Since this tool is designed to feed IF data to a software receiver’s tracking loops, it has also been decided to include the receiver front-end and the quantization (ADC) unit in its design. The coding of this tool was realized in C++, and was an enhanced version of the Matlab GPS C/A software simulator produced by Dong (2003).

### **3.2.2 Initial GPS C/A Software Simulator**

The first version of the IF software receiver was designed to simulate the GPS C/A signal only. It was created as part of a PLAN group effort to realize simulation of a complete GPS software receiver (Dong *et al.* 2004, Ma *et al.* 2004). Although all of the details are given by Dong (2003), this section will summarize the main results of this investigation. The overall IF GPS simulator structure is given in Figure 3.3.



**Figure 3.3 – Initial IF GPS Simulator Structure (from Dong *et al.* 2004)**

The GPS constellation is loaded using a broadcast ephemeris file that can be downloaded, for example, from the *Crustal Dynamics Data Information System* (CDDIS) institute website. This information forms the basis for the simulation of a true GPS constellation at a given time. Moreover, the ephemeris data allows taking into account relativistic effects. The computation of satellite position also allows the discounting of satellites situated below the horizon, or under a specified elevation mask.

Once the satellites in view are selected and their positions computed, the signal coming from each satellite has to be modeled. The model used in the GPS IF signal generator is an extension of Equation (2.3), including all the sources of errors described in Section 3.1 associated with the GPS C/A propagation channel:

$$S_{IF}(t) = \left( \begin{array}{l} \sqrt{2Pd} \left( t - T_d - \delta_{Iono}^{L1} \right) c \left( t - T_d - \delta_{Iono}^{L1} \right) \\ \times \cos \left( 2\pi f_{IF} t - 2\pi f_{L1} \left( T_d - \delta_{Iono}^{L1} \right) + \varphi_0 \right) + n(t) + MP(t) \end{array} \right) \quad (3.7)$$

with

$$T_d = \delta t_{Sat} + \delta t_{eph} + \delta t_{Tropo} + t_p \quad (3.8)$$

where  $f_{IF}$  is the IF frequency,

$\varphi_0$  is the initial phase,

$MP$  represents multipath,

$\delta t_{Iono}^{L1}$  is the delay due to the ionosphere on the L1 frequency,

$\delta t_{Tropo}$  is the delay due to the troposphere,

$\delta t_{eph}$  is the ephemeris error, and

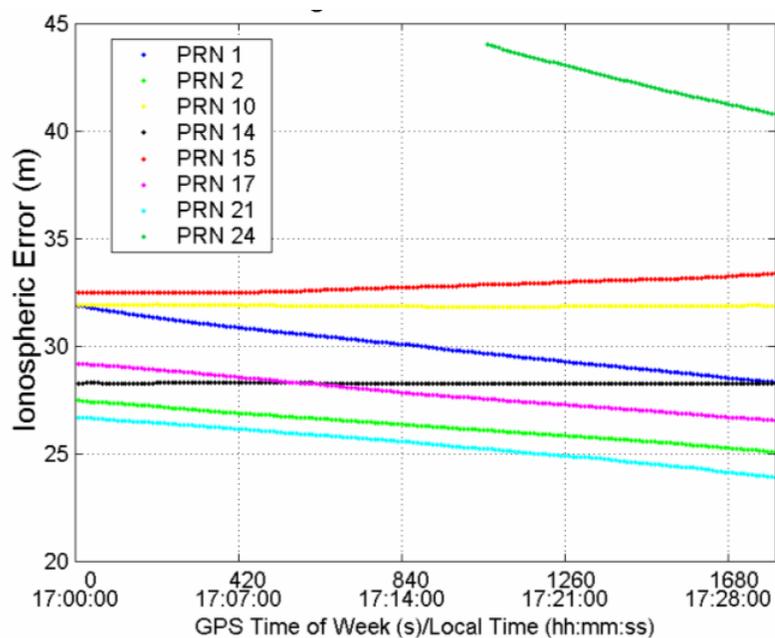
$t_p$  is the propagation delay.

All the sources of error in the signal propagation channel are represented in this particular modeling scheme. Although part of the final tracking error, the receiver clock error cannot be, *a priori*, included in the IF signal model, since it is an error source contributing only at the tracking level. Consequently, this error will be modeled in the IF software receiver.

It should also be noticed that Equation (3.7) includes the signal Doppler (range rate) information through the term  $2\pi f_{L1} \left( T_d - \delta_{Iono}^{L1} \right)$  (Dong 2003).

Because the signal takes some time to travel from the satellite to the receiver antenna, the Earth's rotation has to be taken into account. Indeed, the IF GPS simulator uses the receiving time as its reference time. This means that the satellite position at the time of transmission has to be calculated in order to model the correct range. This calculation is implemented in the simulator architecture.

The ionospheric error model is adapted from the model used in SimGNSSII<sup>TM</sup>, a software package developed by the University of Calgary that can model GNSS measurements. It has been used extensively for carrier phase positioning research (e.g., Alves 2001, Julien *et al.* 2004a, 2004b) as well as for GPS/Galileo reliability testing (Kuusniemi *et al.* 2004). An example of the ionospheric delay model, taken from Dao (2004) is shown in Figure 3.4.



**Figure 3.4 - GPS C/A Ionospheric Errors for all Visible Satellites during a 30 Minute Period**

The satellite clock error is modeled through the three parameters sent by each satellite in the ephemeris data. The thermal noise is added, taking into account the front-end filter shape. Each satellite  $C/N_0$  could be set individually, allowing testing with different levels of received signal power. One-bit quantization is possible after filtering, as most of the mass-market hardware receivers use this quantization.

As part of in-process monitoring of its design, the GPS software receiver was used to validate the performance of IF signal simulator at each major step of realization. A series of tests comparing the IF simulator data against IF data obtained from the combination of a GPS hardware generator and a L1 front-end were done. The corroborating results showed that a correct implementation of the GPS IF signal simulator was successfully achieved, as reported by Dong (2003).

This first version of a GPS software receiver is an extremely interesting starting point for the development of a combined Galileo L1F / GPS C/A signal simulator. Even if the objective of the desired enhanced signal simulator is to test the tracking loops of a Galileo L1F receiver, it is always interesting to consider the broader context of the work and to anticipate construction of a complete signal simulator. Given such an approach to the system's design, enhancements have been made to the propagation channel model in several respects:

- Addition of multipath modeling,
- Enhancement of the satellite clock phase error,
- Choice of different antenna gain patterns, and
- Realization of a received power pattern for the LOS signal.

However the first task of system realization involves the simulation of the Galileo L1F signals, which is discussed in the following section.

### **3.2.3 Galileo Constellation**

For the sake of simplicity, and for properly comparing the GPS and Galileo signals of interest, identical constellations have been chosen for both Galileo and GPS elements. This is not a major concern for tracking loop analysis since, once again, this thesis focuses on the measurement domain only, and not on 3-dimensional position. Moreover, Galileo L1F and GPS C/A signals will be studied independently, so the signal simulator will not be used with one satellite broadcasting both signals. As a result, even if it would be small in theory, no interference problem can occur.

### **3.2.4 Galileo L1F Waveform Model**

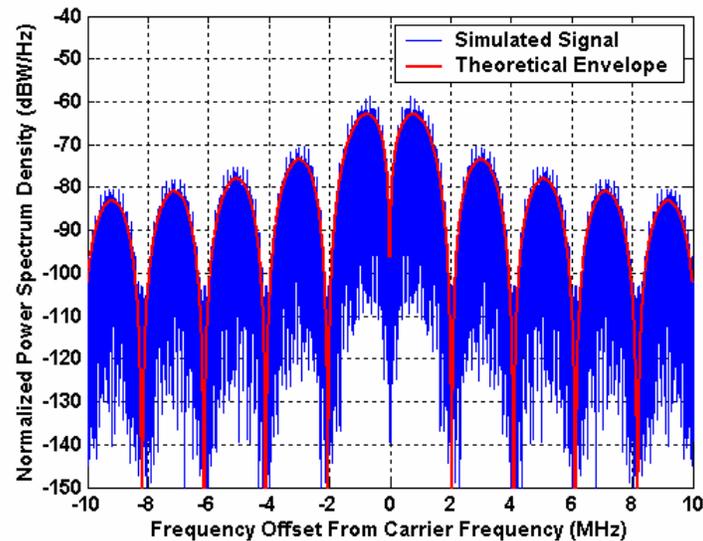
The Galileo L1F waveform design is described in Chapter 2. However, a major problem occurs for the actual implementation of the signal simulator, in that the actual Galileo spreading codes to be used are as yet unknown. The information accessible simply specifies the nature of the modulation, as mentioned in Section 2.3. Since it is not intended to use GPS C/A and Galileo L1F signals at the same time, it has been decided to use for Galileo L1F the same spreading codes as the GPS C/A codes. With the addition modulo-2 of the square sub-carrier, it still gives an sBOC(1,1) modulated signal. Since the Galileo L1F code length is known to be 4,096 chips, it has been decided to use, in the receiver part, coherent integration of 4 ms minimum.

It has been explained in Section 2.3 that the Galileo L1F pilot channel possesses a synchronization sequence in its structure. However, in the IF signal simulator, this synchronization sequence was not modeled, since the code associated with it is still unavailable to the public. Moreover, it is not expected to have a real impact on the results obtained, since this thesis focuses on signal tracking more than on the signal acquisition and transition to tracking, which are the main functions of the synchronization sequence (results on acquisition are shown in Chapter 6, but the synchronization sequence is then ignored). Consequently, all of the simulations realized with the signal simulator tool can be assumed to be analyzing the tracking loop after the synchronization sequence has been acquired.

To take all the parameters appearing in Equation (3.7) into account, the final model for the Galileo L1F signal at the entrance of the tracking loops can be re-written as:

$$S_{IF}^{L1F}(t) = \sqrt{P_{L1F}} \left( \begin{array}{l} c_{L1-B}(t - T_d - \delta t_{Iono}^{L1}) SC(t - T_d - \delta t_{Iono}^{L1}) \\ \times d_{L1-B}(t - T_d - \delta t_{Iono}^{L1}) \cos(2\pi f_{IF} t - 2\pi f_{L1}(T_d - \delta t_{Iono}^{L1}) + \varphi_0) \\ + c_{L1-C}(t - T_d - \delta t_{Iono}^{L1}) SC(t - T_d - \delta t_{Iono}^{L1}) \\ \times \sin(2\pi f_{IF} t - 2\pi f_{L1}(T_d - \delta t_{Iono}^{L1}) + \varphi_0) \end{array} \right) \quad (3.9)$$

In order to confirm the correct spatial occupation of the Galileo L1F signal, Figure 3.5 shows the normalized simulated signal PSD and the theoretical normalized sBOC(1,1) PSD envelope. Thus, the two PSDs agree very well.



**Figure 3.5 – Normalized PSD of the Simulated Galileo L1F Signal and of the Theoretical sBOC(1,1) Envelope**

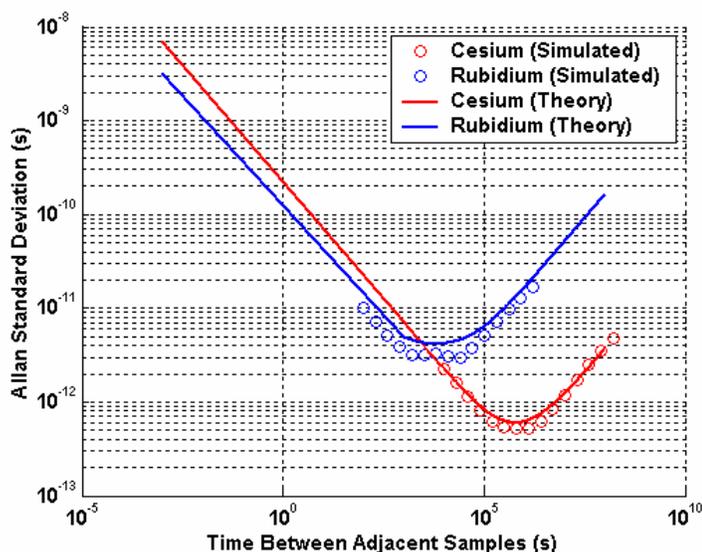
Now that signal modeling has been described, the enhancement of the propagation channel model can be reviewed.

### 3.2.5 Satellite Oscillator Model

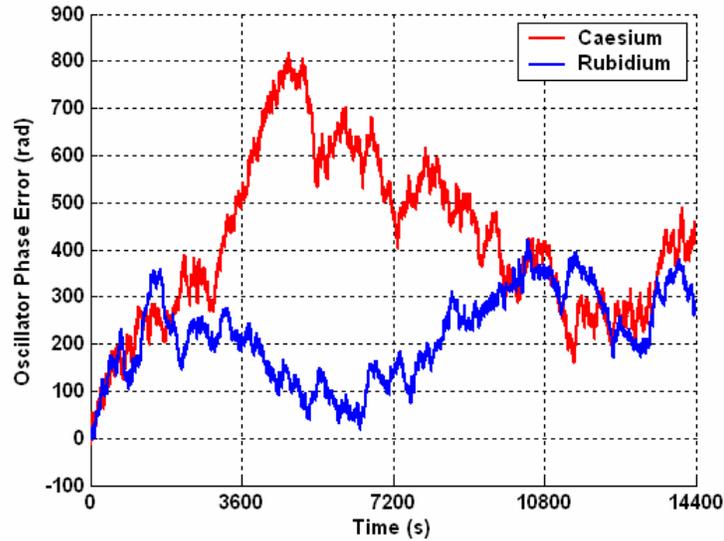
The satellite clock modeling done by Dong (2003) was realized through the three clock parameters transmitted in the satellite ephemeris. However, this modeling provides a very smooth satellite clock error that will not affect the operation of signal tracking since the code and phase-tracking loops will follow the smooth range change obtained from that modeling. In order to better represent the satellite oscillator error in the receiver tracking loop, it has been decided to model the actual satellite oscillator phase noise.

Winkel (2003) proposes a method of generating the oscillator phase error based on a system of differential equations. It offers a very practical way to generate this error

source with very small deviations from the Allan variance. Either Rubidium or Caesium oscillators are used in the satellites. Their theoretical Allan variance and the Allan variance obtained through simulations are shown in Figure 3.6. It can be seen that the Rubidium clock will have better short-term behaviour, but the Caesium oscillator exhibits extremely good long-term stability. Figure 3.7 shows the simulated behaviour of the Caesium and Rubidium oscillator phase errors over 4 hours. The oscillator frequency was assumed to be equal to L1. It can be seen that the induced phase error is varying very slowly, as compared to the tracking loops' typical time response (second level). It can then be anticipated that the satellite clock error will not have a significant impact on the behaviour of the tracking loops; however, modeling a 'low-cost' satellite oscillator (Quartz or TCXO for instance) at the satellite end can be very useful, especially to model very short and sudden changes in the signal dynamics in order to test a software receiver.



**Figure 3.6 – Theoretical and Simulated Allan Standard Deviation for Standard Caesium and Rubidium Oscillators**



**Figure 3.7 – Example of Caesium and Rubidium Oscillator (1575.42 MHz) Phase Error over 4 Hours**

It is also interesting to investigate whether it is possible to model the receiver oscillator phase noise in the propagation channel as well, since it is part of the overall error budget.

This could be made feasible due to the inverse problem:

$$\underbrace{\cos(2\pi f_{IF}t - 2\pi f_{L_1} \delta t_{Rx})}_{\text{Incoming}} \underbrace{\cos(2\pi f_{IF}t)}_{\text{Local}} = \underbrace{\cos(2\pi f_{IF}t)}_{\text{Incoming}} \underbrace{\cos(2\pi f_{IF}t - 2\pi f_{L_1} \delta t_{Rx})}_{\text{Local}} \quad (3.10)$$

However, Equation (3.10) does not take into account the effect of the front-end filter or of the quantization that might change the Allan variance of the simulated receiver oscillator phase noise. As a consequence, it was decided to model the receiver oscillator error only in the software GNSS receiver.

### 3.2.6 Multipath Modeling

The objective for modeling multipath is to enable the testing of multipath mitigation techniques in a fairly realistic environment. Many studies have been done to characterize multipath in different environments (Jahn *et al.* 1996, Brenner *et al.* 1998, Döttling *et al.* 2001, Steingass *et al.* 2004). They all underline that multipath in the signal propagation channel, when a spread spectrum technique is employed, can be divided into near and far echoes. Near echoes represent diffuse multipath and usually have a low power, which decreases exponentially with the magnitude of delay. Far echoes represent specular multipath and have longer delays; their power is a function of the qualities of the reflective surface(s). For the testing required herein, it was necessary to model the behaviour of these two types of multipath.

#### 3.2.6.1 Near Echoes

Because the GNSS IF simulator is intended to simulate close-to-real data, it was necessary to model each type of error, as much as possible, to suit actual conditions. As a result, although stochastic models of the impact of multipath on the direct signal exist (Ma *et al.* 2001), it was decided to model all scattered multipath entirely at the IF level. One way to realize this was discussed by Brenner *et al.* (2003), and used by Hegarty *et al.* (2004). It consists of modeling the effects of 500 small reflectors randomly located in a 100-m circle around the antenna. The mean power of each near echo can be modeled as (Jahn *et al.* 1996):

$$P_{ne}(\delta d) = P_0 e^{-\alpha_m \delta d} \quad (3.11)$$

where  $P_0$  is the average maximum power received from near echoes,

$\delta d$  is the delay of the echo, and

$\alpha_m$  represents the decay of the echo power with its delay.

The amplitude  $a_e$  of each echo varies around its mean value following a Rayleigh distribution:

$$P_{Rayleigh}(a_e) = \frac{a_e}{\sigma^2} e^{-\frac{a_e^2}{2\sigma^2}} \quad (3.12)$$

with

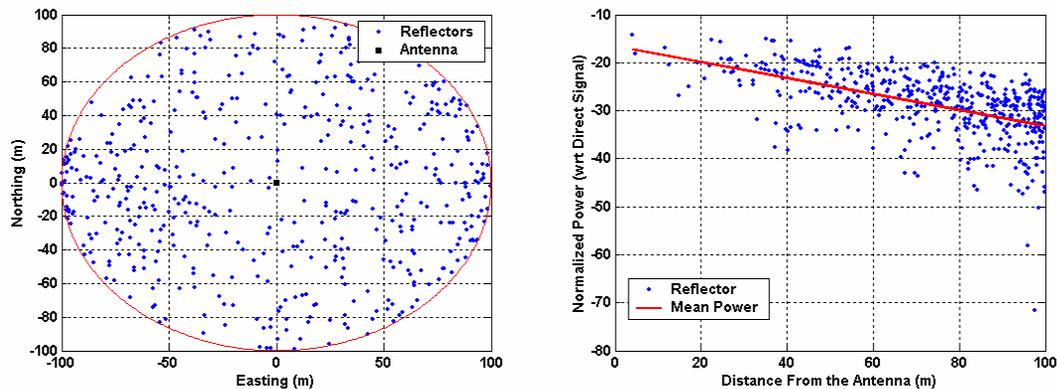
$$2\sigma^2 = P_{ne}(\delta d) \quad (3.13)$$

Different environments can then be modeled by changing the decay of the received power and the reflector's distance from the antenna (Jahn *et al.* 1996). An example of such a model is shown in Figure 3.8 for  $P_0 = -16.5$  dB and  $\alpha_m = -5$  dB/ms.

This approach is suitable for this application and has been tested successfully for the software receiver. However, when high sampling frequencies are used, it brings a high computational burden that might render the simulation inoperable because, for each reflector, the delay and power have to be adjusted with each sample. As a consequence, this tool can be used with signals with relatively small bandwidths and using a narrow filter, typically around a sampling frequency of 5 MHz. For larger sampling frequencies,

a lower number of small reflectors has to be simulated in order to have a reasonable computation time.

The methodology developed to simulate this kind of multipath is ideally suited to dedicated environments where any source of error has to be modeled. It can also be used as another source of strong multipath, when testing an algorithm in harsh environments, and affording the option of increasing the power of each echo signal.

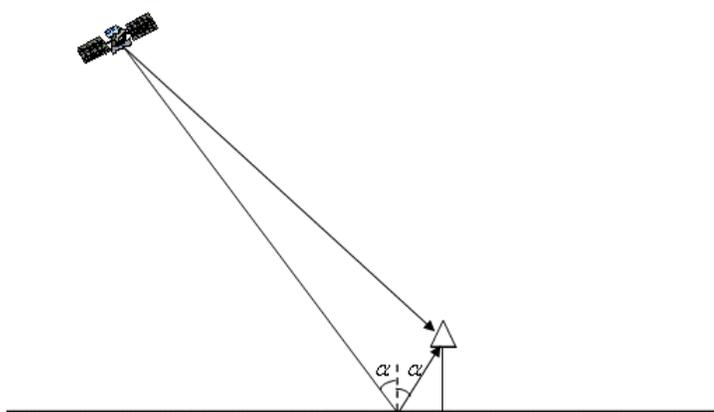


**Figure 3.8 – Repartition (Left), and Normalized Power (Right) of 500 Small Reflectors in a 100-Metre Circle around the User.**

### 3.2.6.2 Far Echoes

Due to their distinct characteristics, far echoes should be modeled differently from near echoes. They originate from larger and smoother surfaces and can have a higher power level, depending upon the properties of the reflective surfaces (however, despite the appellation ‘far echoes’ given in the literature, they can also originate from close objects, in which case they can have a strong power). Two sources of reflection were modeled for the IF GNSS signal simulator; namely the ground and obstacles.

Ground reflections are processed separately. An infinite horizontal ground plane is assumed, and a dedicated reflection coefficient can be assigned according to the surface being simulated. Because the ground is assumed to be flat and smooth, it is possible to apply Snell's laws for reflection where the angle of incidence is equal to the angle of reflection. Knowing where the satellite is at all times from the ephemeris file makes it easy to compute the reflection point, as well as the extra-path delay for the reflected signal. A model is represented in Figure 3.9.



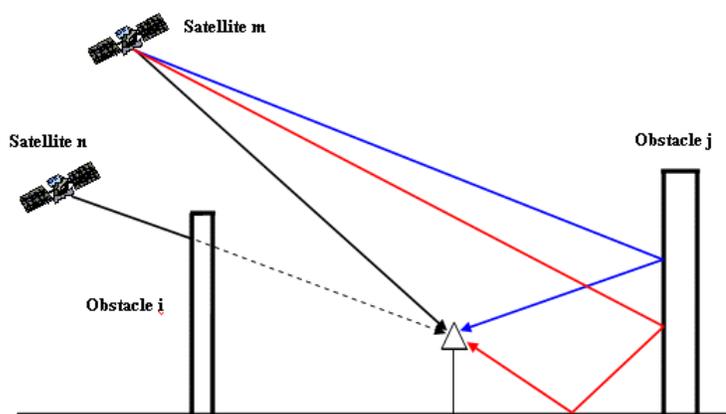
**Figure 3.9 – Ground Reflection Model**

Table 3.2 shows an example of the ground reflection delay for seven GPS satellites in view with different elevations for an antenna 2 m above ground.

**Table 3.2 – Ground Reflection Delay for 6 Satellites in View and an Antenna 2 m Above the Ground**

	Satellite Number						
	1	2	4	13	16	20	25
<b>Elev. (°)</b>	<b>64.1</b>	<b>39.1</b>	<b>20.1</b>	<b>32.6</b>	<b>45.3</b>	<b>57.2</b>	<b>37.8</b>
<b>Ground Reflection Delay (m)</b>	<b>4.33</b>	<b>5.98</b>	<b>11.3</b>	<b>6.32</b>	<b>5.48</b>	<b>3.68</b>	<b>2.87</b>

The second type of reflection simulated is that resulting from signal interaction with pre-defined obstacles. The user has the opportunity to define any kind of object by inputting the coordinates of each corner and the surface reflection coefficient, as well as its fading coefficient in the case of obstructions. The only constraint is for the object to be vertical; as such, a tetrahedral or pyramidal object cannot be virtually modeled by this method. The computation of the satellite coordinates at all times and a knowledge of the antenna position allow the user to determine if the direct signal is blocked - in which case, the obstacle's fading is applied to the direct signal. If the signal is not blocked, the program then searches for possible reflections against the obstacle. Finally, it also searches for a second reflection from the ground (along the satellite-obstacle-ground-antenna path). Figure 3.10 shows an example of the method used.



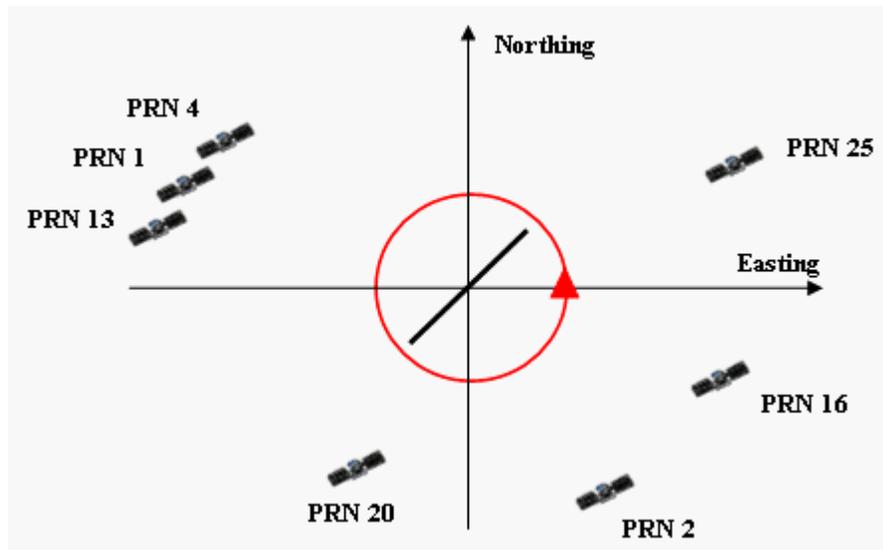
**Figure 3.10 – Example of the Obstacle Reflection/Blocking Model**

The algorithm permits simulation entailing any number of obstacles; however, the possibility of multiple reflections (i.e. involving reflections of the signal on multiple objects) was not implemented.

Such a tool is very interesting for modeling a specific environment of known building/object characteristics. Moreover, it is possible to apply blocking or fading from these obstacles according to the fading parameter chosen. Of course, very complex environments cannot be modeled due to the limitation in the models (vertical objects only, no multiple reflections, assumption of large smooth obstacles); however, it can still be used to simulate harsh environments, in addition to blocking gaps. The tool also incorporates a routine for polarization-inversive reflection of the multipath from RHCP to *Left Hand Circularly Polarized* (LHCP), and vice versa.

In order to demonstrate and verify far echo implementation, a test was set up, as shown in Figure 3.11. The user describes a 50-m circle around a tall obstacle whose edge is defined by the (Easting, Northing) coordinates of (30, 30) and (-30, -30). The obstacle's height is 50 metres. The antenna height was set at 2 m, and the user starts its trajectory at (50, 0). The amplitude of the multipath reflected from the obstacle is half of that of the direct signal. The obstacle's fading parameter was set at 50 dB to make sure the signal was blocked. Only GPS C/A signals were simulated.

The software receiver described by Ma *et al.* (2004) was used to process the IF data. Seven channels were used to track each satellite. A narrow correlator (0.3 chips) DLL was used (see Chapter 5 for details). The correct Doppler and code delays were entered in the receiver settings. The PLL used was set with a large loop bandwidth (30 Hz) in order to ensure carrier tracking after a short period of signal blocking (no re-acquisition procedure was used).

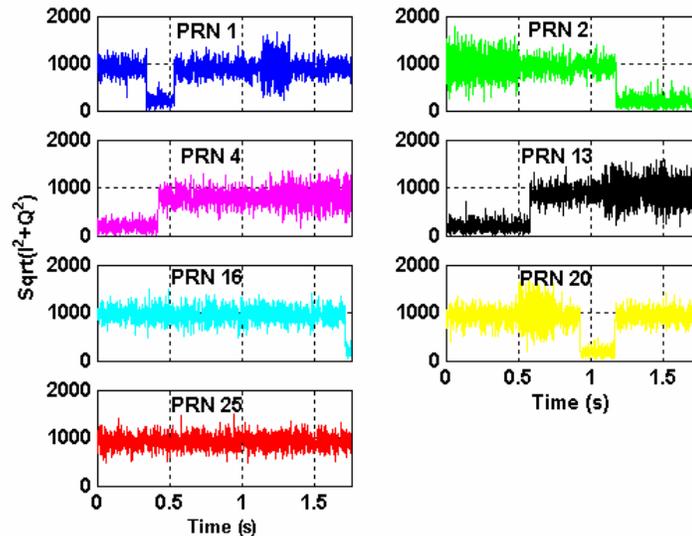


**Figure 3.11 – Far Echo Multipath Test Configuration**

Figure 3.12 shows the signal power for each of the seven satellites for the first 1.75 seconds of the modeled scenario, representing the first quarter of the circle travelled by the user. All of the satellite elevations are given in Table 3.2. To demonstrate the algorithm, two examples are described below:

- Satellite 1, thanks to its high elevation, can ‘see’ the receiver at the beginning, making tracking possible. The user then gets closer to the obstacle, inducing a masking effect that results in a loss of lock. The signal is reacquired shortly thereafter, as the user passes beside the obstacle. Then, because the user is in front of the obstacle and close by, it will be subject to reflections that can be observed through the noisier  $\sqrt{I^2 + Q^2}$  values. Finally, because the obstacle is only 50 m high, and the satellite is at a high elevation ( $64^\circ$ ), it cannot reflect the signal to the receiver, explaining the well-ordered tracking.

- Satellite 20 can ‘see’ the antenna at the beginning of the test and so is initially properly tracked. Shortly after, as the user gets closer to the obstacle, a strong reflection appears and enters the receiver front-end. When the user passes at the top right of the obstacle, no reflected signal occurs and cleaner tracking is possible until the user goes behind the obstacle. This causes masking and a loss of tracking. Finally, because satellite 20 is on the side of the obstacle, its signal can be re-acquired shortly after the obstruction.



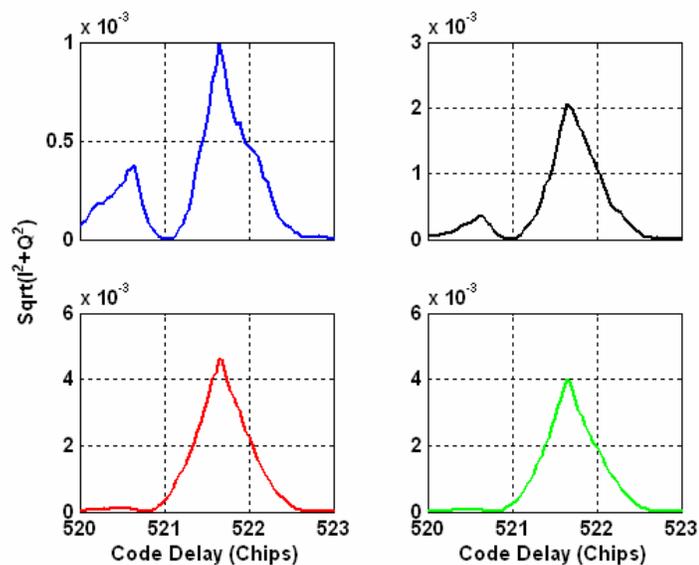
**Figure 3.12 – Estimated Signal Amplitude for the Seven Satellites Simulated During the First 1.75 Seconds**

In order to make the IF signal simulator more realistic, and since it is meant to feed the receiver tracking loops, it is also important to closely model the receiver front-end. In this respect, the received signal power, as well as the antenna gain pattern, is tremendously important.

### 3.2.7 Antenna Gain Pattern

The antenna gain pattern has a tremendous impact on signal tracking. Different applications will use antennas with different radiation patterns. For example, a ship will use an antenna with a high gain at low elevations in order to be able to keep tracking low elevation satellites even in case of large roll or bank angles. On the other hand, a geodetic antenna will have a low gain at low elevation in order to reject any potential multipath coming from the ground. Four antenna gain patterns have been implemented in the current version of the software GNSS IF signal generator. The first one is an isotropic antenna that has the same unit gain for all elevations and for both RHCP and LHCP signals. This antenna allows the possibility of having no discrimination at the antenna level if it is desired to use all the possible signals with predetermined  $C/N_0$ , or in order to receive strong multipath from diverse angles without mitigation thereof. The three other antennas were taken from the NovAtel family: the 501, 503, and 600 (NovAtel 2004). Both the RHCP and the LHCP gain patterns were implemented in order to allow discrimination of multipath with a change in its polarization. Of course, any antenna gain pattern can be implemented if other kinds of antennas are considered useful. Such a tool enables the testing of diverse algorithms using several candidate antennas to assess their impact on algorithm performance. Potential applications include the assessment of the importance of the antenna where a strong signal from a high elevation satellite, undergoing a high antenna gain, can affect the acquisition of a weaker signal from a low elevation satellite, with low antenna gain.

It can also be used to assess the impact of the antenna on multipath rejection, when combined with the multipath generation module of the software. As an example, a GPS C/A signal coming from a 65 degree elevation angle with a  $C/N_0$  of 50 dB-Hz was simulated, along with a long delay multipath (300 metres) with a power level equal to half the power of the direct signal, coming from a low elevation (10 degrees). The four types of antenna were used, and an acquisition process was initiated, employing a 5 ms coherent integration time. Figure 3.13 shows the value of  $\sqrt{I^2 + Q^2}$  during the acquisition process, assuming a correct Doppler estimate, and zooming in around the applicable code delay.



**Figure 3.13 – Estimated Signal Amplitude During the Acquisition Process Using a Uniform Unit Gain Antenna (Top Left), a NovAtel GPS 501 Antenna (Top Right), a NovAtel GPS 503 Antenna (Bottom Left), and a NovAtel GPS 600 Antenna (Bottom Right) in Presence of a Strong Multipath Coming from a 10 Degree Elevation**

It can be observed that the choice of antenna has a tremendous impact on mitigating multipath. While an antenna with a uniform unit gain is greatly affected by the multipath

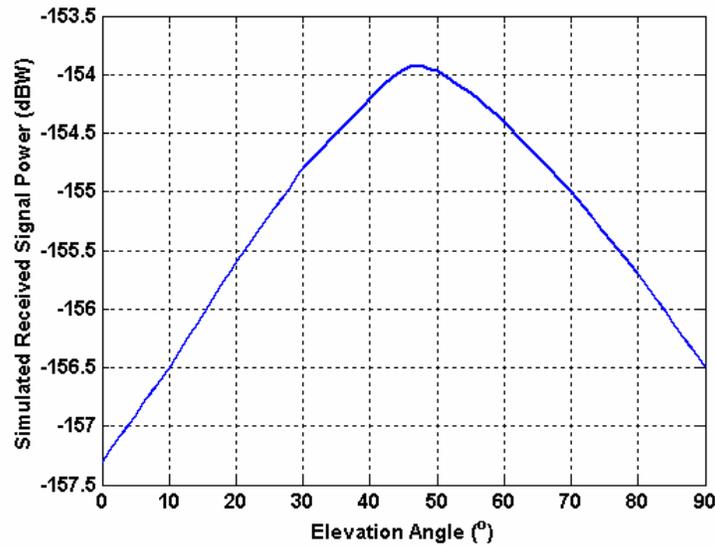
(having the second peak 1 chip away from the direct signal), the other antennas succeed in mitigating the multipath, with different levels of success. Moreover, it is interesting to note the difference in gain as a function of satellite elevation (the y-axis scale is different for three of the four figures).

### **3.2.8 Front-End Design**

The choice of a front-end filter is left to the user. Any type of digital front-end filter can be employed in the signal simulator. The thermal noise entering the receiver is adapted to the width and shape of the front-end filter.

The down-conversion from the L1 carrier frequency to the IF frequency was taken into account in the formation of the signal model shown in Equation (3.7).

It was also decided to attempt to employ relevant incoming signal power levels. The power level of the received signal is a function of the elevation angle. The GPS system was designed to impart the highest received signal power for satellites at a 45 degree elevation, and the same approach was adopted for the GNSS IF signal simulator. The received power, without any obstacles, is shown in Figure 3.14 in relation to the elevation angle (Hudnut *et al.* 2004). The difference in the received signal power between elevation angles of 45 and 90 degrees reaches approximately 2.5 dB, which cannot be neglected. This degree of received power differential will be important in certain cases, especially when a certain antenna gain pattern is required.



**Figure 3.14 – Simulated Received Signal Power as a Function of Elevation Angle**

Since the data and pilot channels of the Galileo L1F signal are supposed to each have the same minimum received power, according to the Galileo Joint Undertaking (2005), it was decided to also assume that their received power in the GNSS software signal simulator is equal to that shown in Figure 3.14.

The use of signal quantization is optional. However, if realized, quantization is done on one bit only. This provides an equivalent  $C/N_0$  loss that is dependent upon the bandwidth of the front-end filter. For example, the equivalent loss for the GPS C/A signal is 3.45 dB for a front-end filter one-sided bandwidth of 2.046 MHz and, similarly, 2.25 dB for a 5.115 MHz bandwidth (Van Dierendonck 1997).

Due to the circuitry of a traditional GNSS receiver, no implementation losses were modeled. As a consequence, the power of the signal after the front-end filter is the power

of the signal that will be processed by the software receiver tracking loops, described hereinafter.

### **3.3 IF Software Receiver Tracking Loops**

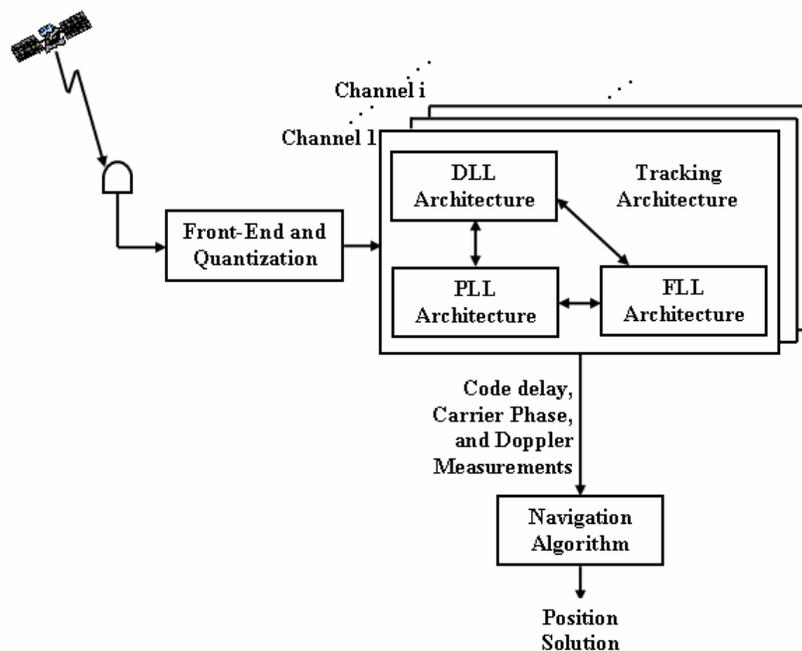
#### **3.3.1 General Architecture**

The general architecture of a GNSS receiver is given in Figure 3.15. As mentioned above, the front-end portion of a conventional GNSS receiver has been incorporated into the signal simulator. Consequently, the IF signal simulator can feed the tracking loops of the software receiver directly, without any requirement for pre-processing of the data.

The IF signal is then fed into the tracking architecture. It is usually composed of several parallel tracking channels that allow simultaneous tracking of several satellites. Each of these tracking devices is composed of two or three tracking loops: the DLL that tracks the spreading code delay; the PLL that tracks the carrier-phase change, and/or the FLL that tracks the signal Doppler. Note that the carrier-phase and Doppler measurements are related, the latter being the derivative of the former. A thorough description of the PLL and the DLL is given in the Chapters 4 and 5 respectively since their design is the main focus of this thesis. The composition of oscillator phase noise will play a role in the design of the tracking architecture, the modeling of which is described in the next section.

The measurements obtained from the tracking loops are provided to a positioning device that will take into account the navigation data message in order to compute each satellite-

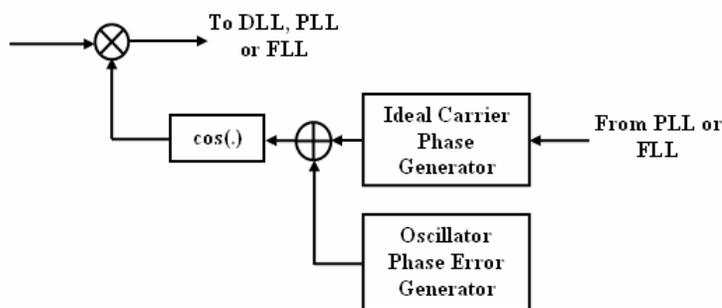
receiver range, and then obtain a final position solution through least squares or Kalman filtering (Axelrad & Brown 1997). It is important to understand that the measurements are made with respect to the GPS reference time. As a consequence, the measurements made by the tracking loops, obtained with respect to the receiver oscillator, have to be corrected. Since this timing error is common to all of the tracking channels, it is then usual to treat the receiver clock error as a fourth unknown in the Kalman filter, in addition to the user's three coordinates. This is not the case for the satellite clock error, since their oscillator is usually of very high quality, and their bias, drift and drift rate are estimated and broadcast in the navigation message.



**Figure 3.15 – High-Level Block Diagram of a GNSS Receiver**

### 3.3.2 Receiver Oscillator Phase Noise Model

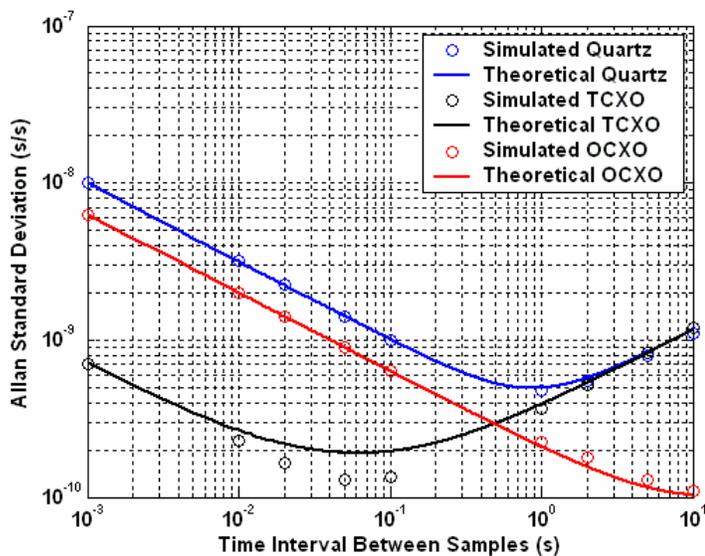
It has been shown that GNSS signal tracking is based on a correlation process, which is meant to extract low-power GNSS signals from thermal noise. The diagram of this correlation process was shown in Figure 2.3. It is easy to understand that the receiver oscillator phase error will have an impact on local replica generation, since it uses the receiver clock for its timing. The carrier generator, as well as the code generator, relies on the same clock and, so, they exhibit similar timing problems. The architecture used to generate the receiver oscillator phase error is shown in Figure 3.16. The same process is used to implement the oscillator phase error on the code and sub-carrier generation.



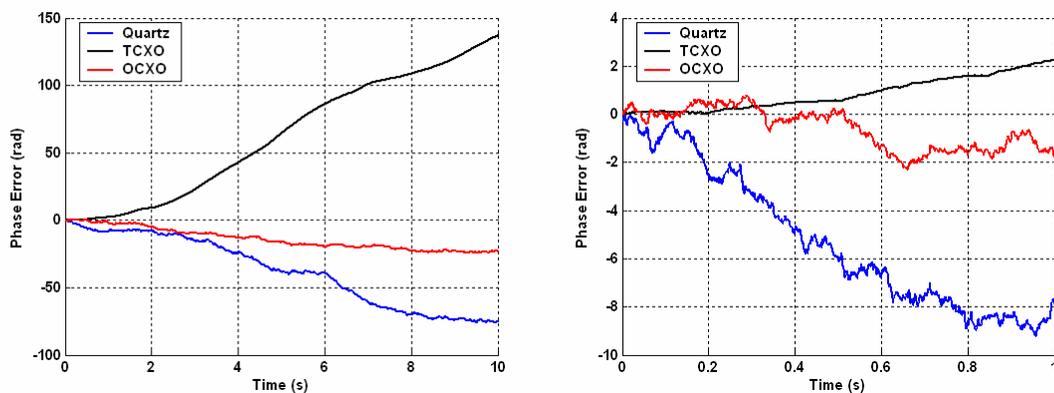
**Figure 3.16 – Block Diagram of the GNSS Software Receiver Carrier Generator for the In-Phase Correlation Component**

The model selected is also used to model the satellite phase noise. Its resulting Allan standard deviation is shown in Figure 3.17, along with the theoretical Allan standard deviation for Quartz, TCXO and OCXO oscillators over the time interval of interest for a receiver (typical tracking loop response time). It can be seen that the model used seems to slightly underestimate the TCXO phase error. Figure 3.18 show the short-term behaviour of the simulated phase noise for each oscillator. The Quartz oscillator exhibits very

sudden phase changes, while the TCXO oscillator seems very stable over short time intervals, as expected from its low Allan standard deviation for short time intervals. The OCXO shows good medium-term stability.



**Figure 3.17 - Theoretical and Simulated Allan Standard Deviation for Standard Caesium and Rubidium Oscillators**



**Figure 3.18 - Example of Quartz, TCXO and OCXO Oscillators (1575.42 MHz) Phase Error over 10 (Left) and 1 (Right) Seconds**

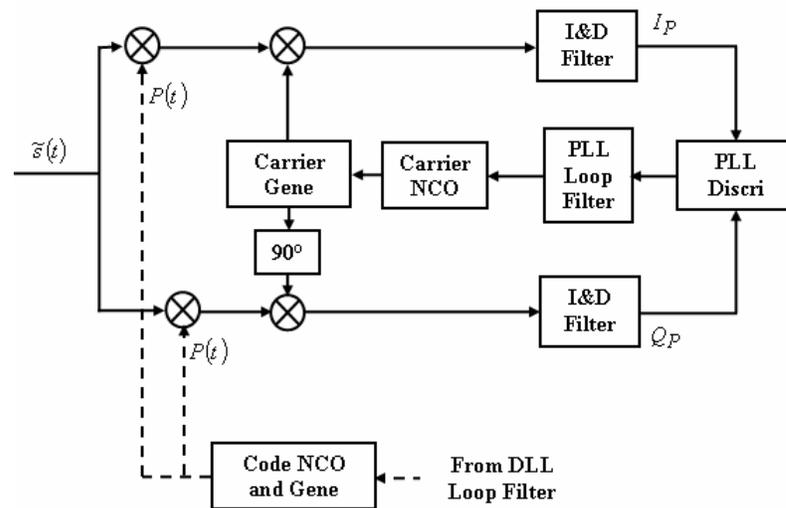
## **CHAPTER 4**

### **GALILEO L1F CARRIER PHASE TRACKING**

This chapter presents a discussion of how to maximally utilize the Galileo L1F signal structure for the implementation of optimal carrier phase tracking. After a brief introduction focused on general PLL theory, various phase discriminators specific to the data and pilot channels, and their influence on the overall phase tracking, are introduced. The main sources of error for carrier phase tracking are presented and their impact on PLL performance is thoroughly described. With reference to the various error models introduced, the tracking sensitivity allowed by the different PLL designs is then quantified and discussed. Finally, different combinations of Galileo L1F data and pilot channels are introduced and tested with the objective of improving the carrier phase tracking accuracy.

## 4.1 Generic PLL Structure

General PLL theory can be found in Holmes (1982) and Peterson *et al.* (1995). A generic PLL architecture is given in Figure 4.1. The incoming signal, filtered by the front-end, is first multiplied by a spreading sequence replica coming from the DLL using an estimate of the incoming signal code delay in order to wipe-off the spreading symbols.



**Figure 4.1 – Generic PLL Architecture**

The signal is then split into two branches. One branch is multiplied by an in-phase local carrier, and the other one by a quadrature-phase local carrier (shifted by  $90^\circ$ , as compared to the in-phase carrier replica). This allows the wiping-off of the signal carrier that is expected to occur at the IF stage. This carrier wipe-off is meant to closely follow the phase variations due to the change in signal propagation time between the satellite and the receiver. To achieve this, the PLL uses a phase discriminator meant to assess the phase error remaining after the multiplication by the local carrier. This discriminator uses the in-phase and quadrature prompt correlation values ( $I_P$  and  $Q_P$ ) obtained through

the use of an ‘*Integrate and Dump*’ (I&D) filter. As will be seen in the next section, the role of the discriminator is extremely important in the design of the carrier-phase tracking loop.

To improve its accuracy, the phase error estimate obtained from the discriminator is then filtered by a low-pass filter. This filter is meant to remove as much noise as possible from the discriminator estimate without filtering out the useful signal (which could be shifted in frequency due to the signal dynamics).

The PLL filter transforms the estimated phase error into an estimated frequency error that is meant to control the carrier NCO frequency. The carrier NCO generates the local carrier, taking into account its estimated frequency error input and, by doing so, is intended to impart a smaller phase error at the next epoch.

The equivalent loop filter,  $h_{PLL}$ , possesses a one-sided filter bandwidth  $B_L$  defined by (Holmes 2000):

$$2B_L = \int_{-\infty}^{+\infty} |H_{PLL}(2i\pi f)|^2 df \quad (4.1)$$

where  $H_{PLL}$  is the Fourier transform of  $h_{PLL}$ .

Examples of loop filter models, using a continuous update approximation (which assumes that  $B_L T_I \ll 1$ ), are given by Ward (1996). However, Stephens & Thomas (1995) have shown that, for larger  $B_L T_I$  values – when  $B_L T_I > 0.1$  – this continuous update approximation does not hold. They proposed a new set of parameters for use in this case, especially when long coherent integration times are considered, and it is these parameters

that are used in the implementation of the tracking loops in this thesis. Because the presence of a pilot channel on Galileo L1F allows for long integration periods, the particulars of these parameters are of major interest.

## 4.2 Phase Discriminators

Phase discrimination is the first step toward accurate phase error estimation. Two main figures of merit are typically used to characterize a discriminator's inherent tracking capacities:

- The *stability region*, which is defined as the region surrounding the zero phase error where a certain phase error input will result in a mean discriminator response having the same sign as the input error. This means that, for a certain input error, the discriminator will react in the correct direction and should converge towards zero phase error.
- The *linear tracking region*, which is defined as the region around the zero phase error where a certain phase error input will result in a mean discriminator response is equal to the input error. This means that, for a certain input error, the discriminator will react perfectly (that is, without bias).

The choice of the phase discriminator can vary, depending on the parameters of the signal structure; of particular importance is the presence or absence of data bits, since binary data will affect the polarity of the signal, which also implies a change in phase. Consequently, discriminators can be divided into two classes; that is, as based on signals with and without navigation data.

### 4.2.1 Data Channel

GNSS signals were designed to carry information through a data message. A rectangular set of NRZ data bits modulates the carrier, which results in apparent polarity switches at every data bit transition. This is equivalent, from the carrier-phase point-of-view, to a 180° phase shift and is clearly not acceptable for obtaining precise measurements since it implies that carrier tracking will be degraded by such sudden changes. Moreover, a jump of half a cycle also entails possible losses of lock if the loop cannot recover from it quickly. Consequently, the presence of data has two main impacts on the design of a PLL:

- The maximum coherent integration time is limited to the duration of the data bit, unless external ways of ascertaining the value of the data bits are available in advance (e.g. AGPS). For the Galileo L1F data channel and the GPS C/A signal, the coherent integration is limited to 4 and 20 ms respectively; and
- The discriminator must be chosen so as to be insensitive to the inherent phase jumps due to the data bits.

To overcome the second problem noted above, special discriminators have been used. The most widely used is the Dot Product (DP) phase discriminator, given by Ward (1996):

$$D_{DP} = I_P Q_P \quad (4.2)$$

Assuming no external disturbances, and using the model given in Equations (2.12) and (2.13), the DP discriminator equals:

$$D_{DP} = \frac{P}{4} \tilde{R}^2(\varepsilon_\tau) \sin(2\varepsilon_\phi) \xrightarrow{\varepsilon_\phi \rightarrow 0} \frac{P}{2} \tilde{R}^2(\varepsilon_\tau) \varepsilon_\phi \quad (4.3)$$

The DP discriminator is insensitive to 180° phase jumps since it is based on the product of the in-phase and the quadra-phase prompt correlation components that will both change sign simultaneously during a data bit transition. It should also be noted at this point that this discriminator actually tracks twice the phase error which reduces its stability domain to  $[-\pi/2; \pi/2]$ . It is also essential to note that this discriminator requires a normalization to remove the impact of the signal power.

Another possible discriminator insensitive to data bit transition is the classic arctangent ( $A_{\tan}$ ) discriminator. Its expression is given by Ward (1996):

$$D_{A_{\tan}} = a \tan\left(\frac{Q_P}{I_P}\right) \quad (4.4)$$

Without any external disturbances, it equals:

$$D_{A_{\tan}} = \varepsilon_\phi \quad (4.5)$$

Compared to the DP phase discriminator, it possesses the same stability range but with an extended linear tracking region. Moreover, this discriminator does not need any normalization. Although this might appear to be an advantage in terms of tracking loop architecture, it might present a drawback in terms of performance. Indeed, an external normalization can be done in many ways, and it may be possible to assume the use of

very precise normalizations that would not introduce a significant amount of noise - for instance, through long coherent/non-coherent integrations. On the other hand, in the case of the arctangent discriminator, self-normalization entails a division between two correlation values. Consequently, if the in-phase correlation value becomes noisy, because of low  $C/N_0$  or a short integration time, this can have a potentially serious impact on the discriminator output. This will be studied more thoroughly in Section 4.3.1.

The two discriminators introduced above have the characteristic of having several lock points separated by  $\pi$  radians, which means that they are subject to half-cycle slips. The overall performance of both data discriminators under the main sources of errors affecting phase tracking will be studied in Section 4.3.

#### 4.2.2 Pilot Channel

In perfect conditions, the use of a pilot channel allows coherent integration for as long as desired. It also allows the PLL to use a discriminator that is not required to be insensitive to  $180^\circ$  phase jumps. Consequently, other sets of discriminators than the ones presented for the data channel can be used. Two main proposed discriminators are studied herein: the coherent and extended arctangent discriminators.

The coherent discriminator has been formulated by Hegarty (1999), namely:

$$D_{Coh} = Q_P \tag{4.6}$$

Without considering any disturbances, this equals:

$$D_{Coh} = \sqrt{\frac{P}{2}} \tilde{R}(\varepsilon_\tau) \sin(\varepsilon_\phi) \xrightarrow{\varepsilon_\phi \rightarrow 0} \sqrt{\frac{P}{2}} \tilde{R}(\varepsilon_\tau) \varepsilon_\phi \quad (4.7)$$

Unlike the DP phase discriminator, the coherent discriminator does not imply any multiplications of the correlator outputs and, as a consequence, no squaring losses are expected, as will be shown in Section 4.3.1. The stability domain of the coherent discriminator is twice as large as for the DP phase discriminator as it tracks the phase error directly. This discriminator also requires normalization.

An extended arctangent (Atan2) discriminator can also be used for the pilot channel. Its expression is given by Macabiau *et al.* (2003):

$$D_{A \tan 2} = a \tan 2(Q_P, I_P) \quad (4.8)$$

In the absence of disturbances, it equals:

$$D_{A \tan 2} = \varepsilon_\phi \quad (4.9)$$

It has the widest linear tracking range of all, as it covers the entire range  $[-\pi; \pi]$  radians, which allows toleration of a fairly large phase error without loss of lock. It is interesting to note that this discriminator equals the classical arctangent for a phase error within the range  $[-\pi/2; \pi/2]$  radians. No normalization is required for the extended arctangent discriminator. However, the same remarks as those made with respect to the traditional arctangent discriminator apply.

The two discriminators proposed for the pilot channel have lock points separated by  $2\pi$  radians, which means that they can have only full cycle slips. This is of critical importance when the user is concerned with carrier phase ambiguity fixing.

Now that the overview of the discriminators being considered is complete, the following section will focus on the sources of errors that affect all the discriminators previously described.

### **4.3 Sources of Error of a PLL**

The four main sources of errors affecting phase tracking are summarized by Ward (1996): thermal noise, oscillator frequency noise, oscillator vibration, and dynamics. They are all studied in detail in this section.

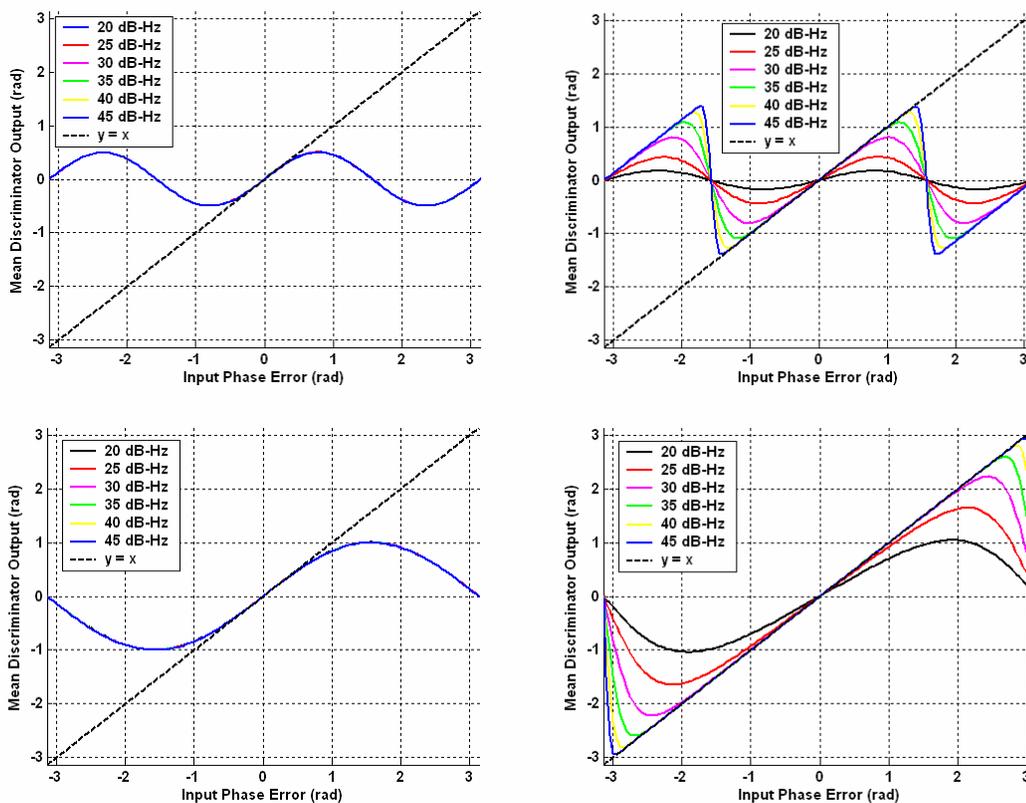
#### **4.3.1 Thermal Noise**

The origin of thermal noise has been discussed in Section 3.1.6. It is interesting, at first instance, to look at the response of the discriminator output to thermal noise stress. The approach used herein is statistical. Since the discriminators use only the I and Q prompt correlator values, it has been observed in Section 2.1.3 that their associated noise components are two independent Gaussian noise values with the same apparent power, as given by Equation (2.16). This is very straightforward to generate. It is then possible to assess, through Monte Carlo simulations, the mean or standard deviation of the discriminator output as a function of the SNR or the  $C/N_0$ . Since it is more common to

use  $C/N_0$  in the GNSS field, it will be used herein as a reference measure for comparison purposes.

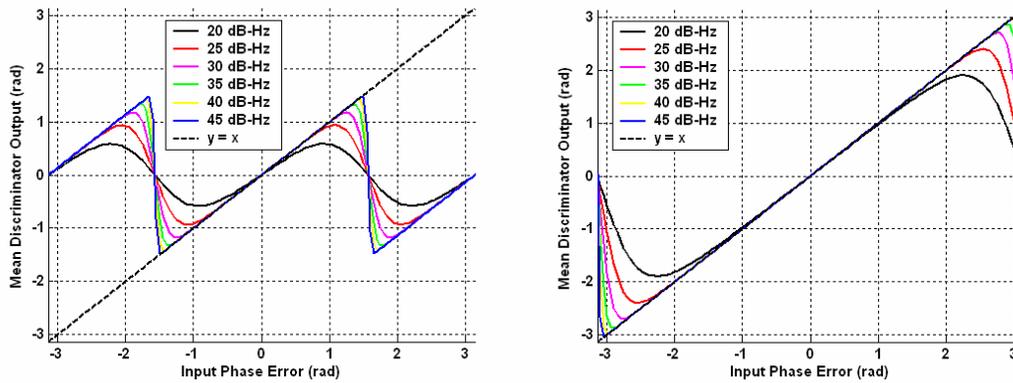
For the DP and coherent discriminators, an ideal normalization profile equal to the discriminator slope at the origin was assumed, which might be overly optimistic, especially in light of low  $C/N_0$  values. However, this strategy always provides an interesting means of assessing optimal tracking performance. Moreover, it is likely that the method and accuracy associated with signal power estimation techniques differs from manufacturer to manufacturer, so adopting an example instead of the ideal case, without attention to other system quantifiers, would be irrelevant.

The results showing each discriminator's mean output are given in Figure 4.2, for a coherent integration time of 4 ms based on simulation of a wide 12 MHz one-sided front-end filter. It can be clearly seen that the performance of the arctangent discriminators is highly dependent upon the  $C/N_0$  value. The lower the  $C/N_0$ , the smaller its linear region will be, and the less likely it is that the discriminator will exhibit the correct response. This might prove to be a problem when high sensitivity is required. On the other hand, the two other discriminators (DP and coherent) are not dependent upon the  $C/N_0$  (all the associated curves in Figure 4.2 are on top of each other), when perfect normalization is assumed. It should be noted, however, that under actual conditions, normalization would also be affected by low  $C/N_0$  values, and would impact the slope of the discriminator output of the DP and coherent discriminators. As a preliminary conclusion based on these observations, it seems that from a linearity domain point-of-view, the choice of the discriminator should be affected by the expected sensitivity of the receiver.



**Figure 4.2 – Mean DP (Top Left), Atan (Top Right), Coherent (Bottom Left) and Atan2 (Bottom Right) Discriminator Output for 6  $C/N_0$  (20, 25, 30, 35, 40, and 45 dB-Hz) and a Coherent Integration Time of 4 ms**

A longer coherent integration would contribute to a better response from the arctangent discriminators. Figure 4.3 shows their mean output based on an integration time of 20 ms. The response characteristics of the arctangent discriminators are significantly improved, with a linear tracking domain having the correct slope, even with  $C/N_0$  as low as 20 dB-Hz for the extended arctangent, and 25 dB-Hz for the classical arctangent.

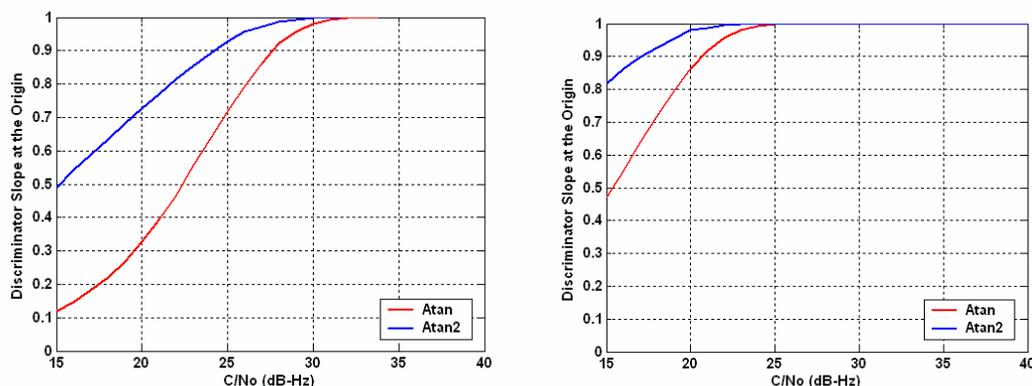


**Figure 4.3 – Mean Atan (Left) and Atan2 (Right) Discriminator Output for 6  $C/N_0$  (20, 25, 30, 35, 40 and 45 dB-Hz) and a Coherent Integration Time of 20 ms**

Figure 4.4 shows the change in slope (at the origin) associated with the classical and extended arctangent discriminators in the respective panels. As expected, it can be seen that for low SNRs, the slope will decrease, degrading both discriminators' performance by reducing their linear tracking region as well as their slope at the origin. It can also be noted that the extended arctangent discriminator seems to better resist Gaussian noise compared to the classical arctangent since its slope at the origin starts decreasing by a few dBs (in terms of  $C/N_0$ ) relative to the one associated with the classical arctangent discriminator. This phenomenon will be explained hereafter.

The discriminator slope at the origin is extremely important since it conditions the possible degree of bias inherent to the discriminator. A low discriminator slope at the origin would certainly imply that tracking is perilous. Consequently, it is possible to set a hard  $C/N_0$  threshold to ensure that the receiver will not try to track the phase if the signal strength is below a certain level, since it might lead to a loss of lock. For example, assuming that a minimum acceptable discriminator slope is 0.9 for robust tracking, it means that the PLL will have a hard  $C/N_0$  threshold of 24 and 27.5 dB-Hz for the

extended and classical arctangent discriminators, respectively, assuming a 4 ms integration time, and 17 and 20.5 dB-Hz assuming a 20 ms integration time.

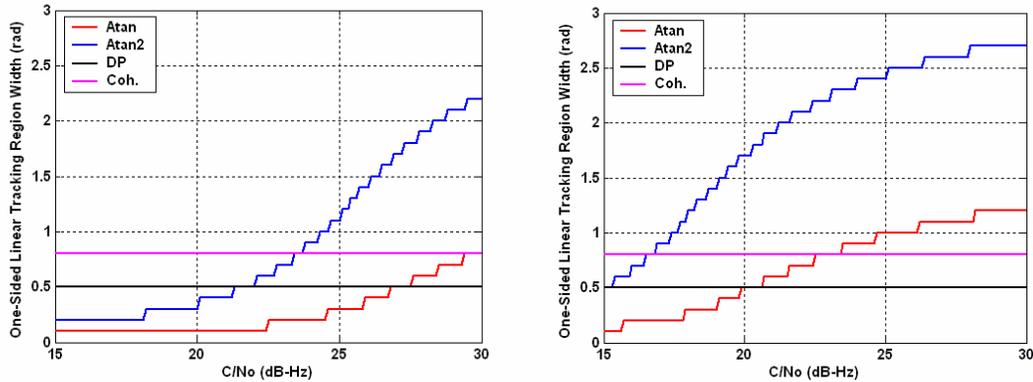


**Figure 4.4 - Atan (Left) and Atan2 (Right) Discriminator Slope at the Origin for a Coherent Integration Time of 4 (Left) and 20 (Right) ms**

It can be observed from Figure 4.2 and Figure 4.3, that the decrease in the extended and classical discriminators' slopes at the origin is also related to a reduction in the width of the discriminator's domain of linearity. The phase discriminator linearity domain will be defined herein as the width of the zone around zero where the discriminator does not differ from the first diagonal by more than 0.1 radians (approximately  $5.7^\circ$ ). For a narrow double-sided linearity domain width of 0.2 radians, a flat discriminator output would still seem like having a linearity domain of 0.2 radians. However, in such a case, the discriminator slope constraint would eliminate this possibility. This definition is of course subjective, since the value '0.1 radians' could be narrowed or enlarged; however, it nevertheless affords a fair representation of the performance of different discriminators.

Figure 4.5 shows the one-sided linearity domain of each discriminator. It shows, when compared to the discriminator's slope, that the linearity domain first shrinks when the  $C/N_0$  decreases, and then the slope at the origin decreases when the linearity domain

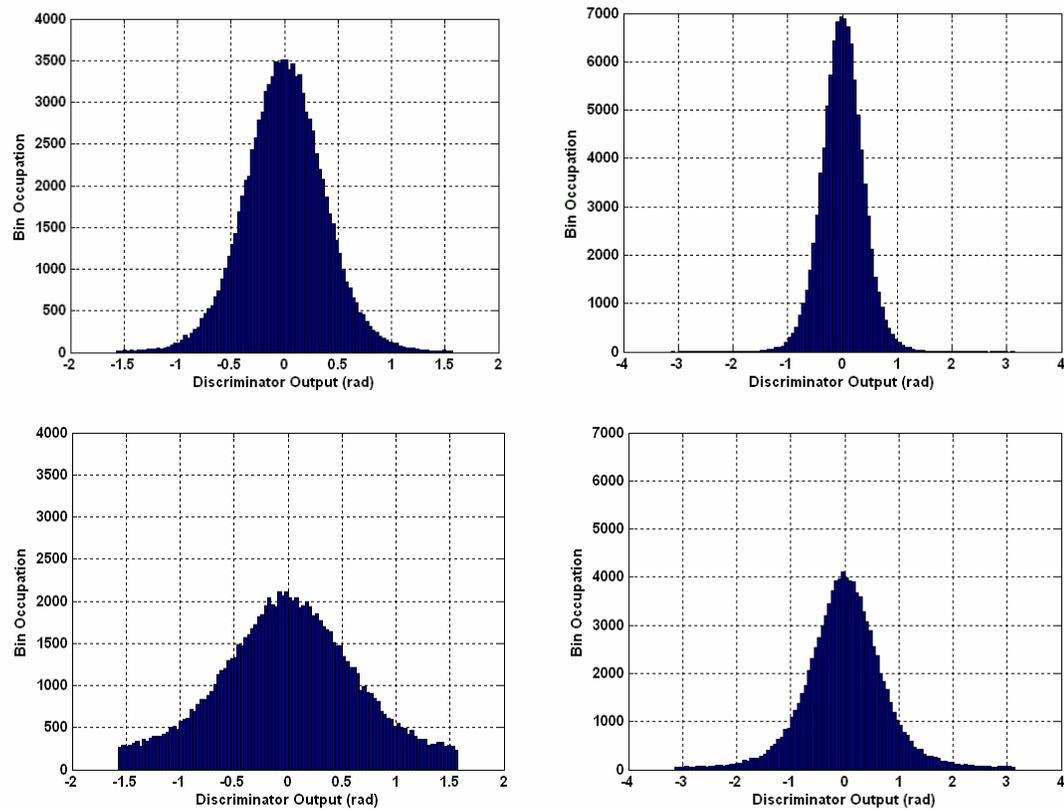
drops below a certain value; this value is evaluated to 1.5 and 0.75 radians for the extended and classical arctangent discriminators, respectively, under the conditions taken for this thesis. It can be observed at this point that these values correspond to half of their respective theoretical linear tracking domains in the absence of disturbances.



**Figure 4.5 – One-Sided Linear Tracking Domain Width for the Four Discriminators Considered and for a Coherent Integration Time of 4 (Left) and 20 (Right) ms**

Before concluding on the choice of the phase discriminator, it is of course extremely important to understand the reason for the arctangent discriminators' behaviour. Figure 4.6 shows the distribution of the classical and extended arctangent discriminators' output for a zero phase error, for a 4 ms integration time, and a  $C/N_0$  of 25 and 30 dB-Hz. These two values correspond to the approximate  $C/N_0$  when the arctangent discriminators' slopes at the origin start to decrease. Figure 4.6 shows that these  $C/N_0$  values also approximately corresponds to the moment when the distribution becomes non-Gaussian due the classical and extended arctangent discriminators' output inherent boundaries,  $[-\pi/2; \pi/2]$  and  $[-\pi; \pi]$  respectively. These limits imply that, with decreasing SNR values, the distribution will become flatter within the discriminator output range such that the discriminator output slope will become flat. The fact that the extended arctangent

discriminator has a wider output range explains its better resistance to noise compared to the classical arctangent. This also clarifies what was observed previously in Figure 4.3 and Figure 4.4.



**Figure 4.6 – Histogram of the Classical (Left) and Extended (Right) Arctangent Discriminators for a  $C/N_0$  of 30 (Top) and 25 (Bottom) for a Coherent Integration Time of 4 ms**

Now that the behaviour of each studied discriminator has been understood, it is of main interest to investigate their role in the overall loop performance. Van Dierendonck *et al.* (1992) showed that, for typical tracking loops, the phase tracking error variance can be written as:

$$\sigma_{\varepsilon_\phi}^2 = 2B_L(1 - 0.5B_L T_I) T_I \sigma_{Discr}^2 \quad (4.10)$$

This expression shows that the behaviour of the tracking loop in Gaussian noise is directly linked to the discriminator response to Gaussian noise. The theoretical PLL phase tracking error variance expressions, when using the DP and coherent discriminators, can be found in the literature for infinite front-end bandwidth (Ward 1996, Hegarty 1999). Their expressions for a limited front-end bandwidth are given herein, using computations very similar to those given in Appendix C, but less complicated:

$$\sigma_{PLL,DP}^2 = \frac{B_L(1-0.5B_L T_I)}{\frac{P}{N_0} \int_{-\infty}^{+\infty} G(f)df} \left( 1 + \frac{1}{2 \frac{P}{N_0} T_I \int_{-\infty}^{+\infty} G(f)df} \right) \quad (4.11)$$

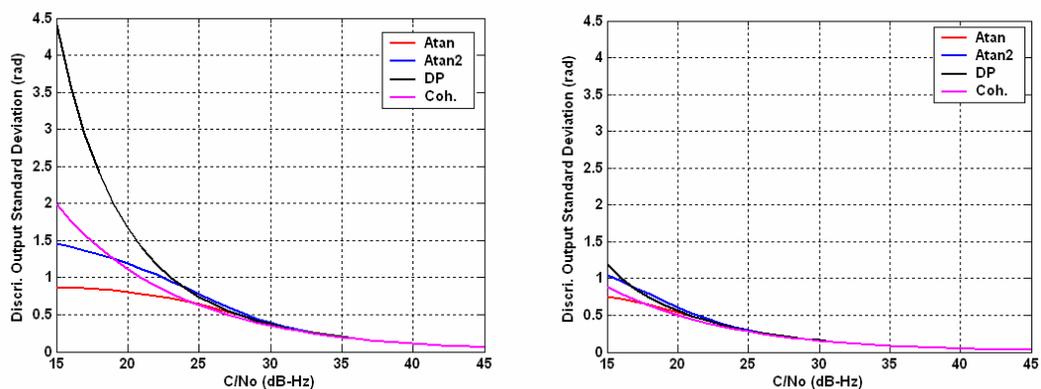
$$\sigma_{PLL,Coh}^2 = \frac{B_L(1-0.5B_L T_I)}{\frac{P}{N_0} \int_{-\infty}^{+\infty} G(f)df} \quad (4.12)$$

Because the phase discriminators simply use the in-phase and quadra-phase prompt correlation values, the filter loss due to the front-end has an impact only on the equivalent  $C/N_0$  values (neglecting the I&D effect). Since the coherent discriminator does not originate from the product of the correlation outputs, it does not suffer from squaring losses. It should then perform better under low SNR values than the other discriminators that are based on the multiplication, or division, of several correlators' outputs.

With respect to the arctangent discriminators, because of the arctangent operator, a theoretical expression of the PLL phase error variance would be extremely difficult to obtain. However, in general, it is assumed that it is close to the DP discriminator for

practical purposes (Van Dierendonck 1997). One way to check this assumption is to statistically compute the discriminator output variance under Gaussian noise using Monte Carlo simulations, and then use the expression given by Equation (4.10).

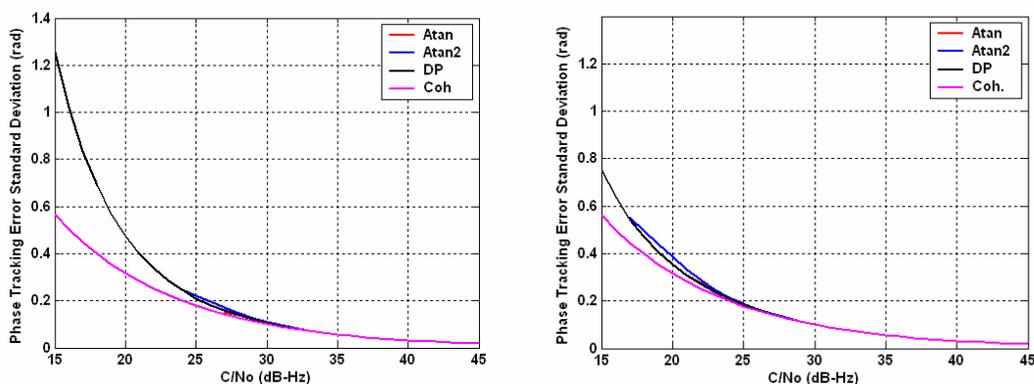
It is now known that the shape of the discriminator output will impact the discriminator output error variance. Consequently, one must be careful when computing the variance of the arctangent discriminator output, as the linear tracking region of the arctangent discriminators is dependent upon both the  $C/N_0$  and the integration time. As already seen, this might lead to serious problems for arctangent discriminators. Figure 4.7 shows the output standard deviation of the four considered discriminators for coherent integration times of 4 and 20 ms. It can be seen that the classical and extended arctangent discriminators have curves with an inflexion. This inflexion corresponds to the point when the discriminator output distribution is constrained by the arctangent output boundaries and hence, also relates to the  $C/N_0$  at which the slope at the origin begins to decrease.



**Figure 4.7 – Discriminator Output Standard Deviation for a 4 (Left) and 20 (Right) ms Coherent Integration Time**

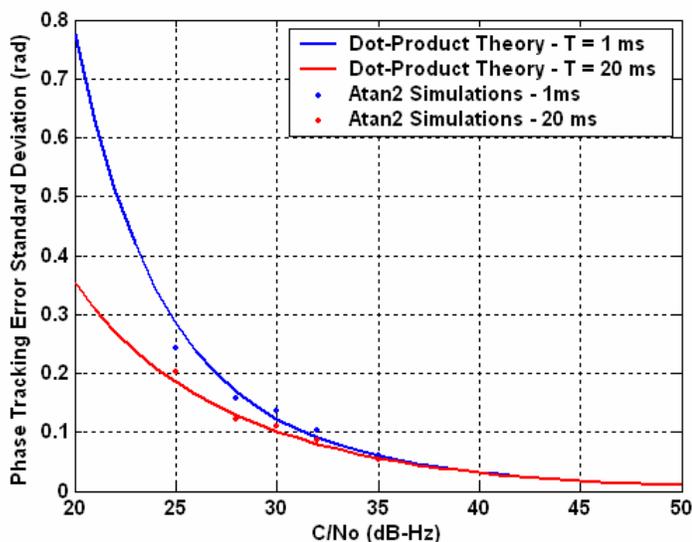
When considering the overall PLL tracking error standard deviation, it is relevant to constrain the curves to values of  $C/N_0$  greater than the hard  $C/N_0$  discriminator threshold defined by the  $C/N_0$  at which the discriminator slope at the origin deviates from the first diagonal. This is done in Figure 4.8 using hard  $C/N_0$  threshold values corresponding to a minimum acceptable discriminator slope at the origin equal to 0.9. The results are shown for the four discriminators presented with integration times of 4 and 20 ms, and a PLL loop bandwidth of 10 Hz.

It appears that for both cases (4 and 20 ms coherent integration durations), the PLL phase error standard deviation using the arctangent discriminators very closely follows the phase error associated with the DP discriminator. The extended arctangent seems to diverge (higher) just before it reaches its hard  $C/N_0$  threshold. Figure 4.9 confirms, through 20-second simulations, that the arctangent discriminator leads to the same phase error standard deviation as with the DP discriminator.



**Figure 4.8 – PLL Phase Tracking Standard Deviation using One of the Four Discriminators Considered for Integration Times of 4 and 20ms and a 10 Hz Loop Bandwidth**

Figure 4.8 confirms that the coherent discriminator will provide the most accurate phase tracking. However, for a long coherent integration time, it will not produce a significant improvement for typical levels of  $C/N_0$ .



**Figure 4.9 – Comparison of Theory and Simulations for Extended Arctangent Discriminator Phase Tracking Error Standard Deviation for Integration Times of 1 and 20 ms and a 10 Hz Loop Bandwidth**

#### 4.3.2 Dynamic Stress Error

Due to the short wavelength of L1 signals ( $\sim 19$  cms), the PLL will be very susceptible to user dynamics since it can rapidly lead to a error greater than the linear tracking domain boundary. Consequently, unless the dynamic pattern of the user is well known, it is preferable to use a high order loop filter. This allows for the ability to track most of the dynamics without bias. In the case of GNSS PLLs, a third order loop is generally used to account for any kind of signal dynamics. This assumption will be applied in the following sections and implies that the loop will be affected, in terms of dynamics, only by jerk and

higher order dynamics. In such a case, the dynamic stress error can be expressed as (Bastide 2004):

$$\theta_e = 2\pi \frac{T_I^3}{K_3} \frac{dR^3}{dt^3} (\text{rad}) \quad (4.13)$$

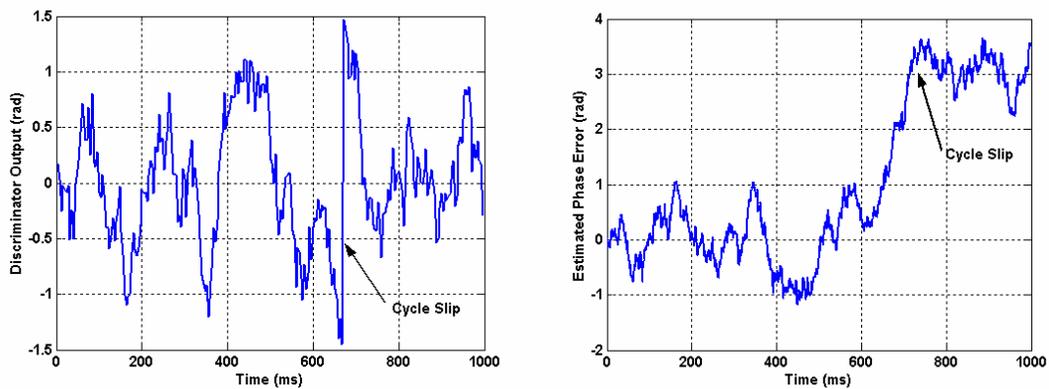
where  $K_3$  corresponds to the coefficient given by Stephens & Thomas (1995), in their description of discrete-update PLLs, and

$\frac{dR^3}{dt^3}$  is the third order component of the carrier phase delay (cycle/sec<sup>3</sup>).

Another potential problem intrinsic to high dynamics, and which is not included in Equation (4.13), is the possibility that the change in the propagation time between the satellite and the receiver will induce a phase change that is greater than the discriminator linear tracking range during one coherent integration period. To quantify this problem, a phase variation of  $\pi$  radians within a time of 1, 10, 20 and 100 ms is reached with relative velocities (not including the estimated Doppler) between the satellite and receiver (in the satellite-receiver direction) of 342, 34.2, 17.1, and 3.4 km/h, respectively. This can present a threat if such a level of dynamics remains stable for long periods of time. This can entail a substantial burden for long coherent integration periods if no external velocity aiding is provided.

It must also be emphasized that the two arctangent discriminators impose the drawback of a sharp mean discriminator output shape at the boundaries of their stability domain for high SNRs, as seen in Figure 4.2 and Figure 4.3. Indeed, if because of dynamics, the input phase error approaches the arctangent stability domain boundary, it would be then

straightforward to undergo a cycle slip since it is possible that the discriminator estimate turns into the opposite of what it should be due to noise. This is illustrated in Figure 4.10 for a high  $C/N_0$  (40 dB-Hz), modeling a low quality receiver oscillator (Quartz - see Section 3.3.2 for details), and a classical arctangent as the discriminator used by the PLL. It can be seen that the cause of the cycle slip is the sudden change of sign of the arctangent discriminator output. This risk must be taken into account when designing a PLL.



**Figure 4.10 – Arctangent Discriminator Output and Estimated PLL Phase Error Using a 40 dB-Hz Signal and a Quartz Oscillator as the Receiver Oscillator**

### 4.3.3 Oscillator Frequency Noise

As explained in Section 3.1.2, the oscillator frequency noise is the result of the instability of the central frequency of the oscillator that produces a phase jitter at the local carrier replica level. For a third order loop, the induced tracking error variance can be expressed as (Irsigler & Eissfeller 2001):

$$\sigma_{PLL,Osc}^2 = 2\pi^2 f_{Osc}^2 \left( \frac{\pi^2 h_{-2}}{3\omega_L^3} + \frac{\pi h_{-1}}{3\sqrt{3}\omega_L^2} + \frac{h_0}{6\omega_L} \right) (\text{rad}^2) \quad (4.14)$$

where  $f_{Osc}$  is the central frequency of the oscillator,

$h_{-2}$ ,  $h_{-1}$ , and  $h_0$  are coefficient characterizing the different components of the oscillator frequency noise and were introduced in Section 3.1.2, and  $\omega_L \cong 1.27B_L$  is for a third order loop (valid only if  $B_L T_I \ll 1$ ).

It has to be noted that the formula shown considers that the PLL can be approximated using the continuous-update theory. This means that, for values of  $B_L T_I$  that are not negligible (typically greater than 0.1), the theory expressed above might not be valid. Therefore, special care must be taken in such cases. This is a very important point since the PLL loop bandwidth is usually in the order of 8 to 30 Hz, meaning that only very short integration periods are included in this theory.

It can also be noted that, in Equation (4.14), the phase tracking error induced by the oscillator frequency noise is dependent upon the loop bandwidth  $B_L$ . This is understandable since  $B_L$  is supposed to control the level of dynamics the PLL can follow. Consequently, any phase variation can be assimilated to signal dynamics and a higher PLL loop bandwidth means better modeling of the oscillator phase error.

Moreover, Equation (4.14) is valid only if the phase error remains within the linearity domain. Indeed, if it does not, the error estimation will be erroneous and will likely lead to a cycle slip. When considering this possibility, the oscillator frequency noise is the

source of error that would most likely create such an event since it constantly changes the phase of the replica from its original value. Because different clocks will exhibit very different behaviours in this regard, this is a troublesome quantity to determine. However, this possibility should be roughly assessed before implementing a PLL. Winkel (2003) gives an example of five different oscillators (previously illustrated in Table 3.1), as well as their Allan variance over a large range of update rates. Table 4.1 and Table 4.2 summarize the standard deviation and 95% phase change rate during different integration periods obtained through simulation, respectively. The results given in the two tables are obtained using Winkel's (2003) model. Table 4.1 and Table 4.2 show that, even when user dynamics are negligible, oscillator instability can pose a threat to PLL tracking if not taken into account.

**Table 4.1 – Standard Deviation of the Phase Change Rate (rad) during a Coherent Integration for Various Oscillators**

	Coherent Integrations Time (ms)					
	1	10	20	50	100	1000
<b>Quartz</b>	<b>0.14</b>	<b>0.44</b>	<b>0.63</b>	<b>0.99</b>	<b>1.41</b>	<b>6.82</b>
<b>TCXO</b>	<b>0.01</b>	<b>0.03</b>	<b>0.05</b>	<b>0.09</b>	<b>0.19</b>	<b>5.11</b>
<b>OCXO</b>	<b>0.09</b>	<b>0.28</b>	<b>0.40</b>	<b>0.63</b>	<b>0.89</b>	<b>3.23</b>
<b>Rubidium</b>	<b>0.04</b>	<b>0.14</b>	<b>0.20</b>	<b>0.31</b>	<b>0.45</b>	<b>2.33</b>
<b>Caesium</b>	<b>0.10</b>	<b>0.31</b>	<b>0.44</b>	<b>0.70</b>	<b>1.0</b>	<b>3.82</b>

**Table 4.2 – 95 % Phase Change Rate (rad) during a Coherent Integration for Various Oscillators**

	Coherent Integrations Time (ms)					
	1	10	20	50	100	1000
<b>Quartz</b>	<b>0.23</b>	<b>0.73</b>	<b>1.03</b>	<b>1.62</b>	<b>2.31</b>	<b>11.10</b>
<b>TCXO</b>	<b>0.02</b>	<b>0.05</b>	<b>0.08</b>	<b>0.15</b>	<b>0.31</b>	<b>8.49</b>
<b>OCXO</b>	<b>0.15</b>	<b>0.46</b>	<b>0.65</b>	<b>1.03</b>	<b>1.46</b>	<b>5.31</b>
<b>Rubidium</b>	<b>0.07</b>	<b>0.23</b>	<b>0.33</b>	<b>0.52</b>	<b>0.74</b>	<b>3.87</b>
<b>Caesium</b>	<b>0.16</b>	<b>0.51</b>	<b>0.73</b>	<b>1.15</b>	<b>1.66</b>	<b>6.29</b>

According to the values presented, assuming that the phase change has to remain within the interval  $\left[\frac{\pi}{4}; \pi\right]$  radians (which corresponds to the ideal linear tracking range of the various discriminators) during the integration duration, the approximate maximum integration times should be restricted to the values shown in Table 4.3. It must be understood that the values given in the table overestimate the actual maximum integration time, since these values assume that the PLL responds perfectly to each phase change, which is not true in reality, due to the noise and the inherent loop response lag. The maximum coherent integration time allowed would conceivably be significantly reduced when the SNR is low.

**Table 4.3 – Maximum Coherent Integration Times due to the Oscillator Frequency Noise for Various Oscillators and a Maximum Tolerable Phase Error of  $\frac{\pi}{4}$  and  $\pi$**

	Maximum Coherent Integration Time Allowed (ms)	
	Linear Region = $\frac{\pi}{4}$ rad	Linear Region = $\pi$ rad
<b>Quartz</b>	<b>10</b>	<b>200</b>
<b>TCXO</b>	<b>200</b>	<b>500</b>
<b>OCXO</b>	<b>25</b>	<b>450</b>
<b>Rubidium</b>	<b>100</b>	<b>850</b>
<b>Caesium</b>	<b>20</b>	<b>400</b>

It can be seen, according to Table 4.3, that the phase variation during one coherent integration period is a limiting factor for long integration times. It is also interesting to see that expensive high quality oscillators with good long-term performance could be not optimal for use in a GPS receiver. However, it has to be kept in mind that the values shown are simply meant to give an appreciation of the scale of the maximum coherent integration durations for different oscillators and several discriminator linear tracking

region widths. It can be seen that it is tremendously important in PLL design to provide a wide tracking region.

#### 4.3.4 Oscillator Vibrations

For a third order loop, the phase tracking error standard deviation resulting from the oscillator vibration can be expressed as Irsigler & Eissfeller (2001):

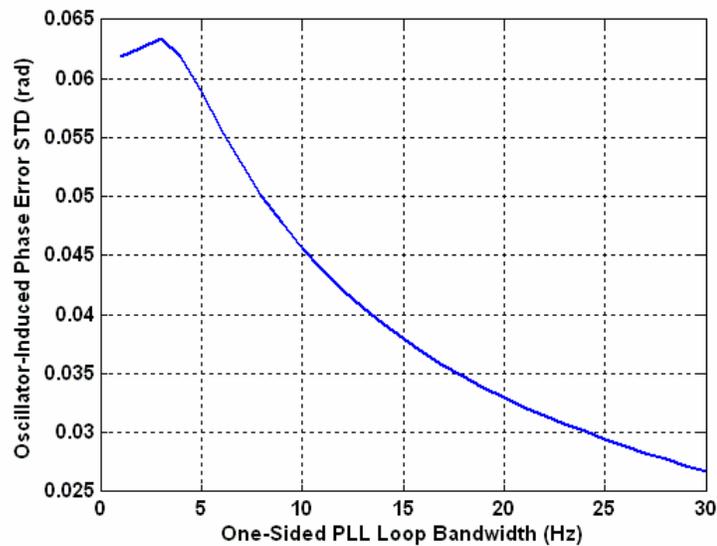
$$\sigma_{PLL,Vib}^2 = \frac{2\pi f_{osc}^2 k_g^2 G_g}{\omega_L} \left[ \frac{1}{3} \left( \arctan\left(\frac{\omega_2}{\omega_L}\right) - \arctan\left(\frac{\omega_1}{\omega_L}\right) \right) \right. \\ \left. + \frac{1}{6} \left( \arctan\left(-\frac{\omega_L\sqrt{3} + 2\omega_2}{\omega_L}\right) \right. \right. \\ \left. \left. + \arctan\left(\frac{\omega_L\sqrt{3} + 2\omega_2}{\omega_L}\right) \right. \right. \\ \left. \left. - \arctan\left(-\frac{\omega_L\sqrt{3} + 2\omega_1}{\omega_L}\right) \right. \right. \\ \left. \left. - \arctan\left(\frac{\omega_L\sqrt{3} + 2\omega_1}{\omega_L}\right) \right) \right. \\ \left. + \frac{1}{4\sqrt{3}} \ln \left( \frac{(\omega_2^2 - \sqrt{3}\omega_L\omega_2 + \omega_L^2)(\omega_1^2 + \sqrt{3}\omega_L\omega_1 + \omega_L^2)}{(\omega_1^2 + \sqrt{2}\omega_L\omega_1 + \omega_L^2)(\omega_2^2 + \sqrt{3}\omega_L\omega_2 + \omega_L^2)} \right) \right] \quad (4.15)$$

where  $G_g$  is the vibration PSD assumed to be constant within  $[\omega_1; \omega_2]$  ( $\text{g}^2/\text{Hz}$ ), and

$k_g$  is the oscillator's g-sensitivity in parts-per-g (1/g).

As in the case of the oscillator frequency noise error, the higher the PLL loop bandwidth, the lower the impact of vibration-induced error on carrier tracking.

The vibration-induced oscillator error in the PLL tracking sensitivity study presented in Section 4.4, was taken according to work done by Irsigler & Eissfeller (2001). Consequently,  $G_g$  was taken to be equal to  $0.05 \text{ g}^2/\text{Hz}$  within 25 and 2500 Hz, and the oscillator's sensitivity was set to  $2e^{-10}$  parts/g. The resulting error is shown in Figure 4.11. It has to be noted that as in the case of the oscillator frequency noise, Equation (4.15) was obtained with a continuous-update assumption.



**Figure 4.11 – Modeled Vibration-Induced Phase Tracking Error Standard Deviation**

#### 4.3.5 Conclusion on Phase Tracking Error Modeling

The various sources of error have been theoretically studied and, consequently, their magnitude assessed; however, it is very important to keep in mind that the PLL is extremely vulnerable to disturbances due to its truncated linear tracking domain. It must be borne in mind that the limited linear domain also has to be considered, since the formulas used assume a perfect discriminator response for any input error. For instance,

high dynamics, if implying a phase error within the integration time that is significantly higher than the discriminator linear domain, will lead to a loss of lock or cycle slips. Consequently, it is dangerous, unless in stable conditions, to use long coherent integration periods. In general, when the level of dynamics is unknown, it is recommended to lower the integration time as much as possible in order not to lose lock too rapidly. To follow high dynamics, it is also important to try to increase the value of the loop bandwidth as much as possible, although this is detrimental for noise mitigation.

The other important issue is that the formulas given for the oscillator phase errors are based on the approximation of analog filtering. For long integration times, the theoretical expression might not hold and the resulting phase error might diverge from the theory. For this reason, and to limit the value of  $B_L T_I$  while still being able to study the impact of a pilot channel, it has been decided to limit the integration duration to only 20 ms in order to produce relevant and instructive conclusions. For longer integration times, Monte Carlo testing would probably be necessary, based on specific operational estimates.

It is also important to realize that the study of the PLL presented herein is meant to minimize loss of lock and cycle slipping. Indeed, a few cycle slips will not appreciably impact the overall tracking, providing that the code tracking loop maintains lock and that the frequency error is not too significant, since the PLL needs only the prompt correlator values. However, PLL phase measurements can be used to obtain precise positioning through carrier-smoothing of the code measurements, or carrier-phase positioning. In this case, cycle slips can be very harmful. It is in the latter case that this thesis intends to

study the PLL. In other cases, the tracking thresholds would probably be slightly lower. A method to assess the cycle slip probability is given by Holmes (1982) and was used by Hegarty (1999).

#### 4.4 Sensitivity Analysis

A PLL tracking sensitivity analysis is usually done for GPS using the condition given by Ward (1996):

$$3\sigma_{\varphi} + \theta_e \leq \frac{\pi}{4} \quad (4.16)$$

where  $\sigma_{\varphi} = \sqrt{\sigma_{PLL,t}^2 + \sigma_{PLL,Vib}^2 + \sigma_{PLL,Osc}^2}$ , and

$\theta_e$  is the dynamic stress error.

In such a case, the discriminator is assumed to be a DP, and then the value  $\pi/4$  corresponds to the approximate one-sided linear tracking region. To adapt this condition to a different discriminator, Equation (4.16) becomes:

$$3\sigma_{\varphi} + \theta_e \leq \frac{L_{\varphi}}{2} \quad (4.17)$$

where  $L_{\varphi}$  is the two-sided phase discriminator linear tracking region.

The linear tracking region has been defined herein as the region around the zero input phase error where the mean discriminator output does not differ from the first diagonal by more than 0.1 radians. As previously mentioned, the value of 0.1 radians is arbitrary and

other values could be used to comparable effect. It should, however, bolster an understanding of which discriminator to use, and under which conditions.

Considering the PLL phase tracking error due to Gaussian noise given in Equations (4.11) and (4.12) (the phase tracking error variance obtained using the arctangent discriminators is assumed to equal the one obtained using the DP discriminator), the tracking threshold can then be easily calculated, yielding:

$$\left(\frac{C}{N_0}\right)_{Th} = A \quad \text{using the coherent discriminator, and}$$

$$\left(\frac{C}{N_0}\right)_{Th} = \frac{1 + \sqrt{1 + \frac{2A}{T_I}}}{2A} \quad \text{using the other discriminators} \quad (4.18)$$

$$\text{where } A = \frac{1}{B_L} \left( \frac{1}{9} \left( \frac{L_\varphi}{2} - \theta_e \right)^2 - \sigma_{PLL, Vib}^2 - \sigma_{PLL, Osc}^2 \right)$$

Taking into account the actual decrease of the linear region of the arctangent discriminators, it is possible to estimate the threshold for tracking in terms of  $C/N_0$  for specific configurations. This will be done only in the case of the pilot channel for different oscillators: a maximum of three jerk conditions (0, 1 and 2 g/s) and two coherent integration times of 4 and 20 ms (limited to these cases for the reasons mentioned in Section 4.3.5).

The linear tracking region is computed with the same method as used in Section 4.3.1. It should be kept in mind that the slope at the origin plays an important role in the tracking phase error standard deviation due to noise. Considering that tracking will be impaired

for  $C/N_0$  that will lead to a discriminator slope at the origin lower than 0.9 (or 1 depending on the expected phase tracking robustness), Table 4.4 shows the corresponding hard  $C/N_0$  threshold values for the two arctangent discriminators.

**Table 4.4 - Hard  $C/N_0$  Threshold corresponding to the Discriminator Slope at the Origin being Lower than 0.9 (Left Value) and 1 (Right Value)**

	<b><math>C/N_0</math> Hard Thresholds</b>	
	<b>Atan2</b>	<b>Atan</b>
<b>4 ms</b>	<b>24 - 30</b>	<b>27.5 – 31.5</b>
<b>10 ms</b>	<b>20 – 26</b>	<b>23.5 – 27.5</b>
<b>20 ms</b>	<b>17 – 23</b>	<b>20.5 – 24.5</b>
<b>30 ms</b>	<b>15.5 – 21.5</b>	<b>19 - 23</b>

Table 4.5 and Table 4.6 show the PLL tracking thresholds with the associated loop bandwidth for integration times of 4 and 20 ms and using the error models described throughout Section 4.3. The values in blue stem from the fact that the sensitivity of the loop is limited by the hard  $C/N_0$  threshold inherent in the use of the arctangent discriminator.

For the pilot channel it can be seen that, according to the integration time, the PLL sensitivity will vary based on discriminators and oscillator behaviour. It can be seen that generally for short integration times, and assuming use of an accurate oscillator (TCXO and Rubidium), the coherent discriminator will provide the highest sensitivity. Assuming that the extended arctangent hard threshold for tracking is the upper boundary given in Table 4.4, use of the coherent discriminator affords a significant advantage. This is enhanced more so by the fact that the values shown assume constant dynamics equal to the specified maximum expected jerk. In reality, for most of the time, the dynamics will be lower, and the sensitivity of the PLL higher.

For longer integration times, the difference between the pilot discriminators decreases, since the extended arctangent squaring losses are reduced and, consequently, the effect of the noise on the mean discriminator output shape is reduced thereby significantly lowering its hard  $C/N_0$  threshold. Moreover, the range of PLL loop bandwidth values that enables the highest sensitivity is larger when using the arctangent discriminator. This is important because, as mentioned above, the values shown assume constant dynamics according to the specified jerk. This means that the PLL loop bandwidth should be configured according to the maximum expected jerk, in order to be sure to enable tracking when these conditions are obtained. However, for most of the time the dynamics will be lower than that and consequently, if the highest sensitivity during maximum dynamics can be obtained with the same loop bandwidth as for lower dynamics, this would ensure a low sensitivity on all occasions. It can be seen in Table 4.5 and Table 4.6 that this is more likely when using the extended arctangent discriminator.

Therefore, it can be anticipated that for long coherent integration times ( $> 10$  to  $20$  ms) using the extended arctangent discriminators would be very beneficial. Moreover, it should be kept in mind that, with low disturbances, the extended arctangent discriminator linear tracking region is extremely wide and thus for high SNR values, it would better prevent loss of lock resulting from high dynamics.

Concerning the data channel, it seems that for fairly good oscillators (TCXO and Rubidium), it is preferable to use the DP discriminator. However, it is very sensitive to significant error levels, since its linear tracking domain is very narrow, and it might not be robust in cases of high dynamics (including oscillator dynamics). On the other hand,

for longer integration times, it seems like the classical arctangent would be preferable by far, since its hard  $C/N_0$  threshold would then be lower than the PLL tracking threshold using the DP discriminator in any configuration. However, as mentioned in Section 4.3.2, the arctangent discriminator has the drawback of having a very sharp mean discriminator output at its stability domain boundaries for high SNR values. This means that when dynamic stress is present, a small amount of noise might cause the PLL to undergo a cycle slip while, if the mean discriminator output was more rounded, it might have been avoided. This is also important, as it implies that the arctangent discriminators (this point is valid for both the classical and extended arctangents discriminators) might not be desirable for high dynamic conditions if there is a chance that the resulting phase error goes around their stability domain boundaries (as already illustrated in Figure 4.10).

When comparing phase tracking sensitivity, it seems that the use of a pilot channel would be extremely beneficial. Indeed, it provides more than 6 dB of improvement under any conditions, which is extremely important since the PLL is usually the weak part of a GNSS receiver. According to the error models used, the expected sensitivity of Galileo L1F phase tracking for 4 ms coherent integration periods is expected to be around 20 to 30 dB-Hz depending upon the oscillator used while for the GPS C/A signal, it would be greater than 27 dB-Hz. For integration times of 20 ms, the Galileo L1F PLL sensitivity is in the low 20 dBs in any of the shown conditions, while it is around 25 dB for the GPS C/A code. These differences are mainly due to the wider linearity domain offered by the pilot discriminators (and to the better noise mitigation in case of the coherent discriminator).

**Table 4.5 – Tracking Thresholds for the Coherent, DP, Atan and Atan2 Discriminators Assuming a Coherent Integration of 4 ms, and Three Levels of Dynamics (Jerk = 0, 1, 2 g/s)**

		Jerk = 0 g/s		Jerk = 1 g/s		Jerk = 2 g/s	
		B <sub>L</sub> (Hz)	Tracking Thresh. (dB-Hz)	B <sub>L</sub> (Hz)	Tracking Thresh. (dB-Hz)	B <sub>L</sub> (Hz)	Tracking Thresh. (dB-Hz)
<b>Coh.</b>	<b>Quartz</b>	26->30	30	>30	<30	28->30	31
	<b>TCXO</b>	5-6	20	12-19	24	16-23	25
	<b>OCXO</b>	11-24	26	17->30	27	25->30	27
	<b>Rubidium</b>	4-5	20	13-23	25	15->30	26
<b>DP</b>	<b>Quartz</b>	N/A	>35	N/A	>35	N/A	>35
	<b>TCXO</b>	7-12	27	14-28	30	17->30	31
	<b>OCXO</b>	N/A	>35	N/A	>35	N/A	>35
	<b>Rubidium</b>	8-16	29	16->30	31	20->30	32
<b>Atan</b>	<b>Quartz</b>	18->30	30	21->30	30	24->30	30
		13->30	31.5	16->30	31.5	18->30	31.5
	<b>TCXO</b>	5-21	27.5	11->30	28	15->30	28
		3->30	31.5	9->30	31.5	10->30	31.5
	<b>OCXO</b>	9->30	29	15->30	29	18->30	29
		5->30	31.5	11->30	31.5	13->30	31.5
	<b>Rubidium</b>	5-16	27.5	13->30	28	17-29	28
		3->30	31.5	10->30	31.5	11->30	31.5
<b>Atan2</b>	<b>Quartz</b>	8->30	25	12->30	25	14->30	25
		5->30	30	10->30	30	11->30	30
	<b>TCXO</b>	3->30	24	8->30	24	13-23	24
		3->30	30	7->30	30	8->30	30
	<b>OCXO</b>	4-28	24	11-27	24	15-26	24
		2->30	30	3->30	30	10->30	30
	<b>Rubidium</b>	3-26	24	10-25	24	14-23	24
		2->30	30	2-30	30	8->30	30

**Table 4.6 – Tracking Thresholds for the Coherent, DP, Atan and Atan2 Discriminators Assuming a Coherent Integration of 20 ms, and Three Levels of Dynamics (Jerk = 0, 1, 2 g/s)**

		Jerk = 0 g/s		Jerk = 1 g/s		Jerk = 2 g/s	
		B <sub>L</sub> (Hz)	Tracking Thresh. (dB-Hz)	B <sub>L</sub> (Hz)	Tracking Thresh. (dB-Hz)	B <sub>L</sub> (Hz)	Tracking Thresh. (dB-Hz)
Coh.	Quartz	26->30	30	>30	<33	N/A	>35
	TCXO	5-6	20	17-16	26	22->30	28
	OCXO	11-24	26	27->30	28	29->30	30
	Rubidium	3-5	20	17->30	27	27->30	28
DP	Quartz	N/A	>35	N/A	>35	N/A	>35
	TCXO	7-10	26	20->30	32	N/A	>35
	OCXO	N/A	>35	N/A	>35	N/A	>35
	Rubidium	8-19	29	24->30	33	N/A	>35
Atan	Quartz	13-26	26	24->30	26	22-30	27
	TCXO	4-11 4->30	22 24.5	14-24	24.5	16->30	26
	OCXO	6-25	24.5	14->30	26	21->30	26
	Rubidium	3-8 3->30	22 24.5	18-19	24.5	17->30	26
Atan2	Quartz	9-11 6->30	19 23	19-21 14->30	21 23	23-28 18->30	22 23
	TCXO	1-10 2->30	17 23	11-21 9->30	20 23	16-25 13->30	21 23
	OCXO	4-7 2->30	17 23	12-27 10->30	21 23	16->30 15->30	22 23
	Rubidium	1-10 1->30	17 23	12-19 9->30	20 23	17-23 14-23	21 23

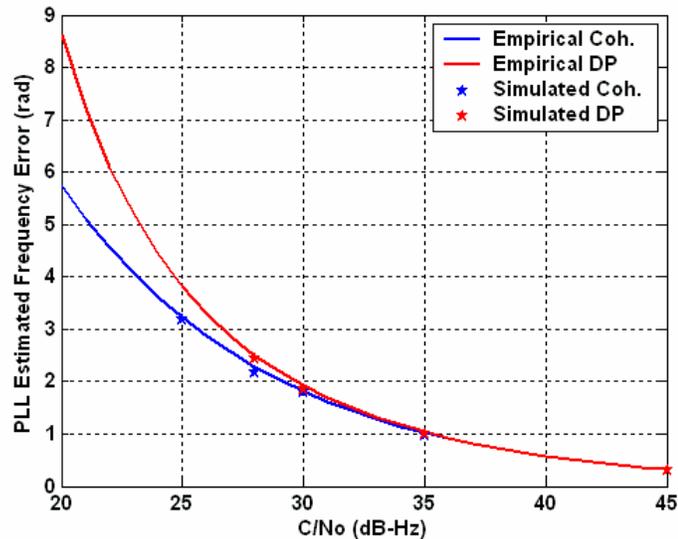
Once again, it is important to keep in mind that the values shown in Table 4.5 and Table 4.6 are based on assumptions that could differ from reality (especially concerning the nature of oscillator phase error, which varies from one oscillator to the next, and the expected maximum dynamics). Consequently, the values have to be seen more as indicators that convey an appreciation of the impact of each parameter, rather than as definitive in terms of magnitude.

## 4.5 PLL Frequency Error

Appendix B shows that the frequency estimation error obtained from the PLL to drive the carrier generation can be expressed as:

$$\sigma_{\varepsilon_f}^2 = \frac{\kappa 4\pi^2 B_L^2}{3} \sigma_{\varepsilon_\phi}^2 \quad (4.19)$$

As a result, it is also directly linked to the discriminator used. Figure 4.12 shows the frequency error standard deviation for a PLL using coherent and DP discriminators and a loop bandwidth of 10 Hz. For a 10 Hz loop bandwidth, the coefficient  $\kappa$  has been estimated to be equal to 4.



**Figure 4.12 –PLL Frequency Error Standard Deviation using a Coherent and a DP Discriminator for a Loop Bandwidth of 10 Hz, and a Coherent Integration Time of 4 ms**

These results are important since they demonstrate the typical frequency estimation error if the PLL is used to assess the Doppler. Moreover, as will be seen in the next chapter,

the PLL is often used as an aid to code tracking. Consequently, having an idea as to the accuracy of this aiding is important when designing the DLL. Finally, it has been shown that the correlation values used by the tracking loops are affected by the frequency accuracy of the carrier wipe-off. It is then of major importance to know the estimated frequency error in order to have an idea of the degradation, as a function of the SNR, that the correlator output will undergo.

#### **4.6 Data/Pilot Implementation**

As explained above, Galileo L1F has both pilot and data channels. If both are tracked using the same local carrier, they can provide two independent estimates of the same phase error that could be of great use in attempting to improve the accuracy of phase measurements.

One must realize that since phase tracking associated with the pilot channel has a tracking threshold significantly lower than phase tracking associated with the data channel (as shown in Section 4.4), the data channel cannot be used at all times to reduce impairment of overall phase tracking. Indeed, for low SNR values, the data discriminator outputs will become extremely noisy and thus become irrelevant to use. A trade-off between reliability and accuracy has to be drawn. Since the pilot PLL significantly outperforms the data PLL, the first realistic implementation is to use only the pilot channel to track the carrier phase. By doing so, the most straightforward implementation is realized, and loop performance can be assessed from the study described in Sections 4.4 and 4.5.

However, for applications where the accuracy of carrier phase measurements is the main issue, it is tempting to try to use more advanced implementations that can lead to higher levels of accuracy. Indeed, the data and pilot information could be combined to further reduce inherent measurement noise, since they estimate exactly the same quantity. Hegarty (1999) proposed a combination at the discriminator level given by:

$$D_{Comb1} = \gamma D_{Data} + \kappa D_{Pilot} \quad \text{with } \gamma + \kappa = 1 \quad (4.20)$$

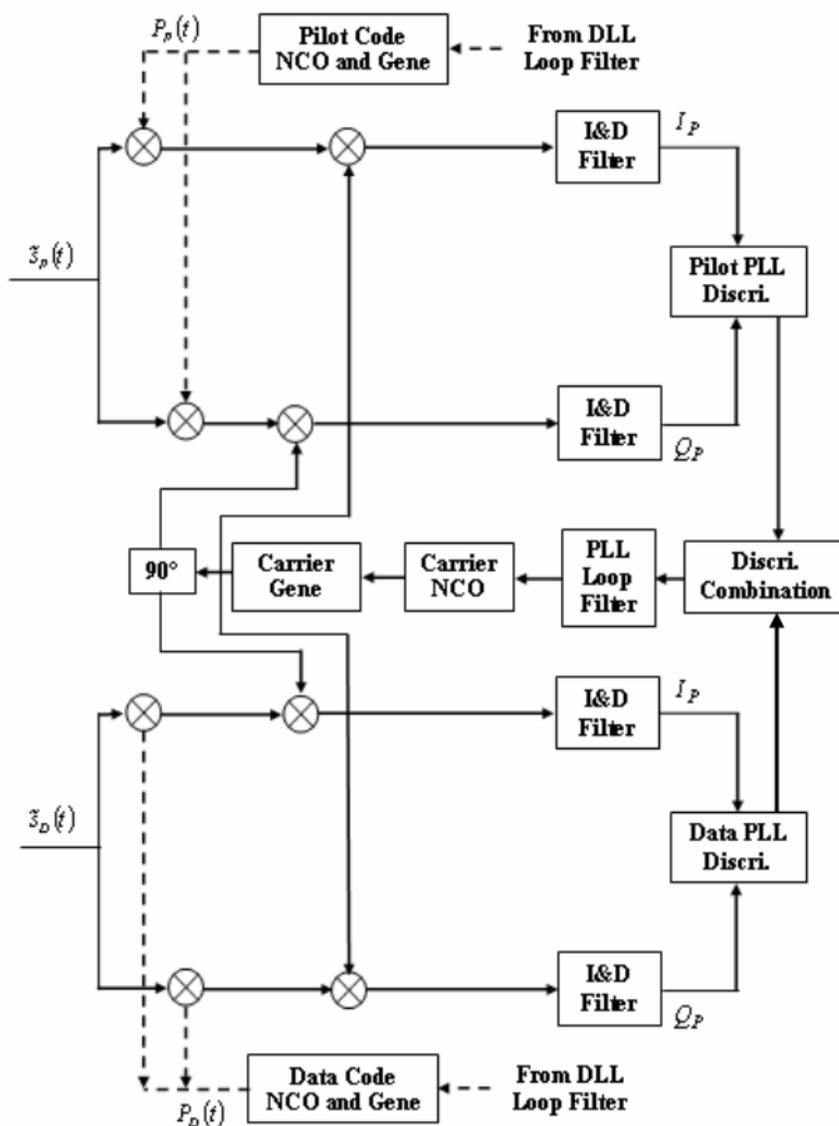
In order to produce an optimal estimation process,  $\gamma$  and  $\kappa$  should be chosen as:

$$\gamma = \frac{\sigma_{D_{Pilot}}^2}{\sigma_{D_{Data}}^2 + \sigma_{D_{Pilot}}^2}; \kappa = \frac{\sigma_{D_{Data}}^2}{\sigma_{D_{Data}}^2 + \sigma_{D_{Pilot}}^2} \quad (4.21)$$

The values of  $\gamma$  and  $\kappa$  can be found using Equations (4.11) and (4.12). The resulting variance of the phase tracking error then equals:

$$\sigma_{D_{Comb1}}^2 = \frac{\sigma_{D_{Pilot}}^2 \sigma_{D_{Data}}^2}{\sigma_{D_{Pilot}}^2 + \sigma_{D_{Data}}^2} \quad (4.22)$$

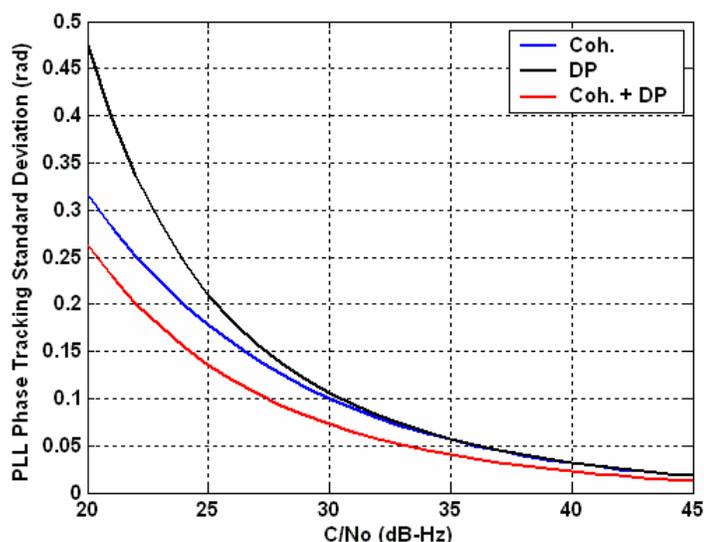
The actual architecture of this combined scheme is shown in Figure 4.13. The resulting tracking phase error is represented in Figure 4.14 for a coherent integration time of 4 ms, in order to mimic Galileo L1F capabilities. The only limitation in this combination is that both channels must provide phase measurements at the same time, constraining the integration time to a level of 4 ms in the case of Galileo L1F.



**Figure 4.13 – Schematic Data/Pilot PLL Architecture**

It can be assumed that if the user wants to improve the carrier phase measurement accuracy as compared to the use of the pilot channel only, it is likely that it is because the best measurement possible is wanted. For this reason, it is then assumed that the discriminator used on the pilot channel is the coherent discriminator, since it provides the most accurate measurements. Consequently, since the PLL tracking error using the coherent discriminator does not depend upon the coherent integration time, the limitation

in terms of integration time is not problematic. It is interesting that such an implementation would lead to optimal use of the data and pilot discriminators' outputs, and an improvement in terms of tracking accuracy as compared to the use of the pilot channel only.



**Figure 4.14 – PLL Tracking Error when using the Coherent Discriminator Only, the DP Discriminator Only, and Combined Coherent and DP Discriminators for an Integration Time of 4 ms and a 10 Hz PLL Loop Filter Bandwidth**

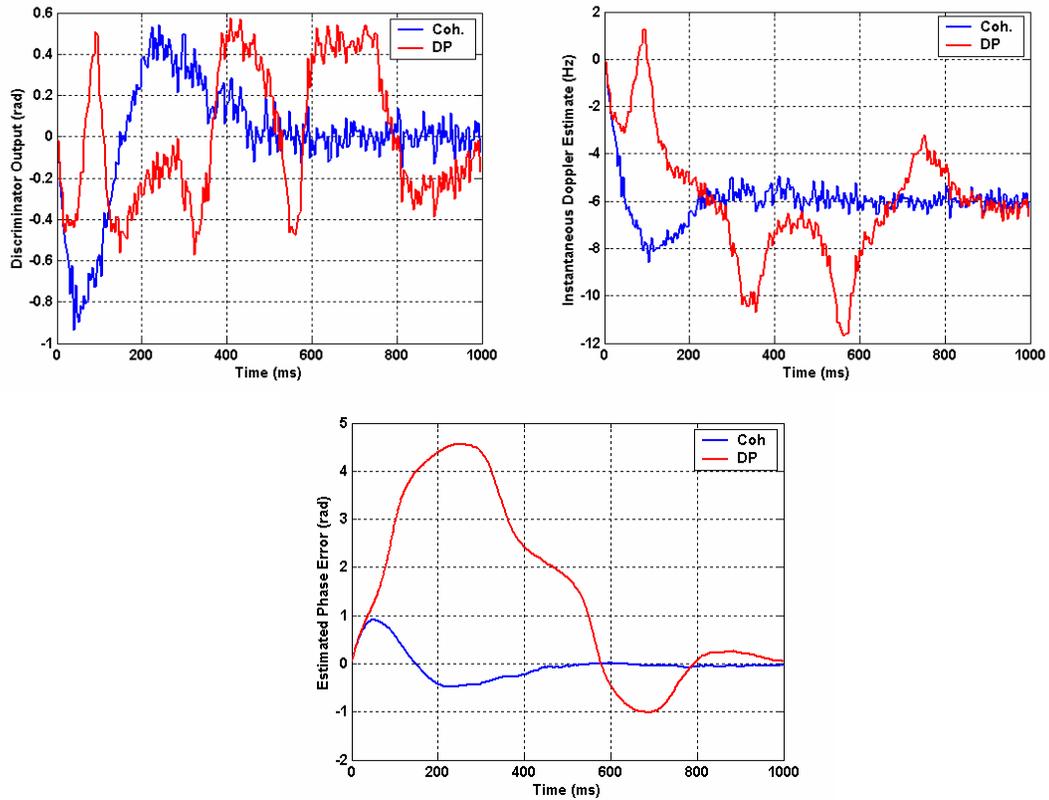
However, the method represented by Equation (4.20) cannot be implemented in its current form for two main reasons. The first reason is the thermal noise. As noted above, the sensitivity of the discriminator on the data channel is lower than on the pilot channel. This means that it is not possible to always use the combination of the two discriminator outputs. To avoid the use of irrelevant information coming from the data discriminator due to low signal power, setting a  $C/N_0$  threshold (dependent upon expected dynamics, etc.) under which the information coming from the data channel would not be considered appears to be indicated. A practical value for this threshold can be taken, for example,

from Table 4.5 or Table 4.6. This means that in this configuration, the data channel will not assist the achievement of greater phase tracking sensitivity for low  $C/N_0$  values.

The second reason is related to the signal's dynamics. First, the tracking parameters required to reach optimal tracking can differ between the pilot and data channels, as seen through the values of the loop bandwidth in Table 4.5 and Table 4.6. Second, large dynamics can make the phase error grow outside the stability region of the discriminator used. This can constitute a major problem in the implementation of a combination of the two discriminators' outputs since it is possible that a jump in the phase error could be sensed by the pilot discriminator, but not by the data discriminator due to their different linearity region widths.

Based on these considerations, a choice needs to be made on the data channel discriminator. It has been shown in Table 4.5 that the DP and classical arctangent discriminators have equivalent sensitivity for 4 ms integration durations and medium dynamics. However, for low dynamics, it appears that the choice of the DP discriminator brings higher tracking sensitivity. Since the goal is to obtain accurate measurements, it is important to have a means of obtaining accurate  $C/N_0$  measurements as low as possible. Those users who require high accuracy phase measurements are likely specialized users. Consequently, it can be assumed that the receivers providing this will have a high quality oscillator such as the TCXO modeled and used in Table 4.5. The limitation of the integration time to 4 ms also allows for limiting potential problems due to high dynamics. Consequently, it seems appropriate to apply the DP discriminator to the data channel.

The problem of high dynamics, although likely not frequent, must still be addressed since the occurrence of a cycle slip is undesirable and potentially disastrous for high accuracy phase measurements. Figure 4.15 underlines the fact that such a problem poses when using the DP discriminator which has a limited linear tracking region. It shows the DP and coherent discriminators' responses, as well as the resulting PLL tracking performance for a  $C/N_0$  of 45 dB-Hz using a coherent integration time of 4 ms, and a loop bandwidth of 10 Hz. An initial Doppler error of 6 Hz was assumed. Since the loop bandwidth inserts a lag into the loop response to the simulated dynamics (the jump in Doppler), the phase error initially increases and, since the two discriminators have the same slope at the origin, they first respond in the same way to that stress. However, once the phase error reaches approximately 0.4 radians (the limit of the DP discriminator linearity domain), the DP discriminator cannot correctly estimate the phase error. It then underestimates the phase error and undergoes a cycle slip. On the other hand, the coherent discriminator can correctly estimate the phase error for a longer period of time, which allows for the loop to catch up with the Doppler error, and provides unbiased phase estimation. In this example, the data channel luckily goes back to the correct phase error. However, it demonstrates that, if only a  $C/N_0$  threshold is used, the implementation might still lead to a biased measurement, or to a loss of lock due to high signal dynamics.



**Figure 4.15 – DP and Coherent Discriminator Output, Estimated Doppler and Estimated Phase Error using 4 ms Integration Time, a Loop Bandwidth of 10 Hz, a  $C/N_0$  of 45 dB-Hz and an Initial Doppler Error of 6 Hz**

In order to take this phenomenon into account, a first method is proposed. It intends to constantly check the consistency between the outputs of the data and pilot channels' discriminators. The algorithm is given by:

$$\begin{aligned}
 D_{Comb2} &= \gamma D_{Data} + \kappa D_{Pilot} && \text{if } |D_{Data} - D_{Pilot}| \leq \eta \\
 D_{Comb2} &= D_{Pilot} && \text{else}
 \end{aligned} \tag{4.23}$$

Since  $D_{Pilot}$  and  $D_{Data}$  are intended to estimate the same quantity, it seems natural to test their consistency in this way. Moreover, since it appears more practical to have noisier but unbiased measurements, rather than biased but clean measurements, the value

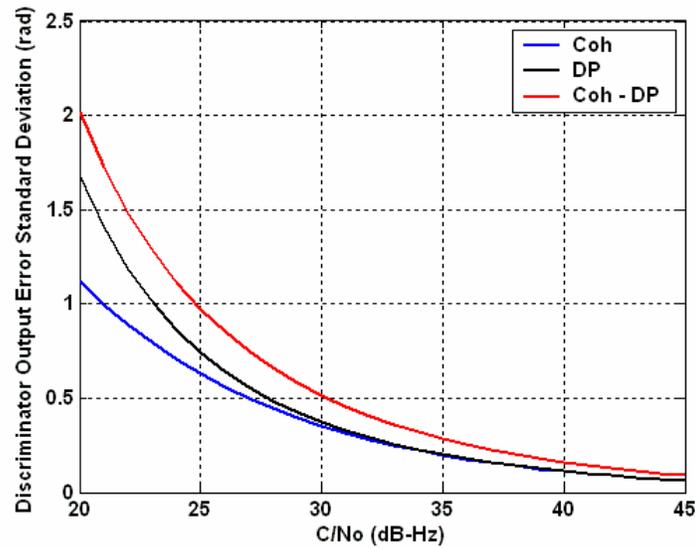
of  $\eta$  should be chosen to be tight ( $\pi/4$  in Julien *et al.* (2004d), which is the value used throughout this section). However, one drawback associated with this method is that since it directly uses the output of the discriminator of each channel, the test function can be quite noisy. The standard deviations of the coherent and DP discriminators' outputs are given by:

$$\sigma_{D_{coh}}^2 = \frac{1}{2T_I \frac{C}{N_0}} \quad (4.24)$$

and

$$\sigma_{D_{DP}}^2 = \frac{1}{2T_I \frac{C}{N_0}} \left( 1 + \frac{1}{2T_I \frac{C}{N_0}} \right) \quad (4.25)$$

These quantities are shown in Figure 4.16 for an integration time of 4 ms along with the standard deviation associated with the test introduced in Equation (4.23). Since the test function uses a combination of the two discriminators' outputs, it seems that for low  $C/N_0$ , it might not be relevant anymore, and it is likely that no combination will be used - not because of cycle slips or dynamics, but because of the high noise of the test function. In such a case, the differentiation between the impact of noise and dynamics might not be clear. It also means that, for low  $C/N_0$ , this implementation will not have a significant impact on the effectiveness of noise mitigation techniques; however, even if the test is noisy, the algorithm should always follow the pilot channel, even for low  $C/N_0$ , which is important from a robustness perspective.



**Figure 4.16 – Coherent and DP Discriminators’ Output Standard Deviation for an Integration Time of 4 ms**

In order to enhance the performance of this method, and since the noise appears to be the main problem, one might be tempted to use a filtered version of the discriminator output. To all appearances, using a filtered measurement might seem problematic since it induces a time lag associated with the filter. This lag may delay the loop’s response and thus irrelevant information might be delivered by the data channel for too long. The solution is then to use a filter with a low lag.

Although more advanced filters could be used, the study of a basic filter - which has been chosen to be the averaging filter herein - gives a good sense of the benefits of using filtered measurements. The main advantage comes from superior mitigation of the Gaussian noise. The use of the same test function as in Equation (4.23) (but with filtered values) seems a natural strategy. Another possible solution is to exploit the fact that if the discriminator output of the pilot channel is outside of the stability domain of the data

channel discriminator, it is because the dynamics are too high for the data channel and, therefore, use of the latter information is not worthwhile. This is valid only if discriminators with different linear tracking domains are used on the data and pilot channels, as it is the case for the coherent and DP discriminators. The advantage of this test is that it uses only the pilot channel information, so it does not suffer from extra noise brought by a test function using a combination of discriminator outputs. This test can be expressed as:

$$D_{Comb3} = \gamma D_{Data} + \kappa D_{pilot} \quad \text{if } \left| \tilde{D}_{pilot} \right| \leq \frac{L_{\varphi, D_{data}}}{2}$$

$$D_{Comb3} = D_{Pilot} \quad \text{else} \quad (4.26)$$

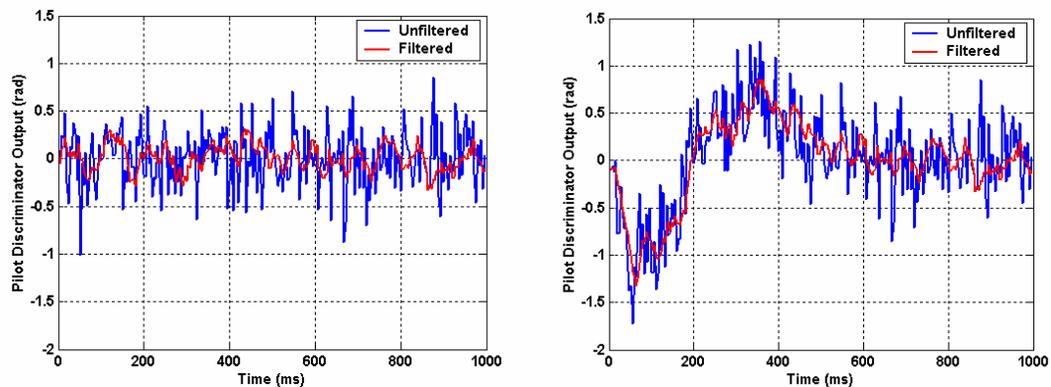
where  $\tilde{D}_{Pilot}$  is the filtered pilot discriminator output, and

$L_{\varphi, D_{data}}$  represents the stability domain of the data discriminator.

One way to calculate the minimum number of samples needed for averaging is to endeavour to have no measurements rejected due to the thermal noise only, when no dynamics are present. This is realistic and it is possible to calculate that minimum value theoretically, assuming that the discriminator output is Gaussian and by using Equations (4.24) and (4.25). The number of samples used for this averaging process can then be chosen so that three times the standard deviation of the test result is smaller than the data channel discriminator one-sided stability domain. To obtain accurate phase measurements, it thus seems natural to use the coherent discriminator on the pilot channel which is less susceptible to Gaussian noise. In such a configuration, and assuming a tracking threshold of 30 dB-Hz for the data channel and an integration time of 4 ms, the

number of samples needed for the filtering according to the previous criterion should be 6 (the linear tracking region of the DP discriminator was assumed to be equal to 0.5 radians, according to Figure 4.4). This means that the averaging will take 24 ms. For some users expecting high dynamics, this time might imply too long a lag and, in such a case, the number of samples might be reduced (to limit the lag while ensuring a reasonable degree of tracking robustness).

Figure 4.17 shows the output of the coherent discriminator for a  $C/N_0$  of 30 dB-Hz and an integration time of 4 ms (with the use of a coherent discriminator), as well as the filtered discriminator output using six epochs. The left figure assumes a correct initial Doppler while the right figure assumes an initial Doppler offset of 6 Hz.

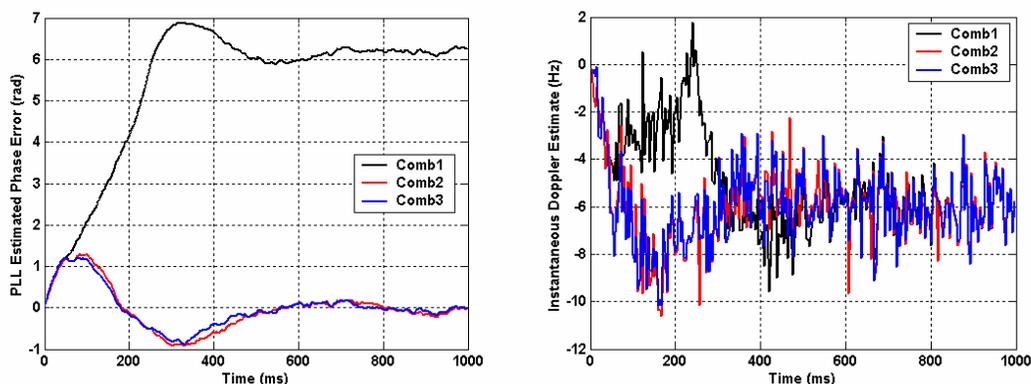


**Figure 4.17 – Pilot Instantaneous and Filtered (6 epochs) Discriminator Output using 4 ms Integration Time, a  $C/N_0$  of 30 dB-Hz, with an Initial Doppler Offset of 0 Hz (Left), and 6 Hz (Right)**

Although filtering introduces a lag, its effect is obviously minor. This lag can be chosen according to the expected intensity of the dynamics. It also shows that, with no Doppler offset and assuming only Gaussian noise, the data channel discriminator linearity domain

boundaries is never crossed when using the test represented in Equation (4.24), which is consistent with the theory.

Figure 4.18 shows the response of the PLL to an initial 6 Hz frequency offset, using the three data/pilot implementation scenarios presented. Combinations 2 and 3 allow avoidance of cycle slips due to their reliance on the pilot channel; on the other hand, the simple combination (combination 1) undergoes a cycle slip since it constantly uses the data channel information. The third data/pilot combined technique seems to exhibit a better response to the Doppler offset since it tends to return to the correct phase error more quickly.



**Figure 4.18 – PLL Tracking Error using the Three Data/Pilot Implementation with a 4 ms Integration Time, a  $C/N_0$  of 30 dB-Hz, and an Initial Doppler Offset of 6 Hz**

One could argue that having a test function based on filtered discriminator output might not be completely relevant since it is made on a quantity (the filtered output) that is not the one actually used by the PLL (combination of instantaneous discriminators' outputs). Although this is true, it has to be emphasized that any problem that might result would lead to an increase in the mean of the pilot discriminator output values and, hence, would

inevitably result in the crossing of the threshold. Thus, the loop reaction would still be relevant.

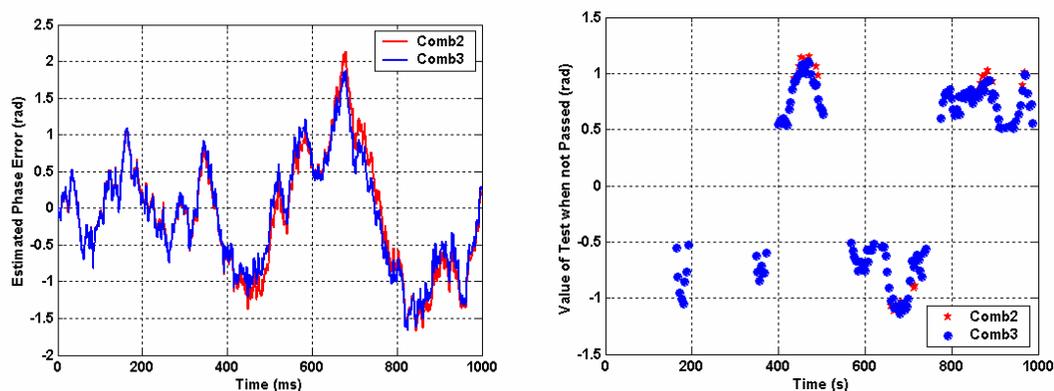
Table 4.7 shows the phase tracking error standard deviation obtained through simulations using the same three implementations as previously used, but with no Doppler offset at the beginning. Since only Gaussian noise is simulated, the first method has the best performance since it constantly uses both channels. However, the third data/pilot proposed method displays the same accurate performance since the test described by Equation (4.26) is always successful due to its filtering of the pilot discriminator output. Finally, the second data/pilot implementation, because of the use of unfiltered discriminator outputs, tends to reject some of the data discriminator estimate and by doing so, it constitutes a less optimal solution for low  $C/N_0$  values. Finally, it should be noted that, as expected, the three combinations outperform the use of the pilot channel only.

**Table 4.7 – Phase Tracking Standard Deviation for the Three Combinations Presented for a  $C/N_0$  of 30 dB-Hz and a Coherent Integration of 4 ms.**

	<b>DP</b>	<b>Coh.</b>	<b>Comb1</b>	<b>Comb2</b>	<b>Comb3</b>
<b>STD (rad)</b>	<b>0.130</b>	<b>0.120</b>	<b>0.090</b>	<b>0.095</b>	<b>0.090</b>

A final test was conducted to compare combinations 2 and 3. A Quartz oscillator was modeled in order to simulate short and sudden changes in dynamics. The  $C/N_0$  for each channel was set to 30 dB-Hz. The results are shown in Figure 4.19. It can be shown that both combinations behave remarkably similarly; however, when looking at the phase error standard deviation, the third data/pilot implementation seems to have an edge since its associated standard deviation equals 0.72 radians, while combination 2 carries a

standard deviation of 0.76 radians. This is visible mainly at the end of the simulation, and is due to the more pragmatic test used by the third methods shown on the right-hand side of Figure 4.19 that illustrates the epochs when only the pilot channel was used and, in such cases, the respective values of the second and third data/pilot implementations' tests.



**Figure 4.19 – Estimated Phase Error using Data/Pilot Combinations 2 and 3 with a  $C/N_0$  of 30 dB-Hz, and a Quartz Oscillator**

It must be emphasized that the methods presented herein are not specific to the Galileo L1F signal, but to all signals possessing a pilot channel in quadrature with a data channel. However, the short data bit duration makes the correlation values and consequently, the discriminator output, noisier. This explains the necessity of employing a filtering technique. On the other hand, if the data bit duration was longer, it would be more appropriate to use the classical arctangent discriminator on the data channel, since it possesses a wider linearity domain and would have a higher sensitivity than the DP discriminator. In such a configuration, one would have to be careful to take into account the reduction of the arctangent mean output for low  $C/N_0$ . As a consequence the second

algorithm might be a more relevant selection since it does not deal with the different linearity region widths.

## 4.7 Synthesis

Phase tracking is always the weak part of current GPS stand-alone receiver since the PLL cannot use optimal discriminators and consequently, tends to lose lock very early due to the presence of the data bit. This chapter emphasized the gain realized by the pilot channel in the Galileo L1F. The choice of the discriminator has been shown to be extremely important to obtaining the best performance in terms of sensitivity, robustness or accuracy. It has been seen that the coherent discriminator provides the highest accuracy without requiring long coherent integrations that could be perilous in changing conditions. On the other hand, if long integrations are allowed, the extended arctangent approach offers the advantage of possessing a potentially very wide linear tracking region. Consequently, it is extremely resistant to high dynamics. However, its linearity domain is dependent upon the SNR and, if misused, could lead to erroneous phase tracking. Moreover (and this is true as well for the classical arctangent) if the input phase error is at the boundaries of its stability domain, it could quickly undergo a cycle slip since its mean discriminator output edge is sharp for high SNR values. It has been shown that any of those two pilot discriminators could lead to an improvement of the PLL sensitivity by more than 5 to 6 dB as compared to the use of a data channel, which is significant in terms of the overall receiver sensitivity.

Finally it has been shown that, even with fast data rate on the data channel, it was possible to use a combined data/pilot implementation in order to increase phase measurement accuracy without sacrificing sensitivity or robustness. Three methods were presented, but one method introduced was shown to be advantageous for noise mitigation purposes. The improvement has been shown to be around 25% as compared to the use of the Galileo L1F pilot channel only in terms of measurement accuracy standard deviation improvement in presence of Gaussian noise for a  $C/N_0$  of 30 dB-Hz. However, the data channel is limited by its own sensitivity, and thus cannot be used under a certain SNR threshold in a combined data/pilot implementation.

## **CHAPTER 5**

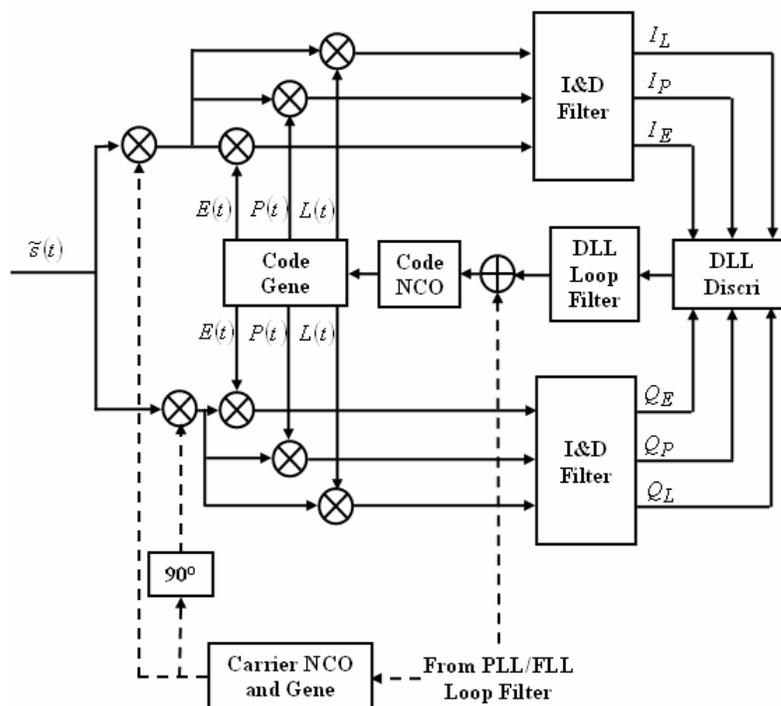
### **GALILEO L1F CODE DELAY TRACKING**

Code delay tracking is extremely important in a GNSS receiver since it provides the user with more robust measurements than the PLL due to the long code chip duration in comparison to the GNSS carrier wavelength. For this reason, many applications that require robust measurements (either owing to the sensitivity of the application or a desire to have measurements in multiple – harsh and friendly - environments) would then rely on code delay measurements. Consequently, in the design of a DLL, it is very important to endeavour to provide accurate but robust and reliable measurements. This chapter first introduces the basis of a typical GNSS code delay tracking loop. The main DLL components are described, with special emphasis on the potential code discriminators for Galileo L1F. Indeed, the discriminator, as in the case of the PLL, is responsible for a great part of the overall DLL behaviour. The various tracking error sources are then described, and their impact on Galileo L1F measurement accuracy assessed and compared to the GPS C/A code tracking performance. The DLL tracking sensitivity is then investigated to evaluate the nature of the degradation a Galileo L1F could encounter without totally impairing code tracking performance. Finally, a new implementation specific to the Galileo L1F signal is introduced to mitigate multipath-induced errors.

## 5.1 Generic DLL Structure

### 5.1.1 Structure Overview

The general architecture of a DLL is shown in Figure 5.1. The following description will use the notations shown in the figure.



**Figure 5.1 – Typical Architecture of a GNSS Delay Lock Loop**

After the carrier wipe-off stage (studied in Chapter 4), the resulting in-phase and quadrature-phase components are multiplied by three delayed spreading sequence replicas (Early, Prompt and Late) that are given by:

$$E(t) = g\left(t - \hat{\tau} + \frac{\delta}{2}\right) \quad (5.1)$$

$$P(t) = g(t - \hat{\tau}) \quad (5.2)$$

$$L(t) = g\left(t - \hat{\tau} - \frac{\delta}{2}\right) \quad (5.3)$$

with the spreading sequence  $g$  defined as:

$$g(t) = c_{C/A}(t) \quad \text{for GPS C/A,}$$

$$g(t) = c_{L1-B}(t)SC(t) \quad \text{for Galileo L1-B, and}$$

$$g(t) = c_{L1-C}(t)SC(t) \quad \text{for Galileo L1-C}$$

and where  $\delta$  is referred to as the correlator spacing.

The result of the multiplication of the in-phase and quadra-phase components with different delayed spreading sequences is followed by filtering with the I&D filters that provide the following six correlation values, assuming a limited phase and frequency variation during the I&D coherent integration time, (see Equations (2.12) and (2.13)):

$$I_X = \sqrt{\frac{P}{2}} \tilde{R}(\varepsilon_\tau + \delta X) D \frac{\sin(\pi \varepsilon_f T_I)}{\pi \varepsilon_f T_I} \cos(\varepsilon_\phi) + n_{I_X} \quad (5.4)$$

$$Q_X = \sqrt{\frac{P}{2}} \tilde{R}(\varepsilon_\tau + \delta X) D \frac{\sin(\pi \varepsilon_f T_I)}{\pi \varepsilon_f T_I} \sin(\varepsilon_\phi) + n_{Q_X} \quad (5.5)$$

where

$X$  represents the early, prompt or late components, and

$\delta X$  represents the early ( $\delta X = \frac{\delta}{2}$ ), prompt ( $\delta X = 0$ ) or late ( $\delta X = -\frac{\delta}{2}$ ) delay.

These correlation values are then fed into a code delay discriminator that will use them to estimate the code delay error  $\varepsilon_\tau$ . This value is then filtered by the DLL loop filter that has a structure very similar to the PLL loop filter described in Section 4.1.

In the exact same way as for the PLL, the output of the loop filter is then fed to the code NCO, which provides the updated local code rate (or code Doppler) in order to compensate for the estimated code delay error. Finally, this code rate is used to generate the code replica for the next correlation period. The equivalent loop filter is referred to as  $h_{DLL}$ .

It is important to note that the local code rate and the local carrier frequency Doppler are linked since they are both due to the ‘satellite – receiver’ signal propagation time variation. The only difference is the opposite effect of the ionospheric delay on the code and phase group delay, and the impact of multipath on code and carrier tracking, as described by Ray (2002). However, the resulting difference in the code and carrier Doppler is often minimal (except in the case of extreme ionospheric activity) and, consequently, it is very common to use carrier tracking information to aid the code tracking loop. As frequency estimation from the PLL is usually very accurate, as seen in Section 4.4, it means that it absorbs the main dynamic component and, consequently, the DLL is not required to perform this task independently. Thus, the DLL loop filter is generally a first order filter with a narrow bandwidth, as it should then be used mainly to remove the noise. The typical one-sided DLL loop filter, with carrier-aiding, is around 0.05 and 1 Hz according to the application (Van Dierendonck 1997). Unless otherwise stated, carrier aiding is assumed to be always available.

### 5.1.2 Discriminators

The choice of the discriminator is extremely important since it conditions the first estimate of the code delay tracking error. Two types of discriminator are widely used in GPS receivers: the *Early-Minus-Late Power* (EMLP) and the *Dot-Product* (DP) discriminators. These two discriminators are considered to be non-coherent as they are insensitive to carrier phase errors, which is extremely interesting for a tracking robustness point of view. They are given by:

$$D_{EMLP} = (I_E^2 + Q_E^2) - (I_L^2 + Q_L^2) \quad (5.6)$$

and

$$D_{DP} = (I_E - I_L)I_P + (Q_E - Q_L)Q_P \quad (5.7)$$

Both can be reproduced for the Galileo L1F signal tracking, and will be the central focus of this section. When building a receiver, the number of correlators has an important impact on the amount of power consumed. This is of major importance, especially for autonomous systems. With this in mind, the DP discriminator is usually preferred to the EMLP discriminator since the ‘Earl-Minus-Late’ part of this discriminator can be obtained through the generation of an ‘Early-Minus-Late’ local replica instead of separate ‘Early’ and a ‘Late’ replicas. Moreover, the use of the prompt correlator for the DP discriminator is not a problem, since the prompt correlation values are, in any case, needed by the PLL. Consequently, only two complex correlators per channel are needed for the DLL/PLL architecture using a DP discriminator, while three are needed for the EMLP discriminator.

Assuming no external disturbances and no frequency uncertainty, their expressions can be re-written using Equations (5.4) and (5.5) as:

$$D_{EMLP} = \frac{P}{2} \left( \tilde{R}^2 \left( \varepsilon_\tau - \frac{\delta}{2} \right) - \tilde{R}^2 \left( \varepsilon_\tau + \frac{\delta}{2} \right) \right) \quad (5.8)$$

$$D_{DP} = \frac{P}{2} \left( \tilde{R} \left( \varepsilon_\tau - \frac{\delta}{2} \right) - \tilde{R} \left( \varepsilon_\tau + \frac{\delta}{2} \right) \right) \tilde{R}(\varepsilon) \quad (5.9)$$

It is important to understand that code tracking is based on the measure of the difference between the Early and Late correlation values that are meant to be situated on each side of the main autocorrelation peak of the spreading sequence. Consequently, to derive the discriminator estimate value under such conditions, two assumptions must be made:

- The estimated error is smaller than half of the correlator spacing, and
- The Early and Late correlator values correspond to points on the spreading symbol autocorrelation function that are on the slope forming its main peak.

These two assumptions can be represented as follows:

$$|\varepsilon_\tau| \leq \frac{\delta}{2} \quad (5.10)$$

and

$$\left| \varepsilon_\tau + \frac{\delta}{2} \right| \leq l \quad (5.11)$$

where  $l$  represents the one-sided width of the spreading symbol's main peak ( $l = 1$  chip for GPS C/A and  $l = 0.5$  chips for Galileo L1F).

The GPS C/A and Galileo L1F spreading sequence autocorrelation functions' main peak are given in Equations (2.1) and (2.30). Both can be included in the following model:

$$R(x) = 1 - \alpha|x| \quad (5.12)$$

where  $\alpha$  corresponds to the absolute value of the slope of the spreading sequence autocorrelation function main peak;  $\alpha = 1$  for a BPSK(1) signal such as the GPS C/A signal, and  $\alpha = 3$  for an sBOC(1,1) (or a sBOC(n,n) in general) signal, like Galileo L1F.

Assuming an infinite front-end filter ( $\tilde{R}(x) = R(x)$ ), Equations (5.8) and (5.9) can then be simplified in the following form:

$$D_{EMLP} = P\alpha(2 - \alpha\delta)\varepsilon_\tau \quad (5.13)$$

$$D_{DP} = P\alpha(1 - \alpha|\varepsilon_\tau|)\varepsilon_\tau \quad (5.14)$$

From these two equations, it can be seen that the EMLP discriminator's output is linear, while the DP discriminator has a quadratic term in  $\varepsilon_\tau$  that will bias its estimate of the error for large values of  $\varepsilon_\tau$ . This bias grows even faster for Galileo L1F than for GPS C/A since its inherent value of  $\alpha$  is greater by a factor of three.

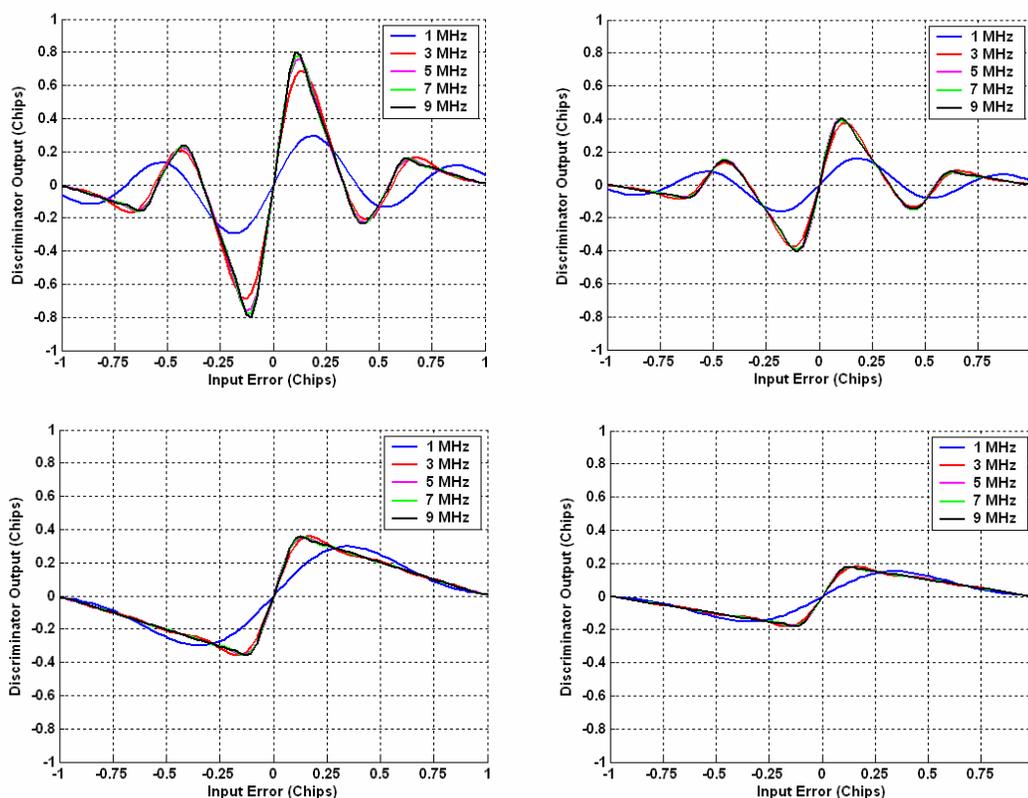
Figure 5.2 and Figure 5.3 show the impact of the front-end filter bandwidth on the EMLP and DP discriminators' output for the sBOC(1,1) and BPSK(1) signals using a correlator spacing of 0.2 and 0.4 chips respectively. It can be seen that if the front-end filter is set too narrow, this will impact the code discriminator slope at the origin, which is also known as the discriminator gain. This comes from the fact that the Early and Late

correlation values are situated on the part of the peak that is rounded off by the front-end filter, as seen in Figure 2.5 and Figure 2.8. Consequently, to follow the theory developed previously and to avoid unnecessary degradation of the discriminator performance, it is important to select a correlator spacing value that shifts the Early and Late correlator on the linear part of the slopes of the spreading sequence main autocorrelation peak.

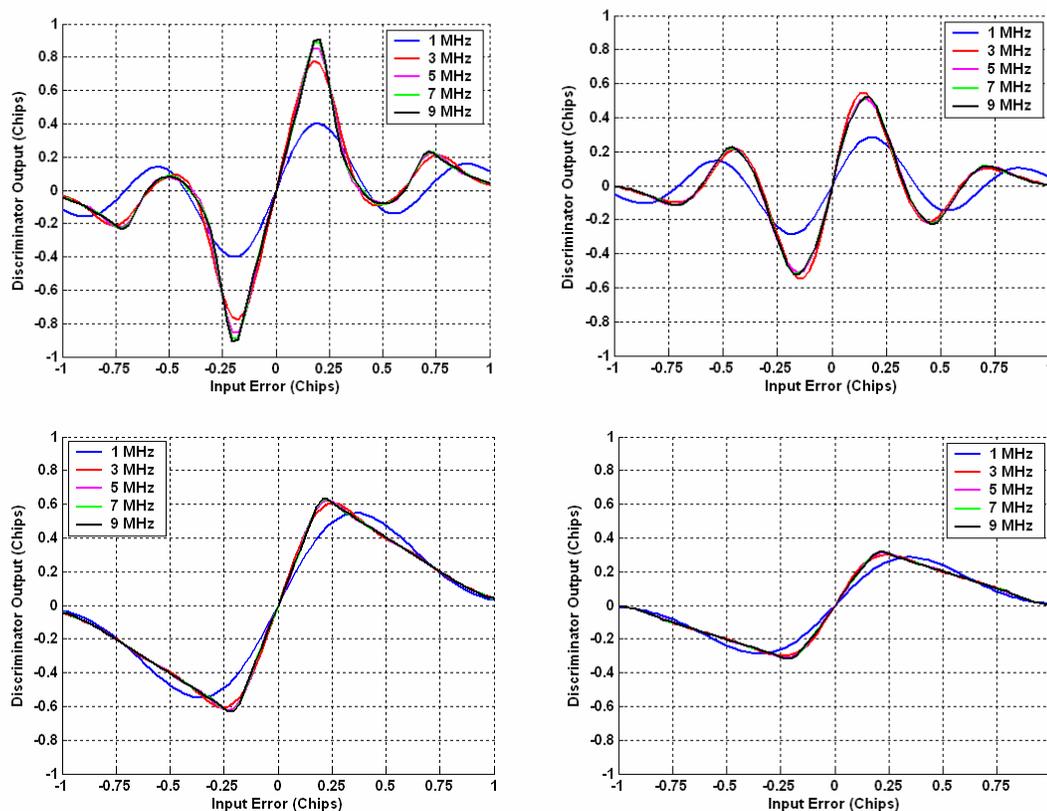
The gain of the discriminator, represented by the slope of the discriminator at the origin, is also dependent upon the discriminator type and the signal modulation, as it can be extrapolated from Equations (5.13) and (5.14). The discriminator gain associated with the sBOC(1,1) modulation is always greater than that associated with BPSK(1) modulation. It is important to note that, for a sufficiently large front-end filter (greater than 2 to 3 MHz one-sided), the discriminator slope at the origin will match the infinite bandwidth case. This implies that assuming a workable choice of the pair of values “correlator spacing – front-end filter bandwidth”, the EMLP and DP discriminators can be normalized by the first derivative of Equations (5.13) and (5.14) to provide unbiased code delay estimation. However, as will be seen in Section 5.2.1.2, it is still necessary to check for potential small mismatches before using these values for normalization in order to avoid degraded tracking performance.

It is also interesting to note that the shape of the discriminator output for the sBOC(1,1) and the BPSK(1) signals are very different. First of all, the sBOC(1,1) discriminator exhibits two false lock points in the range  $\pm 1$  chips, located approximately around  $\pm 0.6$  chips. This means that Galileo L1F signal tracking could lead to biased measurements, if

this issue is not properly addressed. This problem, which is intrinsic to the BOC modulation in general, is the focus of Chapter 6. Secondly, the stability region, already defined for the PLL in Section 4.2, is significantly larger for BPSK(1) modulation ( $\pm 1$  chip) than for the sBOC(1,1) case ( $\pm 0.33$  chips). This means that the discriminators associated with the Galileo L1F signal will be more susceptible to large code delay errors since, in this case, they will respond in an erroneous way to the input stress, and could potentially force the Galileo L1F DLL to lock on a false lock point. This is especially dangerous in case of a short loss of lock that would cause rapid drift in the code delay error.



**Figure 5.2 – EMLP (Left) and DP (Right) Discriminator Output for sBOC(1,1) (Top), and BPSK(1) (Bottom) as a Function of the One-Sided Front-End Filter Bandwidth for a Correlator Spacing of 0.2 Chips**



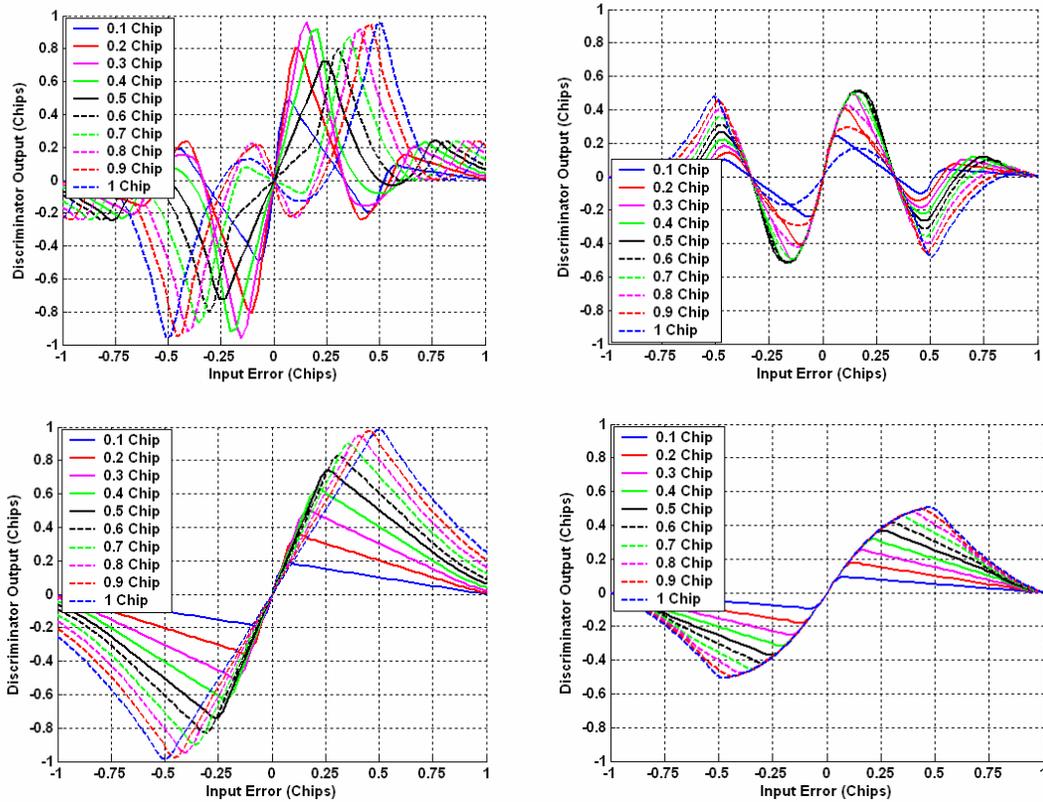
**Figure 5.3 – EMLP (Left) and DP (Right) Discriminator Output for sBOC(1,1) (Top), and BPSK(1) (Bottom) as a Function of the (One-Sided) Front-End Filter Bandwidth for a Correlator Spacing of 0.4 Chips**

However, in terms of tracking robustness, it is more instructive to examine the linear tracking domain than the stability domain, which is also defined in Section 4.2. This domain, assuming correct normalization, corresponds to a correct and unbiased response of the discriminator. A wide linear region means that the discriminator will be able to correctly respond to a large input error, which would make the tracking loop more robust. From Figure 5.2 and Figure 5.3, it can be seen that, given a wide enough front-end filter, Galileo L1F and GPS C/A signals offer the same linear tracking region for the two discriminators studied, a relationship that follows the theory. This linear tracking region

is contained within  $\pm \frac{\delta}{2}$  chips. This is mostly confirmed by Figure 5.4, which shows the EMLP and DP discriminators' outputs for correlator spacing values between 0.1 and 1 chip for a one-sided front-end filter bandwidth of 10 MHz. However, this principle does not always hold in all cases, for two main reasons:

- The linear tracking region is dependent upon the spreading symbol's autocorrelation shape. As assumed through Equations (5.10) and (5.11), to follow the theory, the EMLP and DP discriminators require a correlator spacing value that is smaller than half of the one-sided width of the autocorrelation main peak, corresponding to 1 chip for the BPSK(1) case and to 0.5 chips for the sBOC(1,1) case. As a result, GPS C/A tracking can offer a wider linear tracking region that would result in direct improvement in tracking stability than Galileo L1F tracking. This is confirmed by Figure 5.4, except for the sBOC(1,1) EMLP discriminator. Indeed, the EMLP discriminator uses the square of the sBOC(1,1) autocorrelation values, and consequently, the two negative peaks of the autocorrelation function within  $\pm 1$  chip become positive, shortening the width of the squared autocorrelation main peak to  $\pm 0.33$  chips. As a result, the sBOC(1,1) EMLP discriminator requires a correlator spacing lower than 0.33 chips to function properly.
- Equation (5.14) showed that the DP discriminator had an estimation error that was dependent upon the input error. Consequently, the greater the code delay error, the more the discriminator output will be biased. This shortens the linear tracking

region. This can be seen in Figure 5.4 through the rounding of the discriminator output with increasing input errors.



**Figure 5.4 - Discriminator Output for sBOC(1,1) EMLP (Top Left) and DP (Top Right), and for BPSK(1) EMLP (Bottom Left) and DP (Bottom Right) as a Function of the Correlator Spacing for a 10 MHz One-Sided Front-End Bandwidth**

As expressed in Equations (5.13) and (5.14), the discriminator output does not directly estimate the input error. Consequently, to obtain relevant and unbiased code delay error estimation, the discriminator output has to be normalized by its gain. This procedure can be difficult, and is explained in the next section.

### 5.1.3 Discriminator Normalization

Figure 5.2 showed that, assuming a sufficiently large front-end filter bandwidth, the slope at the origin of the discriminator output was approximately the same as in the case of infinite bandwidth. Thus, it is consistent to use the infinite bandwidth case to effectively achieve normalization. The discriminator output, assuming an infinite front-end filter, has already been given in Equations (5.13) and (5.14). The derivative of these expressions will thus be taken as the normalizing factor. However, this derivative is also proportional to the incoming signal power. This means that an estimation of the signal power is also necessary. This can be done in different ways. Three common examples of normalizations methods follow:

$$N_1 = (I_E + I_L)^2 + (Q_E + Q_L)^2 \quad (5.15)$$

$$N_2 = (I_E + I_L)I_P + (Q_E + Q_L)Q_P \quad (5.16)$$

$$N_3 = I_P^2 + Q_P^2 \quad (5.17)$$

A fourth normalization, representing perfect normalization and thus equal to the discriminator gain, is introduced:

$$N_P = \left. \frac{dD_X}{d\varepsilon_\tau} \right|_{\varepsilon_\tau=0} \quad (5.18)$$

For an infinite front-end filter bandwidth, and with the assumptions given through Equations (5.10) and (5.11), the non-ideal normalizations' expressions reduce to the following:

$$N_1 = \frac{P}{2}(2 - \alpha d)^2 \quad (5.19)$$

$$N_2 = \frac{P}{2}(2 - \alpha d)(1 - \alpha|\varepsilon_\tau|) \quad (5.20)$$

$$N_3 = \frac{P}{2}(1 + \alpha^2 \varepsilon_\tau^2 - 2\alpha|\varepsilon_\tau|) \quad (5.21)$$

$N_1$  is the typical normalization used for the EMLP discriminator.  $N_2$  is the typical normalization for the DP discriminator since it allows full cancellation of the dependence of the DP discriminator expression with the factor  $(1 - \alpha|\varepsilon_\tau|)$  (see Equation (5.14)); thus, when used, it should provide an unbiased estimation of the code delay error within  $\pm \delta/2$  chips. However, since it includes an ‘Early-Plus-Late’ component, it means that the DP normalized with  $N_2$  would require three complex correlators (such as the EMLP), losing its advantage of requiring less complex correlators for lower power consumption.  $N_3$  is often used because it represents an easy way to assess the instantaneous signal power. It also does not require an extra correlator, since it uses only the Prompt correlation values. This means that, owing to its association with the DP discriminator, it would enable efficient operation in terms of power demand. Finally  $N_P$  is meant to show the effect of an ideal normalization on the overall discriminator (and DLL) performance.

Assuming no disturbances and an infinite front-end filter, one can then extrapolate that the normalized EMLP discriminators can be written as:

$$V_{EMLP1} = \frac{(2 - \alpha d) D_{EMLP}}{2\alpha N_1} = \varepsilon_\tau \quad (5.22)$$

$$V_{EMLP3} = \frac{D_{EMLP}}{2\alpha(2 - \alpha d)N_3} = \frac{\varepsilon_\tau}{(1 - \alpha|\varepsilon_\tau|)^2} \xrightarrow{\varepsilon_\tau \rightarrow 0} \varepsilon_\tau \quad (5.23)$$

$$V_{EMLPP} = \frac{D_{EMLP}}{N_P} = \varepsilon_\tau \quad (5.24)$$

Similarly, the normalized DP discriminators are written as:

$$V_{DP2} = \frac{(2 - \alpha d) D_{DP}}{2\alpha N_2} = \varepsilon_\tau \quad (5.25)$$

$$V_{DP3} = \frac{D_{DP}}{2\alpha N_3} = \frac{\varepsilon_\tau}{(1 - \alpha|\varepsilon_\tau|)} \xrightarrow{\varepsilon_\tau \rightarrow 0} \varepsilon_\tau \quad (5.26)$$

$$V_{DPP} = \frac{D_{DP}}{N_P} = \varepsilon_\tau (1 - \alpha|\varepsilon_\tau|) \xrightarrow{\varepsilon_\tau \rightarrow 0} \varepsilon_\tau \quad (5.27)$$

It should be noted that  $V_{EMLP1}$ ,  $V_{EMLPP}$ , and  $V_{DP2}$  constitute the exact unbiased estimate of the code delay input error provided that the assumptions made through Equations (5.10) and (5.11) hold. On the other hand, when using the instantaneous power estimate as the normalization factor, the two resulting discriminators  $V_{EMLP3}$  and  $V_{DP3}$  exhibit a bias that will cause both discriminators to overestimate the error, when it is large. This is even more pronounced for the EMLP discriminator due to the squaring form in the denominator. Finally,  $V_{DPP}$  will constantly underestimate the input error when significant. Note also that, for the sBOC(1,1) modulation, since its associated  $\alpha$

value is three times greater than for the BPSK(1), the associated bias due to normalization by  $N_3$  will grow more rapidly.

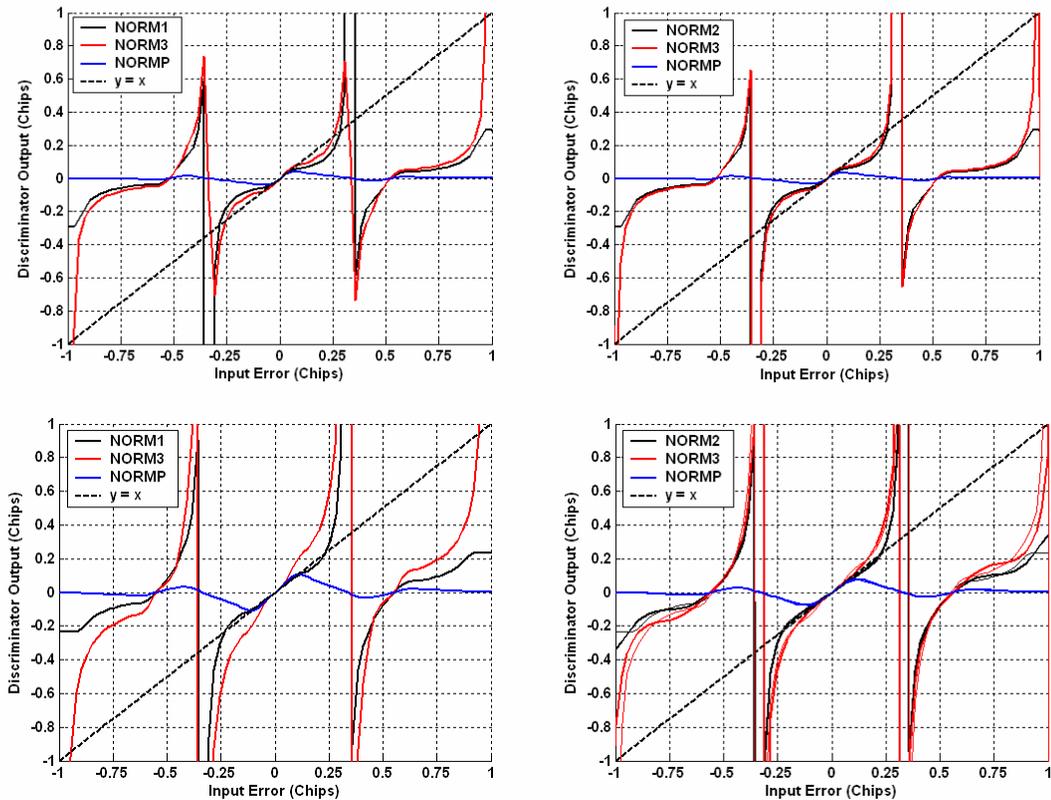
The four normalized discriminators are represented in Figure 5.5 and Figure 5.6 for a 0.1 and 0.2 chip correlator spacing value, a 10 MHz one-sided front-end filter bandwidth, and for an sBOC(1,1) and BPSK(1) signal respectively. As expected,  $V_{EMLP1}$  and  $V_{DP2}$  exhibit identical linear tracking regions since Equations (5.22) and (5.25) have the same value within  $\pm \delta/2$  chips. The use of the ideal normalization also provides an unbiased estimate of the input error within  $\pm \delta/2$  chips (for the DP, the expected deviation is very small). However, for input errors greater than this, the use of the ideal normalization will always result in an underestimation of the error. It is easy to understand that this could be harmful for the DLL since, under high dynamics, it will not be possible to effectively correct a growing error. This would result in a loss of lock. It can then have severe consequences on tracking robustness, especially if no carrier-aiding is used. However, when using accurate carrier-aiding to model the signal's dynamics, this threat is usually limited.

Upon examination of the figure associated with Galileo L1F, it seems as though the discriminator output has a very different shape according to the correlator spacing value:

- For a 0.2 chip correlator value, it appears that  $V_{EMLP1}$  and  $V_{DP2}$  output follow the first diagonal accurately until  $\pm 0.25$  chips, which implies a linear tracking region wider than the theoretical  $\pm \delta/2$  chips ( $\pm 0.1$  chips in this case). This would

result in very robust tracking, even for errors larger than  $\delta/2$  chips. As expected, the linear tracking domain is more limited when using  $N_3$ , since this normalization induces an overestimation of the input error. Nevertheless, it still provides a correct error estimate within approximately  $\pm\delta/2$  chips. The discriminator output shape of  $V_{EMLP3}$  seems to be the worst performer as it diverges from the first diagonal ( $y=x$ ) very quickly as demonstrated by Equation (5.23).

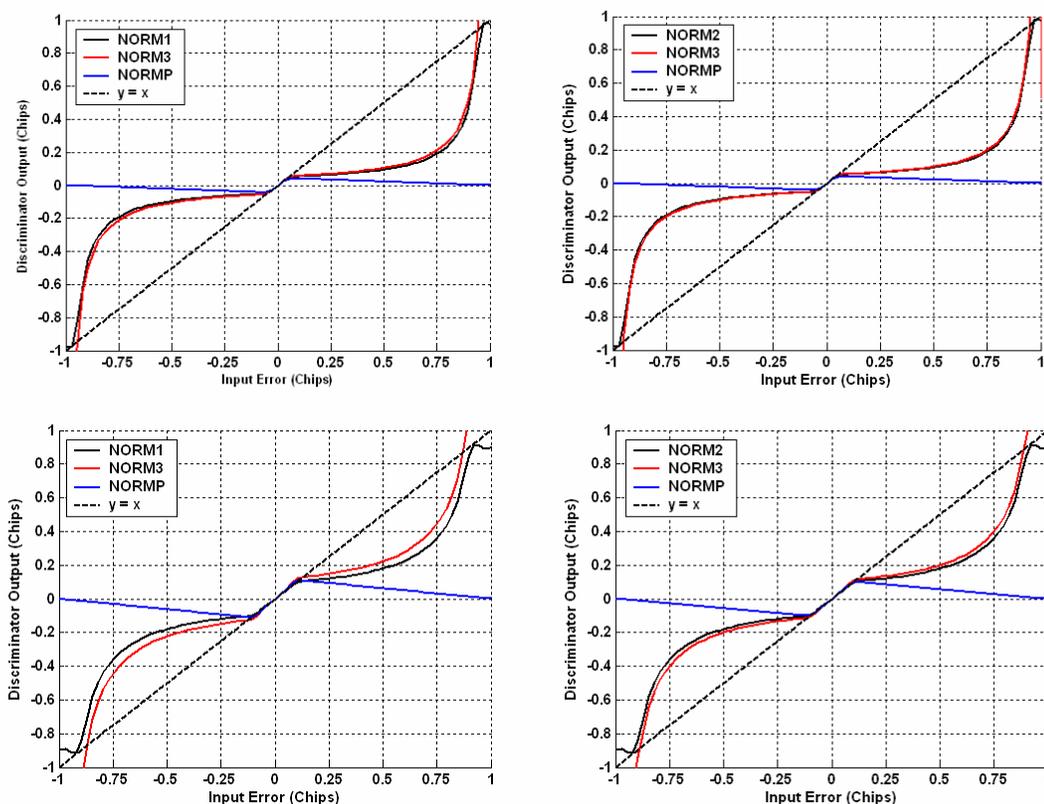
- For a 0.1 chip correlator value, considering all the normalized discriminators, their outputs underestimate the input code delay error if the absolute value of the latter is larger than  $\delta/2$  chips. This is due to the fact that the correlator spacing is significantly smaller than the sBOC(1,1)'s autocorrelation main peak, which is directly used in the non-ideal normalization process. As explained above for the case of the ideal normalization, this discriminator shape might be detrimental for the overall DLL performance.



**Figure 5.5 – sBOC(1,1) EMLP (Left) and DP (Right) Discriminator Output for a 0.1 (Top) and 0.2 (Bottom) Chip Correlator Spacing with all the Normalization Considered for a 10 MHz One-Sided Front-End Filter Bandwidth**

The BPSK(1) modulation offers a very different behaviour. First, it seems like the choice of the normalization (other than the ideal one) does not have a significant effect on the discriminator output shape. Second, outside the linear tracking region, the BPSK(1) normalized discriminators will tend to systematically underestimate the tracking error (for both 0.1 and 0.2 correlator spacing values). As already explained, this discriminator output shape might be very harmful for the tracking process. This is a well known problem faced by narrow correlator spacing tracking techniques used with GPS C/A code. The choice of a larger correlator spacing value can also help to reduce this problem.

Since the BPSK(1) autocorrelation main peak is three times wider than the sBOC(1,1) autocorrelation function, the inherent problem of tracking robustness outside the linearity region will always be more pronounced for the GPS C/A signal than for the Galileo L1F signal.



**Figure 5.6 – BPSK(1) EMLP (Left) and DP (Right) Discriminator Output for a 0.1 (Top) and 0.2 (Bottom) Chip Correlator Spacing with all the Normalization Considered for a 10 MHz One-Sided Front-End Filter Bandwidth**

In all of the cases shown in Figure 5.5 and Figure 5.6, at the discriminator stability region boundaries, the use of a non-ideal normalization creates a vertical asymptote. This is a very interesting phenomenon since it indicates that for large errors the loop will tend to over-react. As long as this reaction is limited, the loop should maintain locked. However, if this is not dealt with, the resultant large discriminator output would produce sudden

tracking jumps that might lead to a loss of lock, or at least to a degraded tracking.

Usually, a discriminator hard limiter is used to control the maximum output of the discriminator to contain this risk. This will be studied in Section 5.2.1.2.

As a preliminary conclusion, the BPSK(1) modulation offers a larger choice of correlator spacing than the Galileo L1F signal, which might provide additional stability to the code tracking loop. However, the shape of the normalized discriminator output indicates that for the same correlator spacing values (i.e. lower than 0.33 chips), Galileo L1F will have a more robust code tracking since it offers superior discriminator response.

To this point, a description of the tracking loop, - and, in particular, the code discriminator - has been realized, assuming no external disturbances in order to introduce the principles and the critical parameters entering into account in the design of the DLL. It is now practical to examine the impact of the main sources of error on the choice of these parameters.

## **5.2 Error Sources**

In a classical DLL, the three main error sources are thermal noise, multipath, and signal dynamics (Ward 1996). Other disturbances, such as the receiver oscillator error, are minimal in the measurement error budget (Ward 1996), and consequently will not be studied herein.

### 5.2.1 Thermal Noise

The origin of thermal noise has been introduced in Section 3.1.6. This study is aimed at achieving a qualitative and quantitative understanding of its impact on code tracking – that is, an understanding of the nature and magnitude of these effects.

#### 5.2.1.1 Theoretical Derivation

Holmes (2000) showed that the DLL estimated code delay tracking error variance in Gaussian noise assuming perfect normalization (equal to the discriminator slope at the origin) and a loop bandwidth negligible compared to the width of the spectrum of the discriminator noise can be written as:

$$\sigma_{X,\varepsilon_\tau,t}^2 = \frac{2B_L(1-0.5B_L T_I)S_{N_X}(0)}{K_X^2} \quad (5.28)$$

with

$$K_X = \left. \frac{dD_X}{d\varepsilon_\tau} \right|_{\varepsilon_\tau=0} \quad (5.29)$$

where

$X$  represents the type of discriminator (EMLP or DP),

$S_{N_X}$  is the discriminator noise PSD, and

$K_X$  is the loop gain associated to the discriminator  $X$ .

The noise PSD resulting from the multiplication of the incoming signal with the local spreading code, is very wide-band. Consequently, with the I&D filter having a very

narrow bandwidth compared to the associated noise and the discriminator being a combination of the different correlator output, one can write:

$$\frac{S_N(0)}{T_I} = \sigma_{D_X}^2 \quad (5.30)$$

where  $\sigma_{D_X}^2$  represents the (not normalized) discriminator output standard deviation.

Consequently, the code tracking error variance can be written as:

$$\sigma_{X, \varepsilon_{\tau}, t}^2 = \frac{2B_L(1-0.5B_L T_I) T_I \sigma_{D_X}^2}{K_X^2} \quad (5.31)$$

This expression is of course similar to the PLL case (Equation 4.10), but with the introduction of the effect of the normalization (through  $K_X$ ). This means that the DLL tracking error is directly dependent upon the discriminator resistance to noise.

The DLL tracking error variance produced by use of an EMLP discriminator, assuming a perfect normalization, no frequency uncertainty in the carrier wipe-off, a front-end filter with unity gain within  $\pm B$  Hz and null elsewhere and a code delay error remaining small, is given by Betz & Kolodziejcki (2000):

$$\sigma_{EMLP, \varepsilon_{\tau}, t}^2 = \frac{B_L(1-0.5B_L T_I) \int_{-B}^{+B} G(f) \sin^2(\pi f \delta) df}{\frac{P}{N_0} \left( 2\pi \int_{-B}^B f G(f) \sin(\pi f \delta) df \right)^2} \left( 1 + \frac{\int_{-B}^{+B} G(f) \cos^2(\pi f \delta) df}{\frac{P}{N_0} T_I \left( \int_{-B}^B G(f) \cos(\pi f \delta) df \right)^2} \right) \quad (5.32)$$

Similarly, as derived in Appendix C, the DLL tracking error variance using a DP discriminator equals:

$$\sigma_{DP,\varepsilon_\tau,t}^2 = \frac{B_L(1-0.5B_L T_I) \int_{-B}^{+B} G(f) \sin^2(\pi f \delta) df}{\frac{P}{N_0} \left( 2\pi \int_{-B}^B f G(f) \sin(\pi f \delta) df \right)^2} \left( 1 + \frac{1}{\frac{P}{N_0} T_I \left( \int_{-B}^B G(f) df \right)} \right) \quad (5.33)$$

It is common, as a first approximation of the loop performance, to assume an infinite front-end filter bandwidth. In such a case, Equations (5.32) and (5.33) can be simplified into:

$$\sigma_{EMLP,\varepsilon_\tau}^2 = \frac{B_L(1-0.5B_L T_I) \delta}{2\alpha \frac{P}{N_0}} \left( 1 + \frac{2}{(2-\alpha\delta) \frac{P}{N_0} T_I} \right) \quad (5.34)$$

$$\sigma_{DP,\varepsilon_\tau}^2 = \frac{B_L(1-0.5B_L T_I) \delta}{2\alpha \frac{P}{N_0}} \left( 1 + \frac{1}{\frac{P}{N_0} T_I} \right) \quad (5.35)$$

Equations (5.34) and (5.35), confirmed by Ries *et al.* (2003), show that the Galileo L1F tracking error standard deviation is expected to be approximately 2.4 dB better than the C/A signal tracking error standard deviation since its corresponding value for  $\alpha$  is three times higher than the one associated with a BPSK(1) modulation. They also indicate the presence of a squaring loss (the term in bracket in Equations (5.34) and (5.35)) due to the multiplication of correlation values. It can be seen that in the infinite front-end filter bandwidth case, the squaring losses are identical between BPSK(1) and sBOC(1,1) for

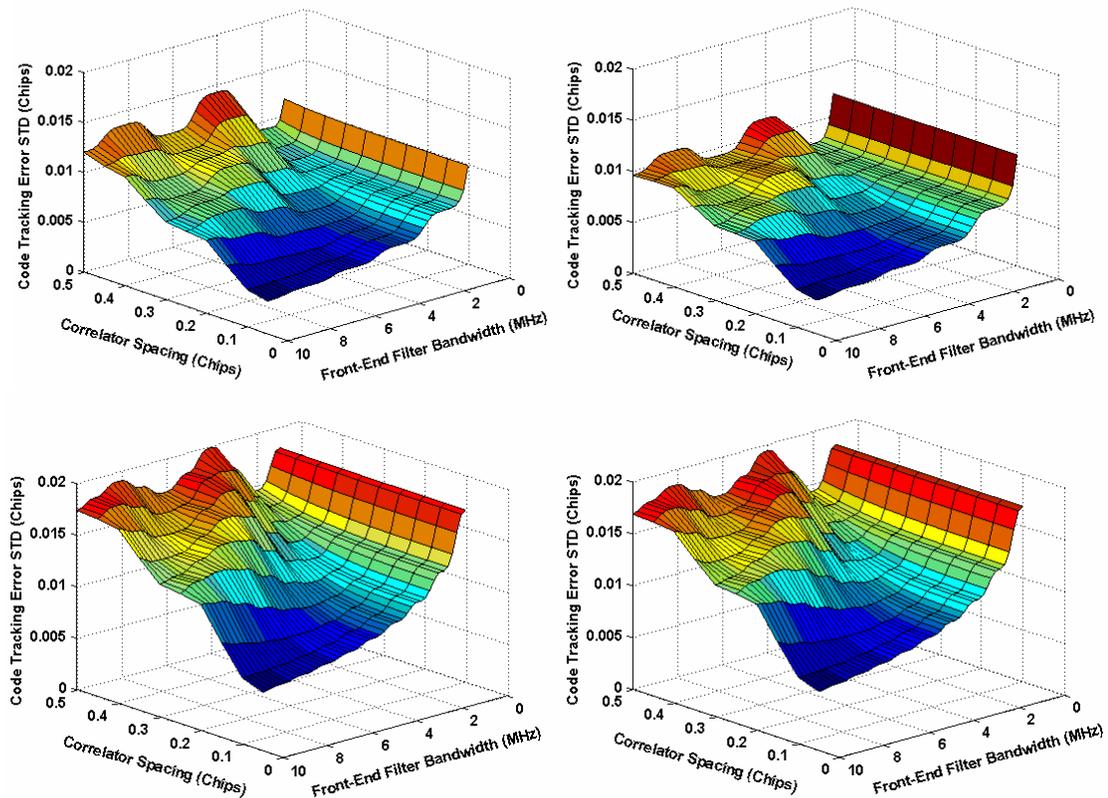
the DP discriminator, but not for the EMLP discriminator where they are larger for the sBOC(1,1) modulation.

It can also be seen that the DLL tracking error variance is sensitive to five main parameters:

- The shape of the spreading sequence PSD, or equivalently the slope of the spreading sequence autocorrelation main peak,
- The front-end filter bandwidth,
- The correlator spacing,
- The  $P/N_0$  (equivalently in this case, and more frequently referred to as  $C/N_0$ ), and
- The coherent integration time.

Figure 5.7 and Figure 5.8 give a good idea of the impact of the correlator spacing and the front-end filter bandwidth on the final code tracking error for two given  $C/N_0$  values (30 and 45 dB-Hz) and for the sBOC(1,1) and BPSK(1) modulations, based on a 4 ms coherent integration time. It can be seen that, in a general sense, the smaller the correlator spacing value and the wider the front-end filter bandwidth, the smaller the code tracking error will be. However, this is not completely true since some variations with the front-end filter bandwidth are visible. These variations are due to the autocorrelation function main peak shape imposed by the front-end filter, and its implication with the choice of the correlator spacing. This is discussed in greater details by Betz & Kolodziejewski (2000). Consequently, an appropriate selection of the pair “correlator spacing – front-end filter” should be realized in order to ensure optimal tracking conditions.

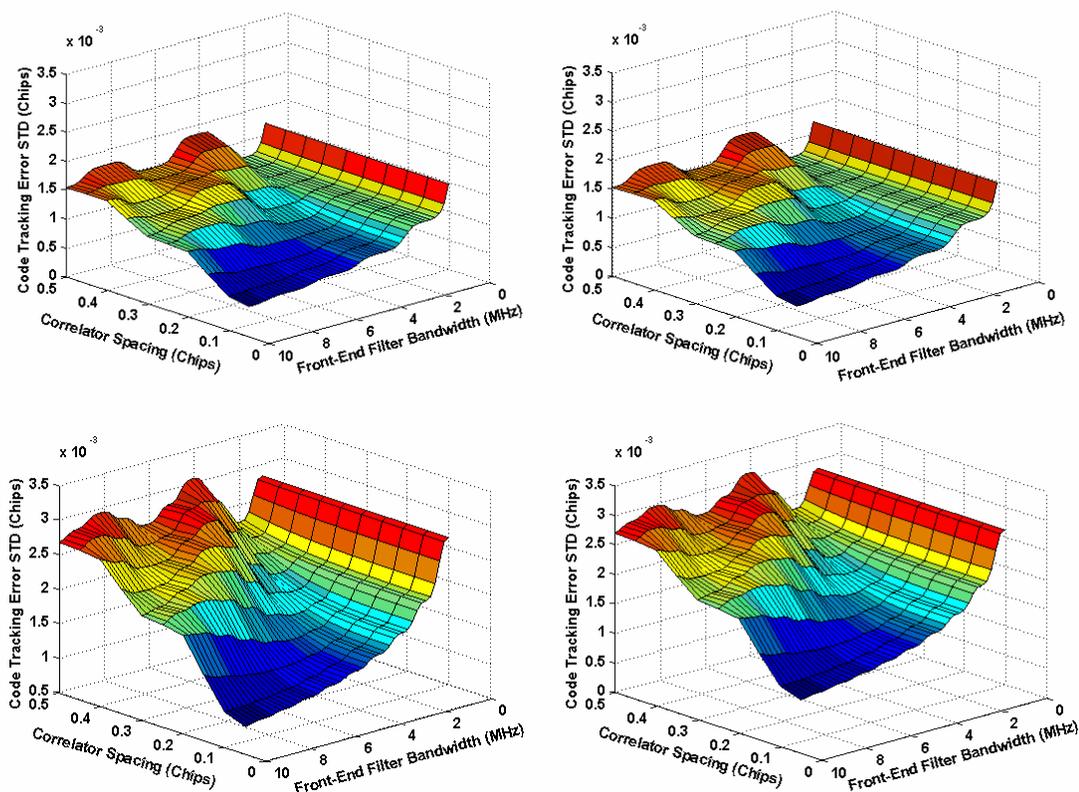
It is well known that the DP discriminator provides a better resistance to noise due to its lower squaring loss (Van Dierendonck 1996). However, in the conditions taken for Figure 5.7 and Figure 5.8, this can be only barely observed; this phenomenon would be more significant for lower SNR values.



**Figure 5.7 – Variation of the Code Tracking Error Standard Deviation using EMLP (Left) and DP (Right) Discriminators as a Function of the (One-Sided) Front-End Bandwidth and the Correlator Spacing sBOC(1,1) Signal (Top) and BPSK(1) (Bottom) for a  $C/N_0$  of 30 dB-Hz and a 4 ms Integration Time**

It is also interesting to observe that, for sBOC(1,1) modulation and for large correlator spacing values, the DP discriminator outperforms significantly the EMLP discriminator. This is a direct consequence of the limited correlator spacing value accepted by the

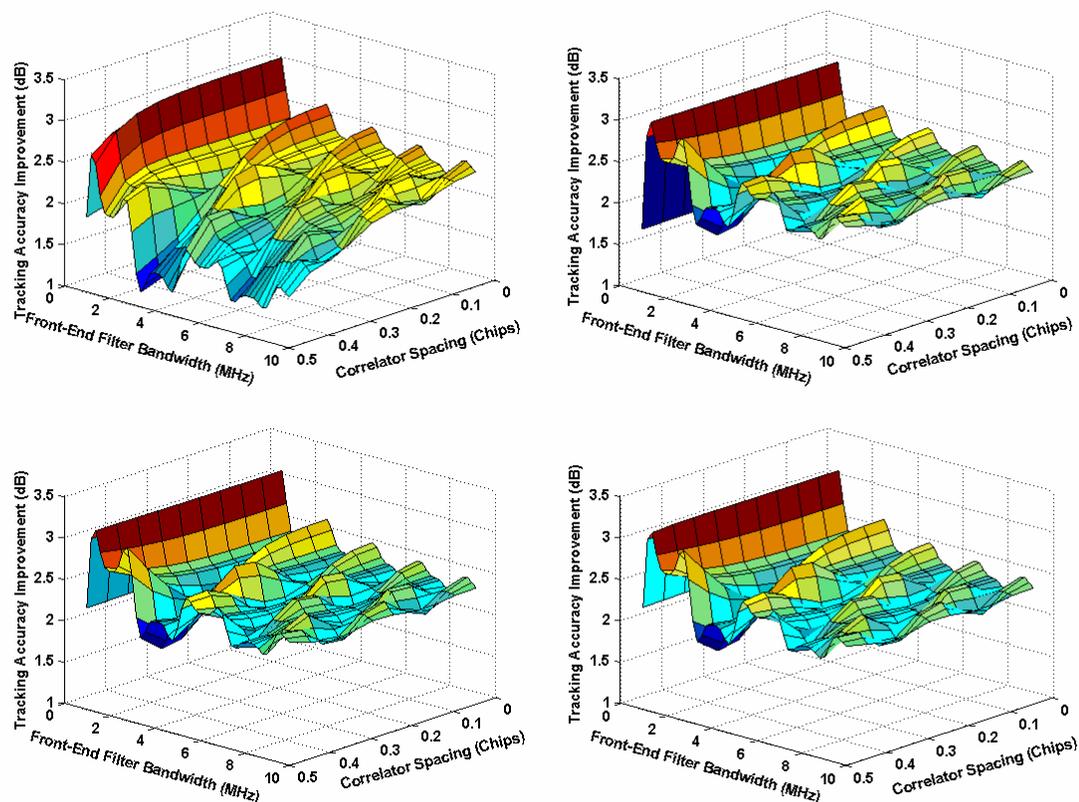
sBOC(1,1) EMLP discriminator (0.33 chips), as explained in Section 5.1.2, and as shown in Figure 5.4.



**Figure 5.8 – Variation of the Code Tracking Error Standard Deviation using EMLP (Left) and DP (Right) Discriminators as a Function of the (One-Sided) Front-End Bandwidth and the Correlator Spacing sBOC(1,1) Signal (Top) and BPSK(1) (Bottom) for a  $C/N_0$  of 45 dB-Hz and a 4 ms Integration Time**

In a general sense, it appears that sBOC(1,1) modulation outperforms BPSK(1) modulation in terms of tracking accuracy, as mentioned above for the case of an infinite front-end filter bandwidth. This result is confirmed in Figure 5.9. On average, and using the same values for the correlator spacing and front-end filter bandwidth, the Galileo L1F tracking error standard deviation gain over GPS C/A is significant, between 2 and 3 dB. To quantify the tracking accuracy in terms of Gaussian noise, for a 3 MHz one-sided

front-end filter bandwidth and a 0.2 chip correlator spacing, the code tracking error standard deviation for a 30 and 40 dB-Hz  $C/N_0$  is 1.83 and 0.52 m, respectively for the Galileo L1F signal; the corresponding values for the GPS C/A signal are 3.04 and 0.87 m, respectively.



**Figure 5.9 – Tracking Standard Deviation Improvement between sBOC(1,1) and BPSK(1) for Using EMLP (Left) and DP (Right) Discriminators for two  $C/N_0$  (30 (Top) and 45 (Bottom) dB-Hz) for a 4 ms Integration Time**

Based on Figure 5.5, Figure 5.6, the above analysis, and since it would be very tedious to present results for the full range of values of front-end filter bandwidth and correlator spacing, it was decided to show results only for a one-sided front-end filter bandwidth of 3 MHz; a correlator spacing of 0.2 chips is used, unless otherwise stated. These values correspond to a relatively narrow front-end filter that still takes more than the main lobes

of the Galileo L1F signal, thus having a reasonable filtering loss of approximately 0.5 dB, as shown in Figure 2.7. With a front-end filter of this type, a 0.2 chip correlator spacing was assessed as the minimum correlator spacing for the sBOC(1,1) signal situated outside the rounded part of the filtered autocorrelation peak. In both the Galileo L1F and GPS C/A signals, it seems to represent a good compromise between resistance to noise and the need for a low sampling frequency.

### 5.2.1.2 Impact of Discriminator Normalization

The previous theoretical study of the noise impact on the DLL tracking performance was based on the assumption of perfect normalization. However, in reality, as has been shown in Section 5.1.3, different types of normalization are used. It is then useful to analyze the impact of these normalization methods on tracking noise, and to compare them to the formulas given in Equations (5.32) and (5.33). The computation of the analytical formula of the tracking noise when normalization is used is extremely difficult due to the division operator. To overcome this, a statistical method is preferred herein. However, in order to employ this technique, one must be able to accurately generate the noise contribution, and particularly its correlation between the correlator outputs, as has been emphasized in Section 2.1.3. Hurst & Knop (1972) propose a way to generate such correlated Gaussian noise. The basic formula is:

$$N_{Corr} = V_C \sqrt{D_C} N_{Norm} \quad (5.36)$$

with  $CV_C = D_C V_C$

where  $N_{Norm}$  represents Gaussian noise,

$C$  is a positive definite matrix and represents the expected correlation between noise,

$V_C$  gathers the eigenvectors of  $C$ , and

$D_C$  is a diagonal matrix with the corresponding eigenvalues of  $C$ .

As explained in Section 2.1.3, the correlation function of the in-phase and quadra-phase Early, Prompt and Late correlation noise corresponds to the spreading symbol autocorrelation function. As shown in Appendix A, this function integrates filtering loss due to the limited front-end filter bandwidth (and neglects the impact of the I&D filter). Consequently, the matrix  $C$  must be taken equal to:

$$C = \begin{pmatrix} R_n(0) & R_n\left(\frac{\delta}{2}\right) & R_n(\delta) \\ R_n\left(-\frac{\delta}{2}\right) & R_n(0) & R_n\left(\frac{\delta}{2}\right) \\ R_n(-\delta) & R_n\left(-\frac{\delta}{2}\right) & R_n(0) \end{pmatrix} \quad (5.37)$$

Because  $C$  is positive definite as required, it is possible to use the proposed algorithm. The power of the simulated Gaussian noise follows Equation (2.15). The in-phase and quadra-phase correlation values are simulated, assuming that the phase and frequency errors are null. It is also important to remember that the in-phase and quadra-phase correlation noise components are uncorrelated. Once the correlated noise values are generated, it is possible to statistically deduce the mean and the variance of the

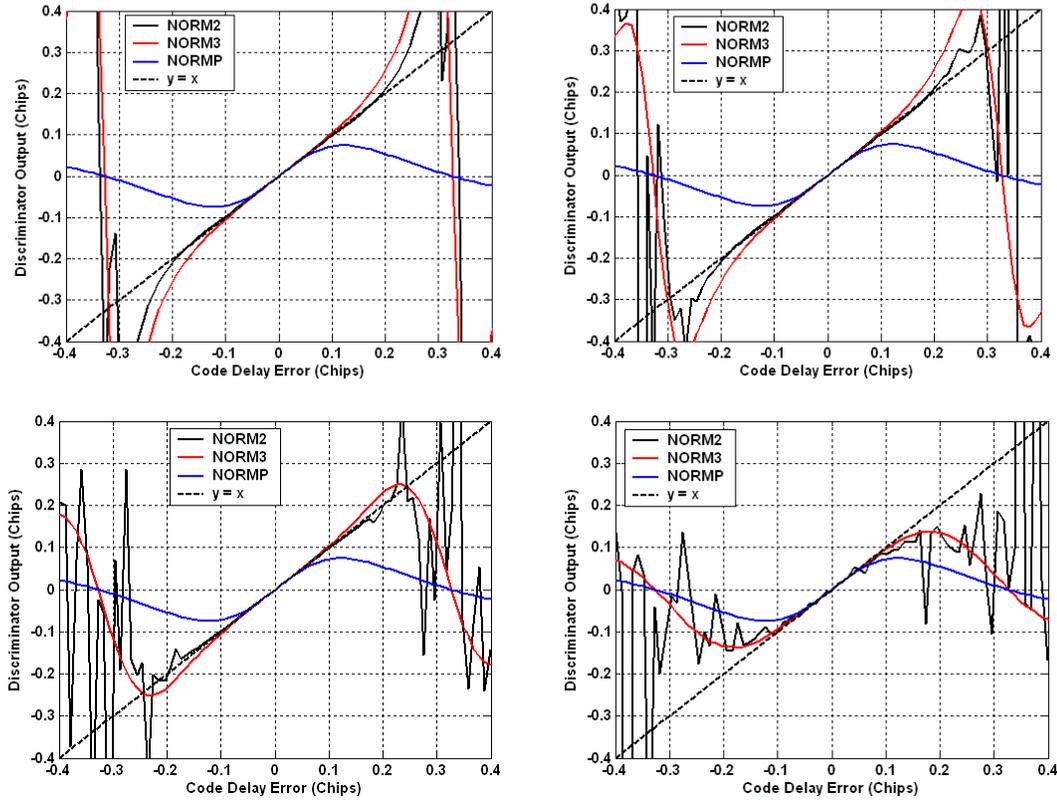
normalized discriminator output through Monte Carlo simulations. Then, using Equation (5.31), it is easy to compute the resulting tracking loop accuracy.

Figure 5.10 shows the mean sBOC(1,1) DP discriminator output using the three normalizations considered, a 4 ms integration time and four different  $C/N_0$  values (30, 35, 40, and 45 dB-Hz). Except for the case of perfect normalization, the impact of the noise is obvious as  $C/N_0$  decreases. It can be noticed in two ways:

- for certain normalizations ( $N_2$  in this example), the mean discriminator output has a ‘noisy’ shape that progressively impacts the linear tracking region as  $C/N_0$  decreases; and
- Outside the linear tracking domain, the mean discriminator output tends to flatten as noise increases, leading toward a less favourable mean discriminator output shape in case of sudden high code delay estimation error, or high dynamics.

It also appears that the impact of the noise on the normalization (the discriminator normalized by  $N_3$ ) seems to resist Gaussian noise better than as normalized by  $N_2$ . This is actually due to the presence of the vertical asymptote at the discriminator output stability domain boundary. One interesting way to realize this is to note that the sBOC(1,1) DP discriminator normalized by the ideal normalization does not have an output that is affected by the level of the SNR. These asymptotes have been observed already in Figure 5.5, and in theory, they lead to a limited maximum error that depends on the normalization factor. It is the magnitude of that maximum error that affect the

impact of the noise on the discriminator output, explaining the difference between the use of  $N_3$  and  $N_2$ .



**Figure 5.10 –sBOC(1,1) DP Discriminator using a 3 MHz One-Sided Front-End Filter, a 0.2 Chip Correlator Spacing, and  $C/N_0$  values of 45 (Top Left), 40 (Top Right), 35 (Bottom Left), and 30 (Bottom Right) dB-Hz**

As discussed in Section 4.1.3, one way to deal with the vertical asymptote problem is to set a discriminator hard limiter, such as:

$$\begin{aligned}
 HL(x) &= k \operatorname{sgn}(x) & \text{if } |x| &\geq k \\
 HL(x) &= x & \text{if } |x| &< k
 \end{aligned} \tag{5.38}$$

Doing so removes the chances of a sudden jump that can cause a loss of lock and allows for the limitation of high magnitude noise samples. One way to choose the discriminator

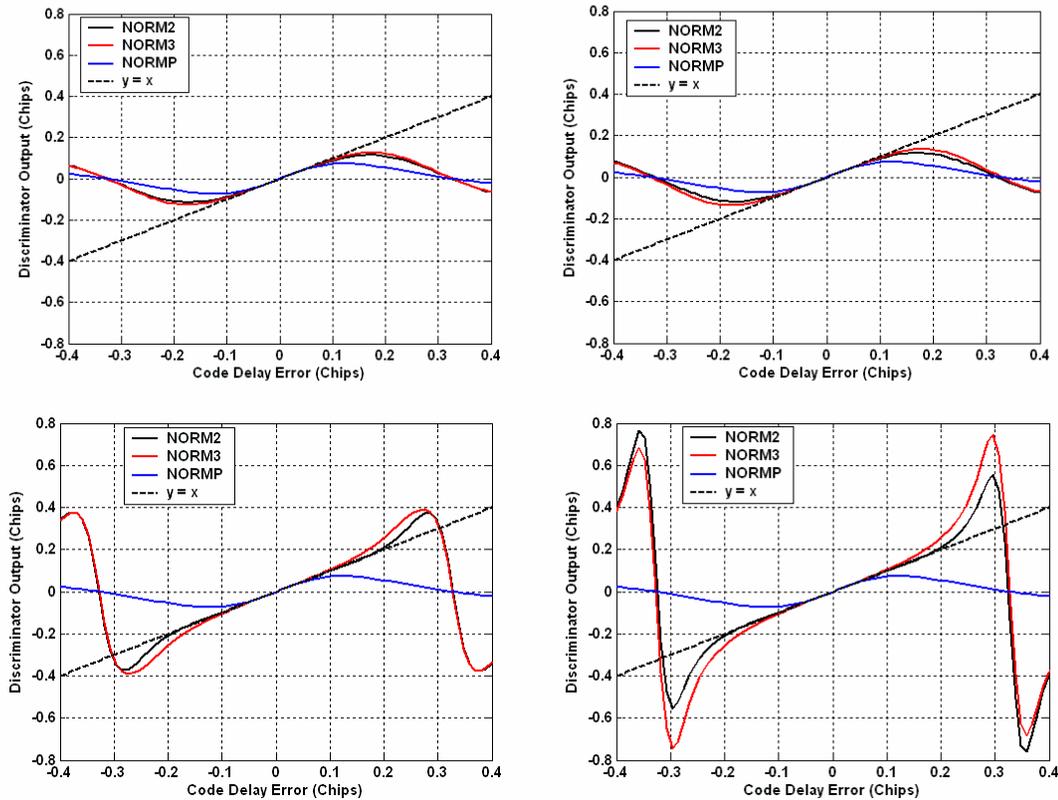
hard limiter value is to look for the maximum discriminator output at the loop tracking sensitivity boundary in the presence of Gaussian noise only through the use of perfect normalization. The DLL tracking error under perfect normalization has been shown in Figure 5.7 and Figure 5.8. Looking at the specific case of a one-sided front-end filter of 3 MHz and a 0.2 chip correlator spacing, it is then possible to assess the tracking threshold with Gaussian noise only using (Ward 1996):

$$3\sigma_{\varepsilon_{\tau,t}} \leq \frac{L_{\tau}}{2} \quad (5.39)$$

where  $L_{\tau}$  is the two-sided discriminator linear tracking domain.

Assuming a DLL loop bandwidth of 1 Hz, the discriminator hard limiter was assessed to be 5 chips for both the GPS C/A and Galileo L1F. Figure 5.11 shows the resulting mean normalized discriminator output using a 0.4 chip and 5 chip hard limiter. It can be seen in both cases that, for a low  $C/N_0$ , the discriminator shape is identical. However, for higher signal strengths, the choice of the discriminator hard limiter value impacts the discriminator output shape at the boundary of the stability domain. In the two cases shown, large code delay error estimation will lead to overestimation of the input error; however, a larger hard limiter value will allow that overestimation (needed for stability of the tracking loop for large errors) for lower  $C/N_0$  values, which is always desirable, especially in harsh conditions. The choice of a large hard limiter value might seem adequate also because it will start changing the Gaussian-shape of the discriminator output error for a lower  $C/N_0$  than would a small hard limiter value which, as previously

shown for the arctangent discriminator in the PLL study (Section 4.3.1), is beneficial for the DLL tracking sensitivity.



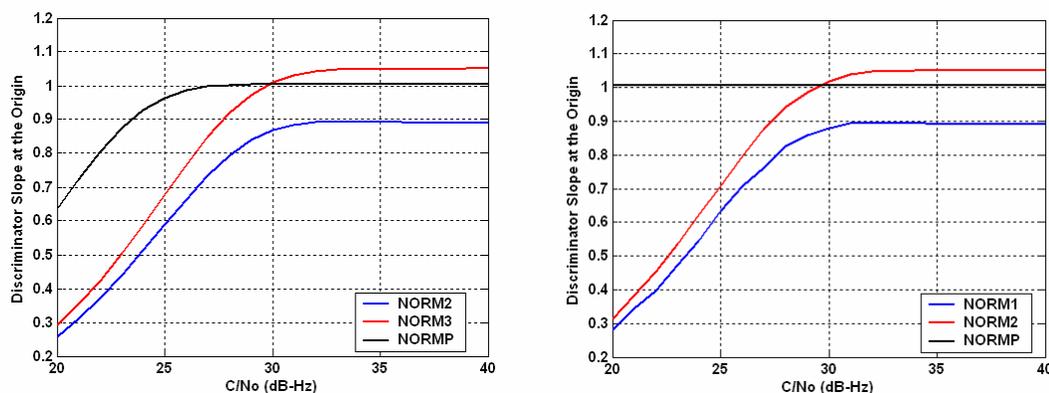
**Figure 5.11 - sBOC(1,1) DP Discriminator using a 3 MHz One-Sided Front-End Filter, a 0.2 Chip Correlator Spacing, a 45 (Top) and 30 (Bottom) dB-Hz and a 0.4 (Left) and 5 (Right) Chip Discriminator Hard Limiter**

However, a high discriminator hard limiter value also means that, at the stability domain boundary, the discriminator output value will exhibit a very sharp sign change for high  $C/N_0$  values. Consequently it means that, for a code error approaching the stability domain boundary, the loop might be more unstable than if the shape of the discriminator output was rounder (as in the case of the arctangent discriminator in the PLL). Once again, if relevant carrier aiding is used, which is likely for high SNR values, this should not be a problem.

Another problem linked to the presence of noise is that the type of normalization ( $N_1$ ,  $N_2$ , or  $N_3$ ) will result, for low  $C/N_0$ , in a biased estimation of the signal power, since they are all based on the product of correlation values (for instance,  $N_3$  has a non-central chi-square distribution). Consequently, they will all tend, at some point, to overestimate the slope of the normalized discriminators. This must be borne in mind since it will very likely degrade the DLL tracking performance. Figure 5.12 represents the sBOC(1,1) mean normalized DP discriminator slope at the origin for a narrow (0.4 chips) and large (5 chips) discriminator hard limiter. For each type of normalization, the slope decreases at some point due to overestimation of the incoming signal power by the different normalizations. This means that, below a certain  $C/N_0$  defined by the slope inflexion, the discriminator output will systematically underestimate the input error, and the DLL will not work properly. It can be seen that, for both hard limiter values, the discriminator slope at the origin starts decreasing at the same  $C/N_0$  value. This implies that the choice of an optimal value for the hard limiter does not bring any improvement in terms of tracking sensitivity, as it would be expected, since the discriminator slope decreases too early when using the normalizations  $N_1$ ,  $N_2$ , or  $N_3$ . However, it can be seen that using perfect normalization, the selection of a higher hard limiter value is beneficial in terms of degradation of the discriminator slope at the origin, following the explanation given previously. However, when using perfect discriminator normalization, it has been discussed that no hard limiter is necessary.

The fact that the impact of the hard limiter is limited in the case studied encourages the use of smaller hard limiter thresholds. Indeed, using a smaller hard limiter would then

allow reduction of the level of the correlator outputs' noise. This has to be done carefully, however since, if too small a level is chosen (i.e., lower than the stability domain), it could significantly impair tracking.

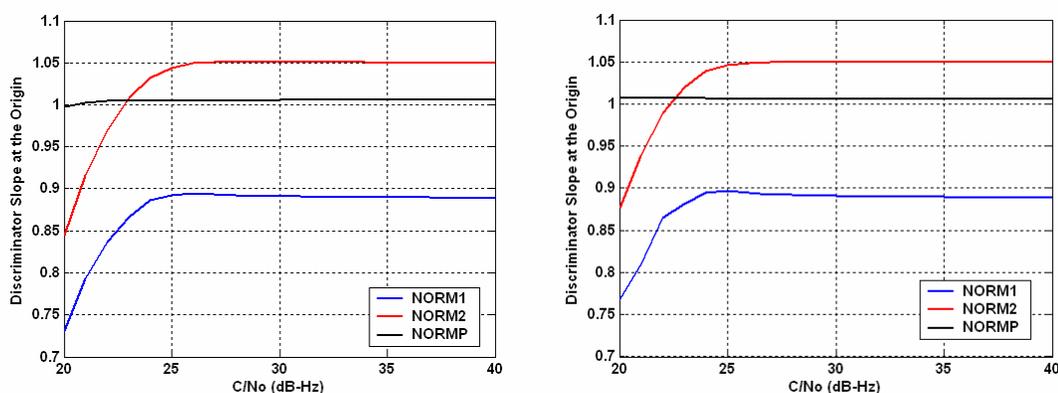


**Figure 5.12 – sBOC(1,1) DP Discriminator Slope at the Origin for a 3 MHz One-Sided Front-End Filter, a 4 ms Integration Time, and a 0.2 Chips Correlator Spacing for a 0.4 (Left) and 5 (Right) Chips Discriminator Hard Limiter**

It must be noted that, in Figure 5.12, for high  $C/N_0$  values, the value of the slope at the origin does not exactly match the value 1 as it should for the two non-ideal normalizations used. This is mainly due to front-end band-limiting, since it was assumed that these normalizations could be extrapolated from the infinite front-end bandwidth assumption. The value of the offset should be estimated and corrected to facilitate optimal tracking.

It is easy to understand that the reduction of the mean discriminator output slope at the origin would be very detrimental for the DLL, since it means that the code delay error will always be underestimated. Consequently, as already explained for the PLL and its arctangent phase discriminators in Section 4.3.1, the  $C/N_0$  at which the slope begins to diminish should be considered as a hard  $C/N_0$  threshold. One easy way to extend this hard

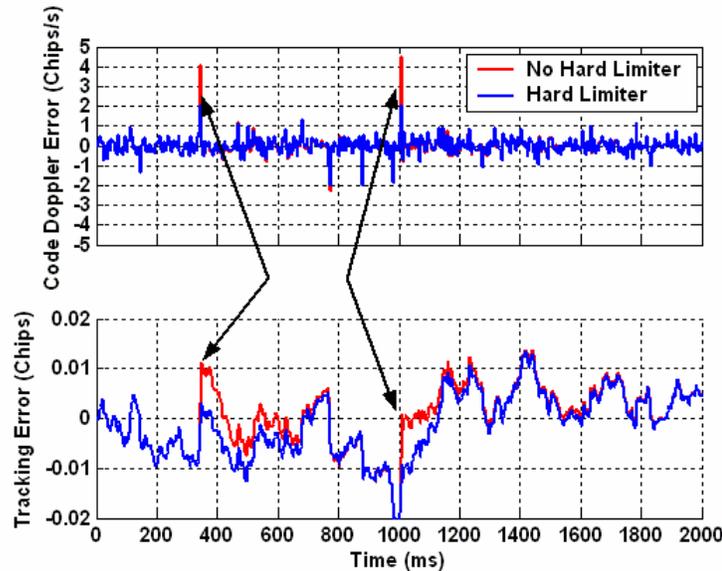
threshold is to increase the coherent integration time. By doing so, the post-correlation SNR will increase, and  $N_1$ ,  $N_2$ , and  $N_3$  will produce better estimates of the signal power for the same  $C/N_0$ , lowering the hard  $C/N_0$  threshold as shown in Figure 5.13 for a 20 ms integration time. Another method would consist of correcting the mean discriminator slope according to the estimated  $C/N_0$ . However, this has not been studied in this thesis (and it introduces the specific problem of the  $C/N_0$  estimate accuracy).



**Figure 5.13 – sBOC(1,1) DP Discriminator Slope at the Origin for a 3 MHz One-Sided Front-End Filter, a 20 ms Integration Time, and a 0.2 Chips Correlator Spacing for a 0.4 (Left) and 5 (Right) Chips Discriminator Hard Limiter**

Following the previous discussion, it has been decided to set the hard limiter to a value slightly higher than the stability domain in order to have a discriminator output that would allow, for medium to high SNR values, returning to the lock point even when the code delay tracking error is at the edge of the stability domain. The fact that the value chosen is slightly higher than the stability domain also allows minimization of noise level. On this basis, it has been decided to use a discriminator output hard limiter value of 0.4 chips for the sBOC(1,1) case and 1.2 chips for the BPSK(1) case. The direct impact of such a choice is shown in Figure 5.14 for the Galileo L1F signal, for a  $C/N_0$  of 30 dB-

Hz, a DLL loop bandwidth of 1 Hz, and a coherent integration time of 4 ms, using a discriminator hard limiter of 0.4 chips. It can be seen that the use of the hard limiter filters out some high discriminator errors, avoiding sudden tracking jumps.

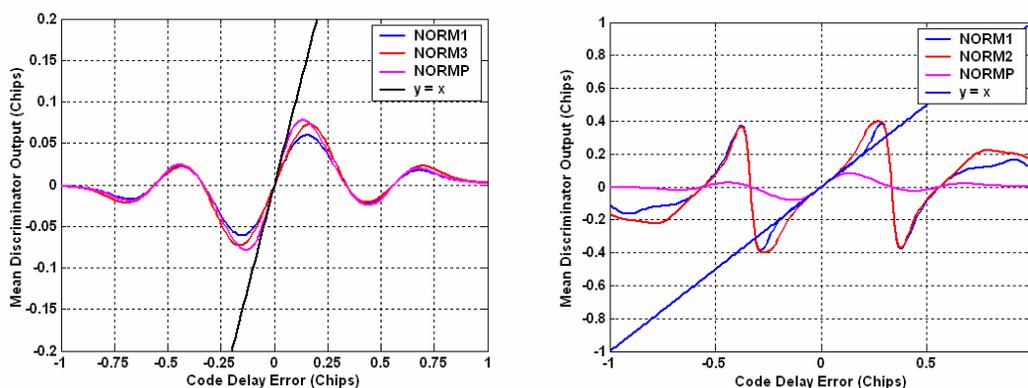


**Figure 5.14 – Galileo L1F Receiver DLL Response under Gaussian Noise with a  $C/N_0$  of 30 dB-Hz, a 4 ms Integration Time, a 1 Hz Loop Bandwidth With and Without a 0.2 Chips Discriminator Hard Limiter**

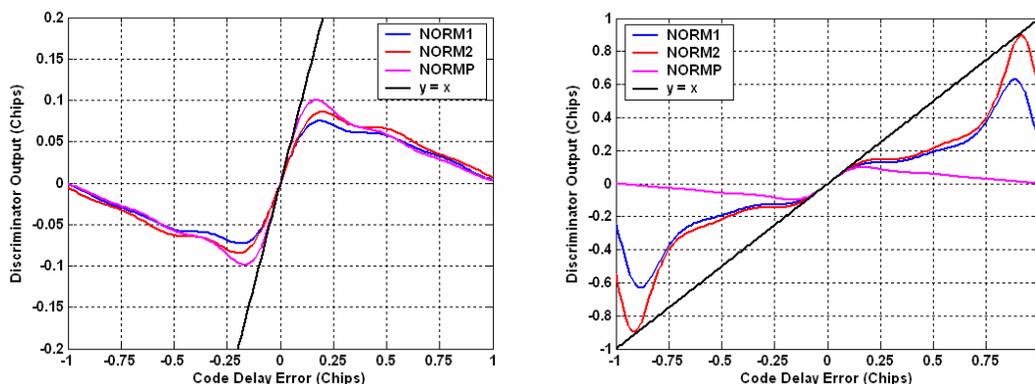
Figure 5.15 and Figure 5.16 show the mean EMLP discriminator output shape for the sBOC(1,1) and BPSK(1), respectively, for a  $C/N_0$  of 25 and 45 dB-Hz with a correlator spacing of 0.2 chips and a 3 MHz one-sided front-end filter. The Galileo L1F and GPS C/A figures are shown with a discriminator output hard limiter of 0.4 and 1.2 chips, respectively.

It is important to reiterate at this point that the curves associated with the use of the normalization  $N_P$  are not affected directly by the hard limiter, allowing for higher tracking sensitivity. However, as previously mentioned, the discriminator output shape associated with this normalization is not optimal since it would always underestimate the

code delay error when the input error is outside the linear tracking region, even for high SNR values. This is undesirable since it makes the loop susceptible to high dynamics and thus less robust in application. This is why, for a typical Galileo L1F receiver, which is intended to work in all conditions, it would be preferable to use one of the three normalizations discussed above ( $N_1$ ,  $N_2$ , or  $N_3$ ).



**Figure 5.15 - sBOC(1,1) DP Discriminator Output Using a 0.4 Chips Hard Limiter, a 3 MHz One-Sided Front-End Filter Bandwidth, a Correlator Spacing Value of 0.2 Chips for a  $C/N_0$  of 25 dB-Hz (Left) and 45 dB-Hz (Right)**



**Figure 5.16 – BPSK(1) DP Discriminator Output Using a 1.2 Chips Hard Limiter, a 3 MHz One-Sided Front-End Filter Bandwidth, a Correlator Spacing Value of 0.2 Chips for a  $C/N_0$  of 25 dB-Hz (Left) and 45 dB-Hz (Right)**

It has been seen that low  $C/N_0$  values provide the tracking loop with less favourable code delay error estimation since, for input errors outside the linear tracking region, the error will be systematically underestimated on average (note the difference in the y-axis scale between the 25 and 45 dB-Hz cases in Figure 5.15 and Figure 5.16). Moreover, as already seen, the lower the signal strength, the more the mean discriminator slope at the origin will deviate from the first diagonal due to the normalization. This means that, for low levels of  $C/N_0$ , not only will the noise become (obviously) more problematic, but the linear tracking domain will shrink, resulting in far less robust tracking. For low signal strength, it seems that all the normalizations are fairly equivalent in terms of linear tracking region (if tracking is possible). However, for higher levels of  $C/N_0$  and for the DP discriminator, normalization by  $N_2$  seems to provide the widest linear tracking region, as shown in Figure 5.15 (and this is true also for  $N_1$  with sBOC(1,1) EMLP discriminator). On the other hand, the use of normalization  $N_3$  creates a mean discriminator output that tends to overestimate the input error for large errors within the stability domain, which will be taken into account by the discriminator hard limiter, so it is not a major drawback, although it should result in extra noise in the tracking process.

Figure 5.15 and Figure 5.16 also show that deducing a standard deviation from these statistical tests has to be done carefully. Indeed, if the discriminator output leads to a systematic biased error estimate (for instance, when the discriminator slope at the origin differs from 1), then the tracking will be greatly degraded, and the computation of the variance through Equation (5.31) will lead to incorrect results. Consequently, the results that will be shown hereafter are valid only when the discriminator output slope at the

origin matches the first diagonal (or, in other words, is equal to 1). To do so, hard  $C/N_0$  thresholds must be defined. The hard  $C/N_0$  thresholds are defined herein by the minimum  $C/N_0$  such that the discriminator slope at the origin is higher than 0.9 (and equal to 1 for the second case). These thresholds are shown in Table 5.1 for integration times of 4 and 20 ms, respectively. It is apparent that the hard  $C/N_0$  thresholds are very similar for the sBOC(1,1) and the BPSK(1), since they depend upon the same normalizations. The use of a longer coherent integration time permits significant lowering of the thresholds.

**Table 5.1 – Hard  $C/N_0$  Threshold due to the Decrease of the Mean Discriminator Slope at the Origin (for Slope equal to 0.9 on the Left and 1 on the Right)**

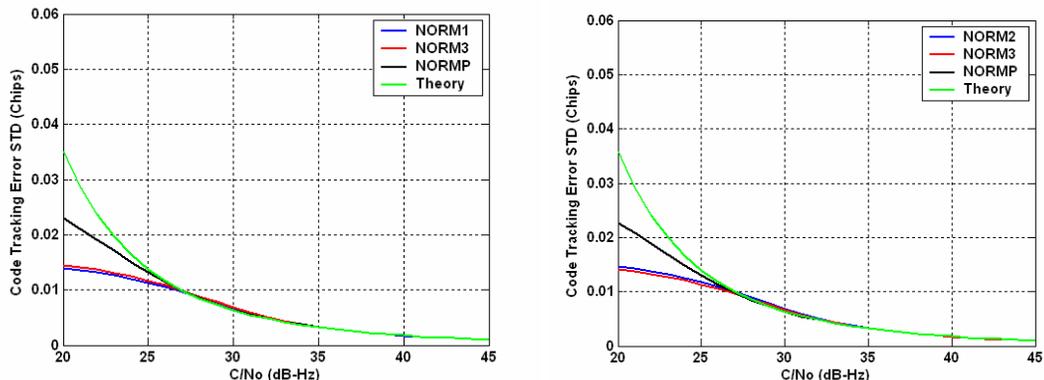
			Threshold (dB-Hz)	
			4 ms (0.9 - 1)	20 ms(0.9 - 1)
sBOC(1,1)	DP	NORM1	28.5 - 33	22-27
		NORM2	28.5 - 33	22-27
	EMLP	NORM1	28.5 - 33	22-27
		NORM2	28.5 - 33	22-27
BPSK(1)	DP	NORM1	28.5 - 33	22-27
		NORM2	28.5 - 33	22-27
	EMLP	NORM1	28.5 - 33	22-27
		NORM2	28.5 - 33	22-27

Figure 5.17 and Figure 5.18 show the resulting code tracking error standard deviation for both modulations: a 0.4 and 1.2 chips discriminator hard limiter, a correlator spacing value of 0.2 chips, a one-sided front-end filter bandwidth of 3 MHz, and all of the normalized discriminators. The results are obtained after correction of the mismatch of the discriminator slope value at the origin shown in Figure 5.12. The empirical estimation of the code tracking error variance, when using the proposed normalization, was verified through software simulations. The results of the 20-second simulations with the Galileo

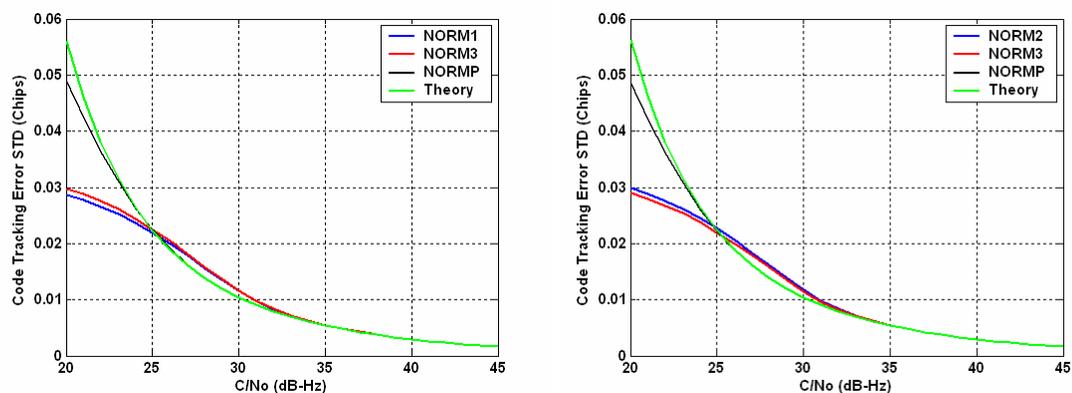
L1F signal, a 0.2 correlator spacing, a 3 MHz one-sided front-end filter, a 2 Hz DLL loop bandwidth and using a DP discriminator, are shown in Figure 5.19. They seem to confirm the theoretical results.

It can be seen from Figure 5.17 and Figure 5.18 that the normalization does not play an important role in the final code tracking error standard deviation. This is due to the use of a tight discriminator hard limiter and to the fact that, when the noise coming from the normalization really impacts the discriminator output, it means that the slope at the origin changes and thus, that the hard  $C/N_0$  threshold is reached. Since the hard limiter was chosen at a greater level for the BPSK(1) modulation, it means that the noise coming from the normalization will be more dominant than in the sBOC(1,1) case. This can be seen in Figure 5.18. Once again, it must be kept in mind that the code tracking error values shown in Figure 5.17 and Figure 5.18 and corresponding to  $C/N_0$  values below the hard threshold have to be disregarded since, in reality no robust tracking could then be achieved. In quiet conditions, tracking with a slightly degraded slope at the origin is most likely possible, but should be avoided.

It has to be emphasized that the perfect normalization  $N_P$  would not require any discriminator hard limiter, as already explained. This is confirmed when looking at Figure 5.17 and Figure 5.18, where the code tracking error standard deviation associated with the normalization  $N_P$  follows the theoretical values for lower  $C/N_0$  values and for higher discriminator hard limiter values more closely than when another normalization is used.



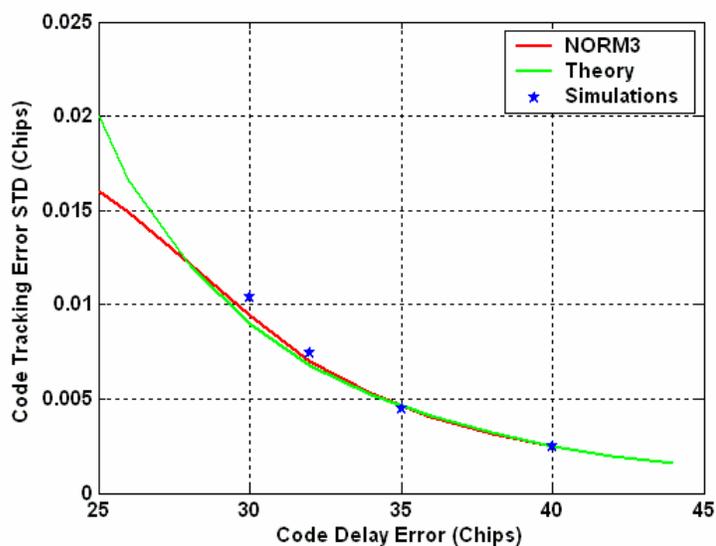
**Figure 5.17 – Code Tracking Error Standard Deviation with sBOC(1,1) Modulation, with a 0.4 Chips Hard Limiter for EMLP (Left) and DP (Right) Normalized Discriminators Using a 3 MHz One-Sided Front-End Filter Bandwidth, a 0.2 Chips Correlator Spacing and an Integration Time of 4 ms**



**Figure 5.18 – Code Tracking Error Standard Deviation with BPSK(1) Modulation, with a 1.2 Chips Hard Limiter for EMLP (Left) and DP (Right) Normalized Discriminators Using a 3 MHz One-Sided Front-End Filter Bandwidth, a 0.2 Chips Correlator Spacing and an Integration Time of 4 ms**

The code tracking error associated with the sBOC(1,1) modulation is, as expected, significantly lower than with the BPSK(1) modulation. However, it is important not to take the figures given in Table 5.1 as the exact constrained values. Indeed, it is possible to have discriminators with less noise than the one studied herein by simply using a longer coherent or non-coherent integrations for the normalization factors only (independently from the coherent integration time used for the not-normalized

discriminator). Doing this will lower the hard  $C/N_0$  threshold significantly. However, it will not increase the difference between the Galileo L1F and GPS C/A signals' hard thresholds.



**Figure 5.19 – Simulated, Theoretical and Empirical Code Tracking Error Standard Deviation with Galileo L1F Signal using a 0.4 Chips Hard Limiter, a DP Normalized Discriminators with a 3 MHz One-Sided Front-End Filter Bandwidth, a 0.2 Chips Correlator Spacing and an Integration Time of 4 ms**

### 5.2.1.3 Conclusion

This DLL analysis has shown that, for code delay tracking in terms of Gaussian noise disturbance, the Galileo L1F signal presents several structural advantages over the GPS C/A signal that can be summarized as:

- A lower estimated code error standard deviation by 2 to 3 dB,
- A more robust discriminator output shape,
- The use of potential very long integrations that allow further mitigation of discriminator noise and that have a very low hard  $C/N_0$  threshold when using one of the non-ideal normalization proposed herein, and
- The use of a lower discriminator hard limiter since its stability domain is narrower, without impairing the loop robustness.

On the other hand, GPS C/A signal has also several advantages over Galileo L1F tracking, namely:

- A potentially wider stability domain, and
- A wider choice of correlator spacing values for robust tracking.

It has been shown that the choice of normalization was not critical, provided that the potential mismatch of the discriminator slope at the origin and the hard  $C/N_0$  thresholds are well assessed. For these reasons, and because of its more modest requirement in terms of correlators, the DP discriminator normalized by  $N_3$  will be used as the reference discriminator in the remainder of this thesis.

## 5.2.2 Multipath

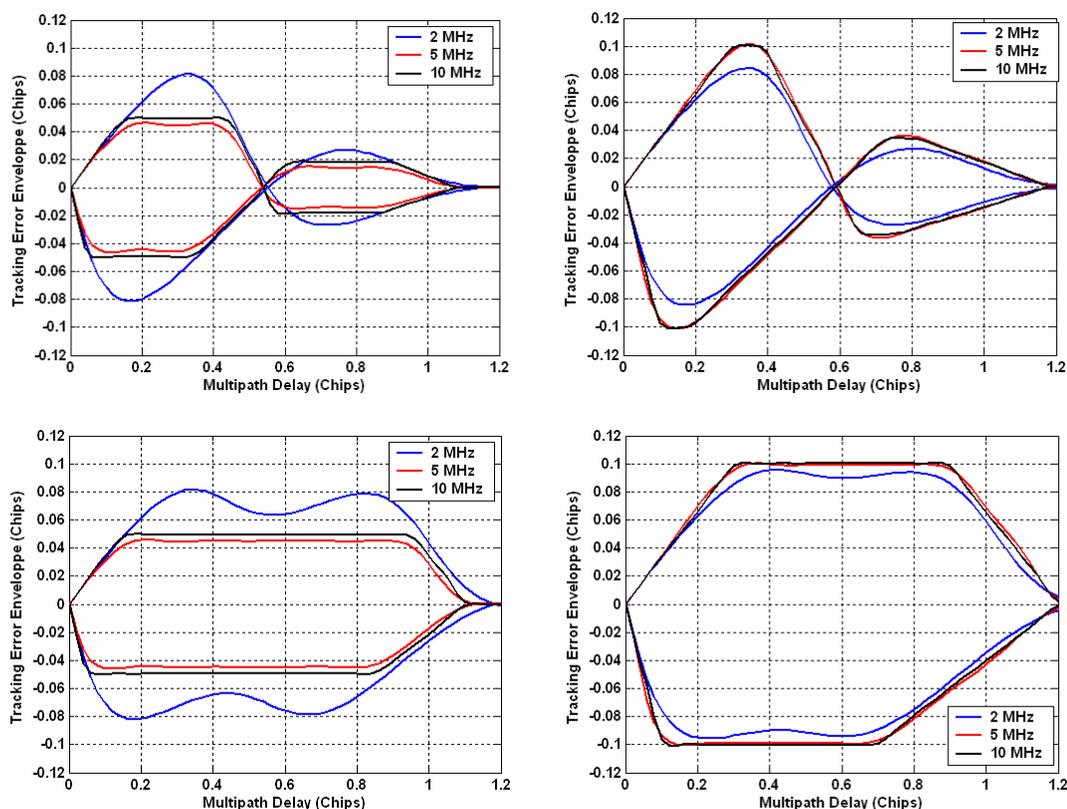
The origin of multipath has already been discussed in Section 3.1.5. The impact of multipath on code tracking accuracy is often represented as an error envelope representing the maximum error resulting from one single multipath with a certain phase,

delay and amplitude. The same method of analysis applied to the multipath-induced error will be used for the discriminators considered herein. It is important to note that, when computing the multipath-induced code tracking error envelope, it is equivalent to finding the point where the discriminator output crosses the origin, since this means that it represents the stability point where the loop will lock. Consequently, a study of the discriminator without normalization is sufficient for this purpose.

Figure 5.20 shows the code tracking error envelope for the sBOC(1,1) and BPSK(1) modulation using a DP discriminator, assuming a multipath with half the magnitude of the direct signal for two correlator spacing values (0.2 and 0.4 chips). It should be noted that the multipath-error code tracking envelope is exactly the same when using an EMLP discriminator. The sBOC(1,1) offers better resistance to long delay multipath than the BPSK(1) tracking; however, it has the exact same envelope for short delay multipath. It is obvious that both the front-end filter bandwidth and the correlator spacing have an influence on the envelope. A large correlator spacing will result in a greater susceptibility of the tracking loop with respect to multipath. Usually, a narrow front-end filter will tend to increase the multipath-induced error envelope. However, this is not always the case, as shown in Figure 5.20. In general, however, the wider the front-end filter bandwidth, the better the tracking.

When compared to the thermal noise impact on code tracking accuracy studied in Section 5.2.1 and shown in Figure 5.17, it appears that multipath is (in the case of a high magnitude multipath) a very serious source of tracking error. This is particularly true for Galileo L1F and GPS C/A signals because they use a low spreading symbol rate (0.1

chips represents approximately 30 metres). The use of a higher rate, as employed by GPS L5 or GPS P(Y) signals would provide better inherent multipath mitigation (Braasch 1996).



**Figure 5.20 – Code Tracking Error Envelope for sBOC(1,1) (Top) and BPSK(1) (Bottom) Modulations for One Multipath with an Amplitude Half of the Direct Signal, for a DP Discriminator, Several Front-End Filters, and a Correlator Spacing of 0.2 (Left) and 0.4 (Right) Chips**

One way to mitigate multipath-induced tracking errors is to lower the loop bandwidth as much as possible (but without impairing the tracking reliability). Indeed, in general, and especially in changing environments, the multipath-induced error time-correlation is small (Lachapelle *et al.* 2003). Consequently, if this time-correlation is higher than the DLL loop response (specified by the choice of the loop bandwidth), the DLL will not be

able to track the multipath-induced error, and it will then inherently result in a more accurate tracking.

### 5.2.3 Dynamic Stress Error

The impact of signal dynamics on the tracking loop has already been discussed in Section 4.3.2 for the PLL, and the same remarks apply to the DLL. However, the influence of the signal dynamics in the DLL might be different when carrier-aiding is used, as discussed in Section 5.1.1. Indeed, as already explained, in most of the current receivers, the DLL receives aiding from the PLL to model the Doppler variation. Since the PLL provides an accurate estimation of the signal Doppler, as shown in Section 4.4, the signal dynamics that could affect the DLL are then absorbed by the PLL aiding. Consequently, using such an architecture, the actual impact of the signal dynamics on the DLL is negligible. As a result, a first order DLL, only sensitive to velocity stress is usually used to account for the uncertainty on the PLL aiding, or to cope with the divergence of the ionospheric effect on the code and phase group delay. In this regard, and as already specified, in the frame of this thesis only a first order DLL will be investigated, assuming a negligible error due to signal dynamics.

If no carrier-aiding from the PLL is used, Stephens & Thomas (1995) show that the impact of the signal dynamics on a first order DLL is given by:

$$\theta_{\varepsilon_{\tau,d}} = \frac{T_I}{K_1} \frac{dR}{dt} \quad (5.40)$$

where  $K_1$  corresponds to the parameter of the loop filter given by Stephens & Thomas (1995), and

$\frac{dR}{dt}$  is the first order component of the code group time delay (chips/sec).

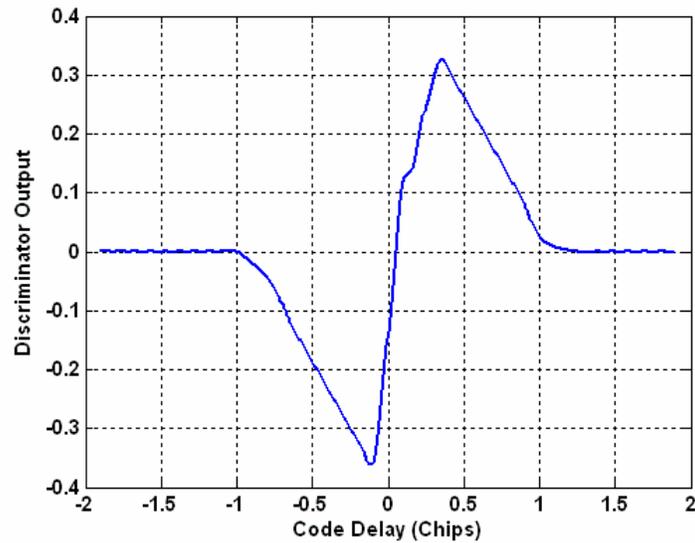
### 5.3 DLL Sensitivity Analysis

As for the PLL, the rule of thumb when looking at the DLL tracking threshold is to have 3-sigma of all the errors within the linear tracking region, which, in theory is  $\pm \frac{\delta}{2}$  chips.

Consequently, this is given by (Ward 1996):

$$3\sigma_{\varepsilon_{\tau,th}} \leq \frac{\delta}{2} \quad (5.41)$$

It is important to note that the multipath-induced tracking error is not included in Equation (5.41). Indeed, the tracking error induced by multipath can be seen as a bias that will shift the discriminator output stable point away from where it would otherwise be without multipath. However, it does not imply a tracking error in the sense that it would push the tracking loop away from its stability point. Figure 5.21 shows the example of the BPSK(1) DP discriminator in the presence of a single in-phase multipath with a delay of 0.25 chips and half the magnitude of the direct signal. It can be seen that the linear tracking domain is approximately the same, but the location where the output crosses the zero is shifted from the expected code error. Consequently, in theory, for a single multipath, it cannot be considered as directly affecting the code tracking sensitivity.



**Figure 5.21 – BPSK(1) DP Discriminator Output Assuming the Presence of One In-Phase Multipath with a Delay of 0.25 Chips and a Half the Magnitude of the Direct Signal**

However, in practice, a high number of low-amplitude multipath could be equivalent to adding extra noise to the discriminator output. It also happens that, for certain code delays, the multipath will slightly shorten the linear tracking region. But these effects are hard to quantify because they depend on the specific multipath environment and, so, are not considered herein.

It has been seen that, for most of the time, the stability domain is significantly larger than the linearity domain and, consequently, Equation (5.41) might seem overly pessimistic (Ward 1996). However, one has to realize that, when the SNR decreases, it also results in a less favourable discriminator output shape and, a significantly reduced stability domain, as seen in Figure 5.15 and Figure 5.16 for the case of a DP discriminator. Thus, Equation (5.41) remains a good way to assess the tracking sensitivity of the loop.

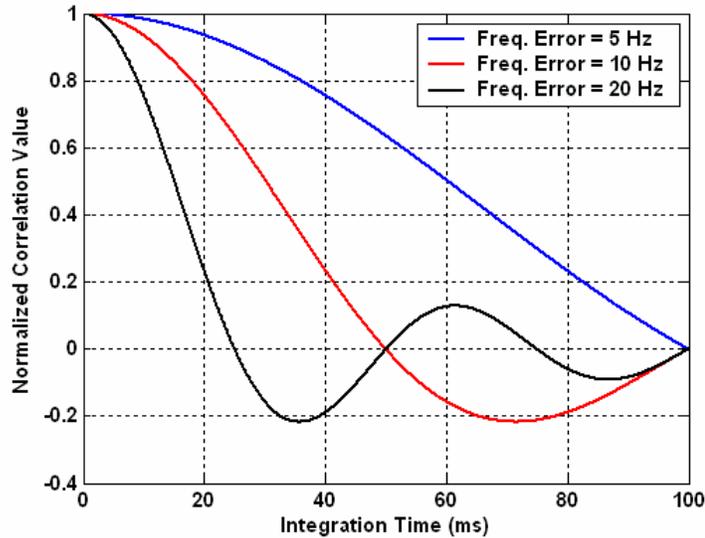
One of the main parameters that may be adjusted in order to test the Galileo L1F receiver DLL sensitivity is the coherent integration time. Indeed, with the availability of a pilot channel, it is theoretically possible to integrate for very long periods of time. This will provide a significant increase in the post-correlation SNR that will result in increased sensitivity.

In terms of tracking loop design, the increase of the coherent integration time means that the product  $B_L T_I$  will increase as well, and that the usual continuous-update approximation (Ward 1996) will become irrelevant. This was shown by Stephens and Thomas (1995). However, as already mentioned in the PLL study in Chapter 4, Stephens and Thomas (1995) also provide new sets of loop parameters to take this effect into account, and have a tracking loop that will still produce the expected tracking error variance.

However, several other problems are inherent to longer coherent integration times. The first is that the longer the integration, the more likely it is that the signal conditions will change during that period. The change in the phase/frequency error could be problematic for the correlation values since, as already seen in Section 2.1.3, it implies that the correlation computation will not be optimal. A long coherent integration means that the bandwidth of the I&D filter will decrease proportionally. Consequently, it means that the frequency uncertainty in the carrier wipe-off process should significantly decrease as well. Assuming a constant frequency error during coherent integration, Figure 5.22 shows the resulting in-phase prompt correlation value, normalized by the ideal value the in-phase prompt correlation would have without any frequency error, as a function of the

integration time. As already represented in Equations (2.12) and (2.13), the equivalent

correlation loss can be modeled by  $\frac{\sin(\pi\Delta f T_I)}{\pi\Delta f T_I}$ .



**Figure 5.22 – Normalized In-Phase Correlation Value For Different Constant Frequency Errors**

It can be expected that, under actual conditions, the frequency error will vary during the integration time, which might further degrade the correlation values. The frequency error included in the carrier-aiding received by the DLL from the PLL was given in Section 4.5 considering only Gaussian noise. It has been observed to be dependent upon the PLL loop parameters as well as the SNR. The lower the signal power is, the lower the frequency estimation accuracy will be. This means that, for a standalone receiver, the coherent integration time should be set according to the minimum  $C/N_0$  the receiver can (or wants to) track, or according to the estimated instantaneous  $C/N_0$ .

In order to limit the carrier wipe-off error, a common method consists of using a shorter coherent integration on the PLL than on the DLL. By doing so, the receiver updates the phase/frequency estimation more often to better model the Doppler variations. The correlation values used by the DLL are then obtained through superior modeling of the signal Doppler during the integration. The only limitation in this case is the sensitivity of the PLL. In this respect, it has been shown that the presence of the Galileo L1F pilot channel yielded more accurate phase measurements, as well as a higher PLL sensitivity than that obtained with the current GPS C/A. For very low SNR values, the potentially insufficient sensitivity of the PLL could ideally be compensated by the use of an FLL or external estimators, such as inertial measuring units.

Another problem that must be faced when trying to use long coherent integrations is the change in the signal power during the integration time. This would degrade the discriminator behaviour since the normalization might not be relevant anymore. This is especially true during periods of high dynamics or under harsh environments such as indoors or urban canyons. However, this will not be considered directly herein. It will still be addressed though since, in such difficult environments, shorter integration times (also studied herein) would likely be used.

Finally, it has been decided not to limit the study of the DLL tracking sensitivity to any particular coherent integration time duration. Indeed, it is still interesting to know the theoretical limit of DLL sensitivity, assuming that, by external means, it is possible to perfectly estimate the signal Doppler. These values could be used in research laboratories (Watson 2005) or, possibly in future, with receivers incorporating very high quality

sensors. In such a design framework, the normalization is assumed to be perfect, which means that there is no  $C/N_0$  hard threshold, and that the tracking sensitivity will be decided only by the theoretical code tracking error under Gaussian noise. Note that it is possible that for very low SNR levels certain effects considered negligible, such as the I&D filtering effect, could become problematic. This is not considered herein.

The DP tracking error can be modeled, examining Equations (5.33) and (5.35), as:

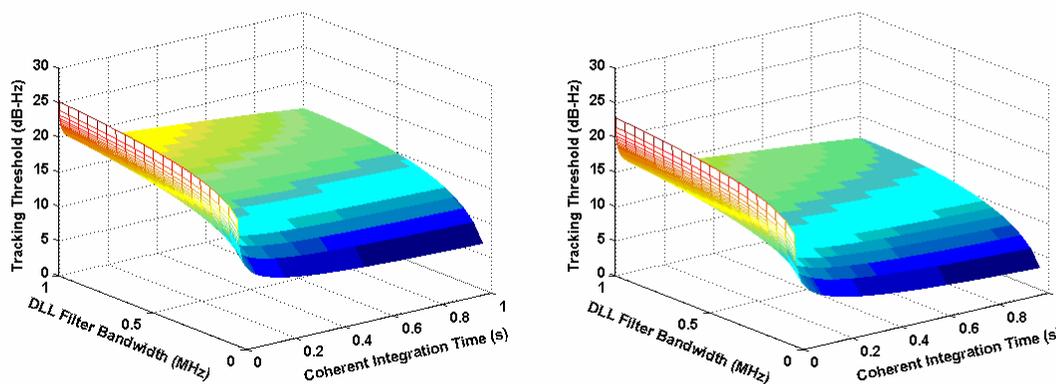
$$\sigma_{\varepsilon_\tau, DP}^2 = \frac{Y}{C} \left( 1 + \frac{1}{Z \frac{C}{N_0}} \right) \quad (5.42)$$

The resulting tracking threshold, using Equation (5.41), is given by:

$$\left( \frac{C}{N_0} \right)_{Th} = \frac{18Y \left( 1 + \sqrt{1 + \frac{\delta^2}{9YZ}} \right)}{\delta^2} \quad (5.43)$$

The resulting tracking thresholds (assuming an infinite front-end filter) are given in Figure 5.23 as a function of the coherent integration time and the DLL loop bandwidth. As expected, the tracking threshold is decreasing with an increasing integration time and decreasing DLL loop bandwidth. For very long integration durations, extremely low  $C/N_0$  values can be tracked (below 5 dB-Hz for integration times of 0.5 to 1 second for the sBOC(1,1) modulation!). For the same values of DLL loop bandwidth and coherent integration time, the sBOC(1,1) signal has a tracking threshold between 2 and 3 dB lower than the BPSK(1) signal, as predicted in Section 5.2.1.2. However, in reality, without external aiding, the GPS C/A code cannot have a coherent integration longer than 20 ms.

Consequently, the actual potential gain of the Galileo L1F signal using its pilot channel is far more important. This is a very important advantage for Galileo L1F over the current GPS C/A signal, since the gain can be of tens of dBs. However, it is still possible to use non-coherent integrations, as discussed in Section 2.2.1, on the GPS signals; yet, this would still be subject to important squaring losses when the SNR is very low. It should be kept in mind, however, that the use of non-coherent summations allows limiting the impact of user acceleration and oscillator frequency drift.



**Figure 5.23 – Tracking Threshold as a Function of the DLL Loop Bandwidth and the Coherent Integration Time for a BPSK(1) (Left) and sBOC(1,1) (Right) Using a DP Discriminator with a Correlator Spacing of 0.2 Chips**

In practice, it has been seen that perfect normalization should not be used (even assuming an ideal frequency estimation during the integration time), and that tracking can be properly done only if the discriminator output does not underestimate the input error for robustness purpose, as discussed in Section 5.2.1.2. It is then very instructive to assess the mean discriminator output shape in order to set a hard  $C/N_0$  threshold imposed by the normalization. This has already been done for coherent integration times of 4 and 20 ms in Table 5.1. Table 5.2 shows, for the Galileo L1F and GPS C/A signals, hard  $C/N_0$  thresholds for up to 1 second coherent integration times and for a DP discriminator

normalized with  $N_3$  using a 0.2 chips correlator spacing and a 3 MHz one-sided front-end filter. It can be seen that, the longer the integration, the lower is the hard threshold. These values are significantly higher than those shown in Figure 5.23 for limited integration times, but seem to approach or go lower than the values obtained with an ideal normalization for long coherent integration times. This means that normalized discriminators would still allow attainment of extremely low values of  $C/N_0$  - in fact, down to approximately 5 to 10 dB-Hz, which is more than 5 dBs lower than what current GPS high-sensitivity receivers reach (MacGougan 2003). However, when compared to current AGPS receivers, which also use long coherent integrations, this threshold might appear unrealistically low. There are two main reasons for that: (1) the frequency error (user dynamics and oscillator frequency drift) has been ignored herein; and (2) the formula given for the tracking accuracy in Gaussian noise tends to overestimate the real performance of the tracking loop for low SNR values.

It appears that the BPSK(1) signal has very similar characteristics to those of the sBOC(1,1) signal, in terms of sensitivity, using the discriminators studied. However this is true only under the assumption that the BPSK(1) is without navigation data, which is not the case with the GPS C/A. It is also interesting to see that, even if the Galileo L1F and GPS C/A code have approximately the same sensitivity due to the limitations inherent to the discriminator normalization, the Galileo L1F would yield more accurate measurements.

**Table 5.2 – Galileo L1F C/N<sub>0</sub> Threshold for Different Coherent Integration Times and using a DP Discriminator with a Correlator Spacing of 0.2 Chips**

	C/N <sub>0</sub> Threshold (dB-Hz)					
	4 ms	20 ms	50 ms	100 ms	500 ms	1 s
<b>sBOC(1,1)</b>	<b>28.5</b>	<b>21.5</b>	<b>17.5</b>	<b>14.5</b>	<b>7.5</b>	<b>4.5</b>
<b>BPSK(1)</b>	<b>28</b>	<b>21</b>	<b>17</b>	<b>14</b>	<b>7</b>	<b>4</b>

It must also be remembered that what prevents realization of very long coherent integrations is the Doppler error during the DLL coherent integration. Indeed, as foreseen in Figure 5.22, for coherent integration times greater than 500 ms, the Doppler would have to be estimated within less than 0.1 Hz, which is very unlikely outside of laboratory conditions (Watson 2005), or for signals with very high SNR values (when long coherent integration times are not required). Since the required Doppler estimation depends only upon the length of the coherent integration time, what would seem more relevant would be to use a coherent integration as long as possible, followed by several non-coherent summations. By doing so, the non-coherent integration operation should not significantly suffer from squaring losses because the (still) long coherent integration should have raised the post-correlation SNR to a sufficiently high level. This would lead to a very similar performance as AGPS receivers that have external access to the navigation bits, but without the need for the aiding architecture.

#### **5.4 Galileo L1F Multipath Mitigation Technique**

As established in Section 5.2, the main sources of error for the DLL are multipath and noise. It has been shown that the use of the Galileo L1F pilot channel can produce significant improvement in terms of tracking sensitivity, due to its modulation and since

the coherent integration time is not limited. It has also been discussed that long coherent integrations, under normal conditions, would not be a real threat since it is possible to have frequent Doppler estimation through a shorter PLL correlation duration during the DLL correlation process. Consequently, since the data channel can realize only short correlations, it is likely that it will not be of great help in increasing the mitigation of Gaussian noise. However, if it is considered to be the main disturbance of the DLL, a data/channel implementation, based on the same algorithm as the one proposed through Equation (4.20) for the PLL could still be used (Hegarty 1999). Yet, it is more likely that the main source of error to be mitigated for Galileo L1F is multipath, as discussed in Section 5.2.2.

Several techniques have been specifically developed to mitigate multipath-induced errors in GPS C/A code tracking. However, for most of the time, they are based on the blanking of part of the incoming signal, and thus suffer from a significant reduction of the SNR. Consequently, they undergo high correlation noise that degrades their performance, as it will be discussed in the next section. However, the existence of a pilot channel, with possible long coherent integrations and thus potential higher correlation gain, could solve that problem.

#### **5.4.1 High Resolution Correlator (HRC)**

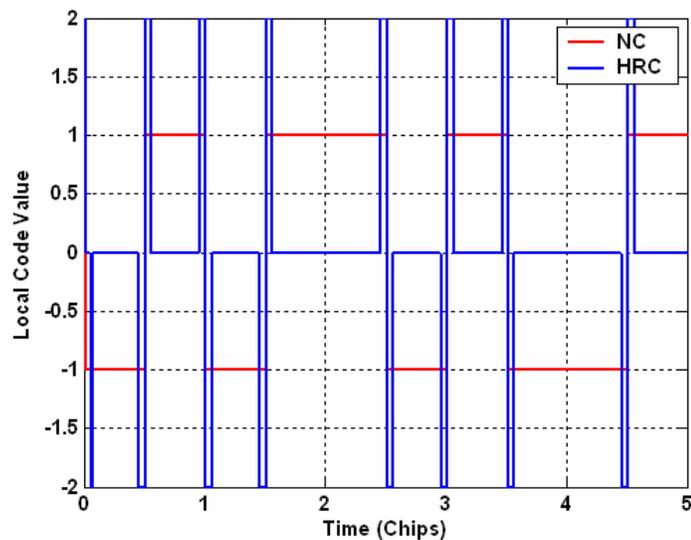
The High Resolution Correlator (HRC) is one of the main versions of code tracking that approximates tracking using a locally generated second derivative of the spreading code described by Weill (1997). This tracking technique is described in detail by McGraw &

Braasch (1999). It uses a locally generated code that is different from the usual local replica. It is shown in Figure 5.24 and can be modeled as:

$$c_{HRC}(t) = 2 \times c(t) - c(t + \delta_{HRC}) - c(t - \delta_{HRC}) \quad (5.44)$$

where  $\delta_{HRC}$  is a parameter defining the HRC local replica.

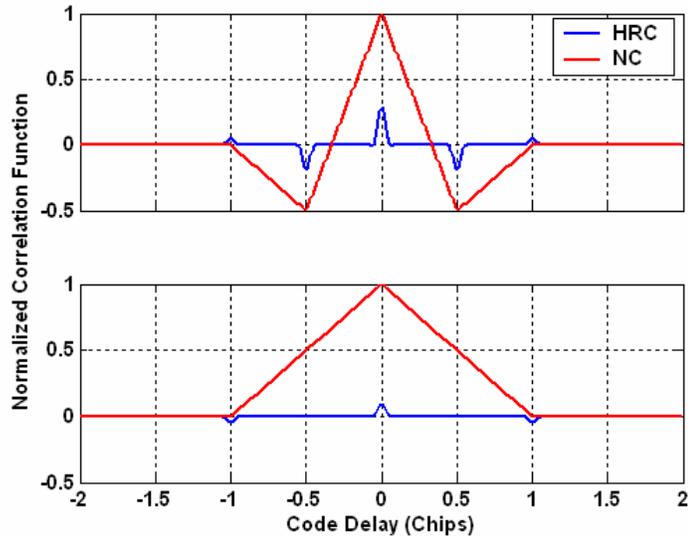
As seen in Figure 5.24, this method consists of blanking a significant part of the incoming code. This will have a great impact on the mitigation of multipath, as they will be blanked if their delay is greater than  $2\delta_{HRC}$ . It is then understood that the lower  $\delta_{HRC}$  is, the better the mitigation of multipath will be.



**Figure 5.24 – Example of Local Code Replica for sBOC(1,1) Tracking Using Traditional and HRC Tracking Techniques**

However, this blanking also seriously degrades the correlator output, since it zeroes most of the incoming signal. As an example, for a BPSK(1) signal using the HRC method with  $\delta_{HRC} = 0.05$  chips, the equivalent degradation in  $C/N_0$  is 13 dB, as described in McGraw

& Braasch (1999) and shown in Figure 5.25 for the BPSK(1) and sBOC(1,1). This figure shows the correlation function between the incoming signal and the local code using the HRC and the traditional local code replicas.



**Figure 5.25 – Normalized HRC ( $\delta_{HRC} = 0.05$  Chips) and Traditional Correlation Function for an sBOC(1,1) (Top) and a BPSK(1) (Bottom) Signal**

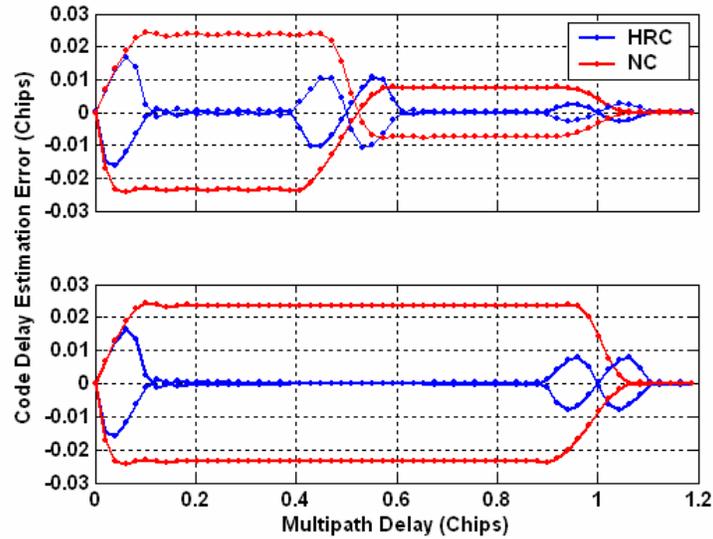
The HRC discriminator proposed by McGraw & Braasch (1999) is:

$$D_{HRC} = \frac{I_{E-L}^{HRC} I_P + Q_{E-L}^{HRC} Q_P}{I_P^2 + Q_P^2} \quad (5.45)$$

The correlator spacing is chosen equal to  $\delta = 2\delta_{HRC}$ . This discriminator  $D_{HRC}$  has the advantage of using the correlator prompt values obtained by the traditional correlation of the incoming signal with the actual spreading code, and not the HRC correlator prompt values. This results in tracking performance that are only slightly degraded, as compared to traditional tracking, instead of suffering directly from the 13 dB SNR loss inherent in

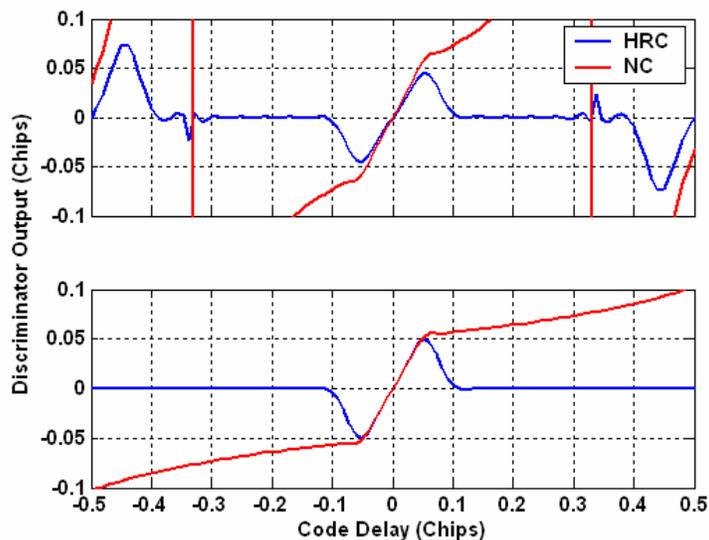
HRC correlation (for a BPSK(1), the code delay error variance using  $D_{HRC}$  is only 3 dB higher than that obtained with the use of a traditional tracking technique with the same correlator spacing, as explained by McGraw & Braasch (1999)).

The resistance to multipath offered by the HRC discriminator is shown in Figure 5.26 along with the resistance of the traditional tracking technique for BPSK(1) and sBOC(1,1) modulations using a 12 MHz one-sided front-end filter. It should be noted that, because very narrow HRC correlator spacing values are desired to efficiently remove multipath, large front-end filter bandwidths are also needed, requiring a higher sampling rate, and thus higher receiver power consumption. Hence, this method is often reserved for high-end receivers. Figure 5.26 shows that the improvement realized through the HRC technique is significant for medium-scale multipath delays. It appears to be less optimal for sBOC(1,1) signals due to the two lobes around 0.5 chips, although constituting a drastic improvement over the traditional sBOC(1,1) tracking technique. This is mainly due to the fact that the HRC technique was designed for the GPS C/A signal structure, and not for a BOC signal. However, multipath mitigation techniques close to HRC and specific to sBOC(1,1) signals, removing the two side-lobes at 0.5 chips, have already been found (e.g. Nunes *et al.* 2004; Garin 2005). Since the objective of this section is to show a tracking concept to efficiently remove multipath-induced tracking errors, the use of such state-of-the-art techniques was not necessary; however, it would be easy to replace the HRC method by these BOC-specific tracking techniques without impairing the method presented herein.



**Figure 5.26 – Traditional and HRC Multipath Envelopes for sBOC(1,1) (Top) and BPSK(1) (Bottom) Signals Using a 12 MHz One-Sided Front-End Filter and a 0.1 Chips Correlator Spacing**

Aside from the problem of lower post-correlation SNR as compared to a traditional tracking technique, the second main drawback for the HRC discriminator is its limited tracking domain, as shown in Figure 5.27. The shape of its discriminator output is not well suited to tracking in degraded environments (e.g. under high dynamics or high noise level) as it fades away when the code delay error reaches  $\delta_{HRC}$  chips. This characteristic departs from traditional tracking that possesses a vertical asymptote at its stability domain boundary, as seen in Section 5.1.3. This is one of the reasons why it might not be suited for low levels of  $C/N_0$  or, at least, for tracking without carrier aiding. In the method presented herein, it is nevertheless beneficial with respect to other derivations of the HRC discriminator (using the degraded ‘HRC’ prompt correlator) that would endure a much higher level of noise. The reason for this will be explained in the next section.



**Figure 5.27 – HRC and Traditional Discriminators Output Using a 12 MHz One-Sided Front-End Filter and a 0.1 Chips Correlator Spacing**

It has been seen in Section 5.1.3 that the traditional tracking with narrow correlator spacing has a more favourable discriminator output shape for an sBOC(1,1) modulation than for a BPSK(1) modulation since it exhibits a discriminator response that will not underestimate the input error when the latter is greater than the discriminator linear tracking domain. This is confirmed by Figure 5.27, and it means that traditional tracking using a narrow correlator spacing should be more robust for Galileo L1F than for GPS C/A code in degraded environments.

It is also important to notice that the HRC method directly applied to the sBOC(1,1) signal has a false lock point situated at  $\pm 0.5$  chips, as is already the case for traditional sBOC(1,1) tracking.

### 5.4.2 Multipath Mitigation Technique Implementation

It can be foreseen that the use of the two code tracking techniques introduced are highly complementary, since traditional tracking (although susceptible to multipath) offers robustness while the HRC provides an excellent multipath mitigation, although suffering from a lack of robustness. When trying to couple these two methods, however, it seems hard to use a discriminator combination such as that used for the data/pilot PLL architecture presented in Section 4.6. Indeed, in the DLL case, the tracking states are different if multipath-induced effects are to be cancelled and can be separated into two cases:

- Tracking using the HRC technique is desired (high multipath case), in which case it is not appropriate to use the traditional technique since it would feed the combined discriminator with its multipath-induced error,
- Traditional tracking is desired (i.e., where robustness is critical), and the combined discriminator should not integrate the potentially unreliable HRC discriminator output.

Consequently, it seems that a combined discriminator implementation would not be relevant herein, but a technique based on a ‘either/or’ would be preferable. Thus, in such a case, since no combinations are used, there is no need to use both channels. In fact it may even be preferable to use the pilot channel only for both tracking techniques since it offers longer possible integration times that would contribute to improving overall measurement accuracy. However, if one prefers to mitigate the measurement noise, provided that the presence of multipath is not significant, it would be preferable, as

already mentioned, to implement a data/pilot combination technique such as that presented for the PLL through Equation (4.19), using a traditional code delay discriminator on both data and pilot channels.

Since the coherent integration on the pilot channel is potentially unlimited, and given that the HRC discriminator will be very noisy due to HRC blanking, it is interesting to note that it must be possible, through use of only the HRC discriminator and long integration times, to ensure reliable tracking. However, the limited HRC discriminator stability domain is a problem, since any disturbance - such as sudden very high noise or short loss of lock - is a threat that could cause the DLL to drift away from the HRC linear tracking domain; consequently, it would be very risky to rely only on the HRC implementation, especially since very low  $\delta_{HRC}$  values are desired.

To reduce the risk of prematurely losing lock on the pilot channel, it is then possible to employ traditional tracking. Moreover, as seen in Equation (5.45), the HRC requires correlation values that could already correspond to the Early and Late values required by the traditional tracking technique. The Early and Late HRC values require the generation of a local code replica with an offset of  $-2\delta_{HRC}$ ,  $-\delta_{HRC}$ ,  $0$ ,  $\delta_{HRC}$ , and  $2\delta_{HRC}$ . Thus, it is possible to implement the traditional tracking technique without extra correlators by using a correlator spacing of  $2\delta_{HRC}$  or  $4\delta_{HRC}$ . Furthermore, as previously discussed, using a medium to narrow ( $> 0.1$  chips) correlator spacing is not a significant problem for tracking robustness when using a sBOC(1,1) signal. Still, if the

choice of a very narrow correlator spacing were to pose a problem in terms of robustness, it would then still be possible to use extra correlators to increase the correlator spacing.

Owing to its robustness, traditional tracking on the pilot channel can then be used as a ‘guardian’ to ensure that the code tracking error remains within the tracking domain of the HRC discriminator. To rigorously define the test function to decide on which tracking to rely on, two cases have to be defined:

- Multipath is present, and the HRC technique is tracking without error, while the traditional tracking error oscillates between the boundaries of its multipath error envelope, and
- The HRC technique is beyond its linear tracking region, and traditional tracking is needed for robustness.

Since robustness is essential for the overall code tracking, it would be preferable to use traditional tracking with an appropriately wide correlator spacing. As mentioned above, use of the HRC method gives access to 2 inherent correlator spacing values for the traditional methods:  $2\delta_{HRC}$  or  $4\delta_{HRC}$ . Hence, it might seem preferable to use  $4\delta_{HRC}$ . Once this value decided, it is important to know the magnitude of the multipath-induced code tracking error that could occur in the case of traditional tracking. It has been seen in Figure 5.20 that this magnitude for a 0.2 chips correlator spacing would not be greater than 0.08 chips, assuming multipath with an amplitude of half of the direct signal (it is extremely unlikely to receive a multipath with a higher amplitude than this). Consequently, it appears that the use of a test based on the value of the traditional

tracking technique's discriminator output is clearly indicated: if tracking is correctly achieved by the HRC technique, the traditional method should not have a discriminator output greater than 0.08 chips. On the other hand, if the output of that discriminator is greater than a certain threshold  $\mu$ , it indicates that the HRC technique probably lost lock, suggesting replacement by the traditional method. The tracking algorithm would then be:

$$\begin{aligned}
 D_{comb} &= D_{Trad} && \text{if} && |D_{Trad}| > \mu \\
 D_{comb} &= D_{HRC} && \text{else} && 
 \end{aligned} \tag{5.46}$$

Such an approach will bring robustness to the implementation, since it should be as robust as the traditional technique. Strong multipath mitigation can then be achieved. The use of long coherent integration should allow a fairly reliable test to decide between the two discriminator outputs shown in Equation (5.46). It should be noted that it is not necessary to have the same integration times on both tracking techniques as long as they have a common divider. However, if a different integration time is chosen for each channel, the algorithm will have to be modified slightly.

### 5.4.3 Test of Multipath Mitigation Capability

This series of tests was designed to assess the ability of the proposed technique to mitigate multipath. Two scenarios were designed for this purpose. The first test simulates a static receiver that receives a strong specular multipath with slow phase change. This should magnify the impact of the multipath on tracking while embodying an approach corresponding to the multipath mitigation capacity of the new technique. Several types of multipath were simulated. One originates in a large flat obstacle situated 50 metres from

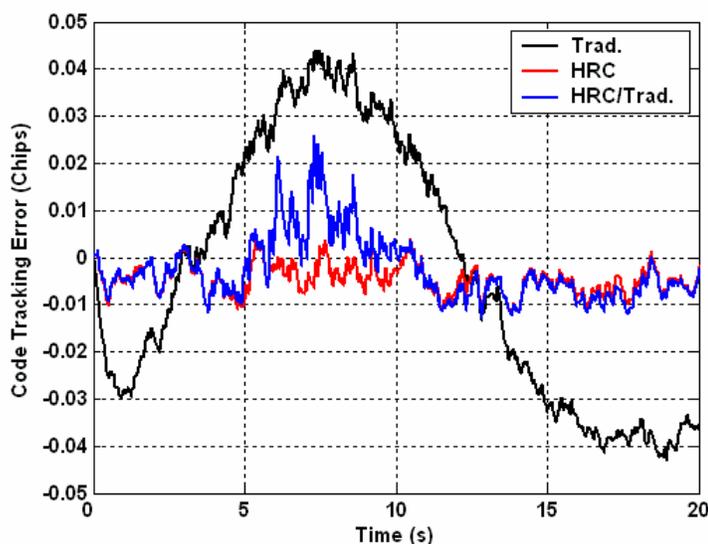
the antenna, and which reflects the incoming signal with an amplitude of half the direct signal; two multipaths consist of ground reflections with amplitudes of one-third of the incoming signal (coming from the satellite and coming from the obstacle). Ten ‘scattered’ multipaths having a power at least 16 dB lower than the direct signal are simulated on the same basis as explained in Section 3.2.6.1. A 37 dB-Hz level of  $C/N_0$  was simulated in this test. The signal then goes through a 6 MHz single-sided bandwidth filter and a 1-bit quantization, further reducing the SNR (by approximately 3 dB). Although this bandwidth is not optimal for use of very narrow correlator spacing values (beneficial for the HRC technique), it nevertheless allows an improvement that should be apparent in the results; it also saves a substantial amount of computation time as it allows a lower sampling frequency. A receiver oscillator phase jitter was simulated using the model presented in Section 3.3.2. The oscillator simulated is a TCXO. The second test has exactly the same configuration, except that the receiver describes a 50-metre radius circular trajectory just beside the same obstacle at a speed of 100 km/h. This allows the simulations of high dynamics conditions.

At the receiver tracking loop level, the integration time was set to 28 ms for both the HRC and traditional tracking techniques.  $\delta_{HRC}$  was set to 0.05 chips and the traditional sBOC(1,1) tracking was designed to use a 0.2 chip correlator spacing. The DLL loop bandwidth was set to 1 Hz.

The following three methods were compared: (1) HRC tracking only on the pilot channel; (2) the traditional tracking technique only on the pilot channel (with a correlator spacing

of only  $2\delta_{HRC}$  chips to test the case of narrow correlator spacing); and (3) the combined technique described above with  $\mu = 0.1$  chips.

Figure 5.28 shows the code delay residual estimation result in the static case. The traditional sBOC(1,1) tracking technique only shows a typical behaviour when the multipath delay is slowly changing. The code delay has maximum and minimum errors reaching slightly less than 0.04 chips which is typical for the correlator spacing and the type of filter used with the kind of multipath generated (delay around 48 metres).

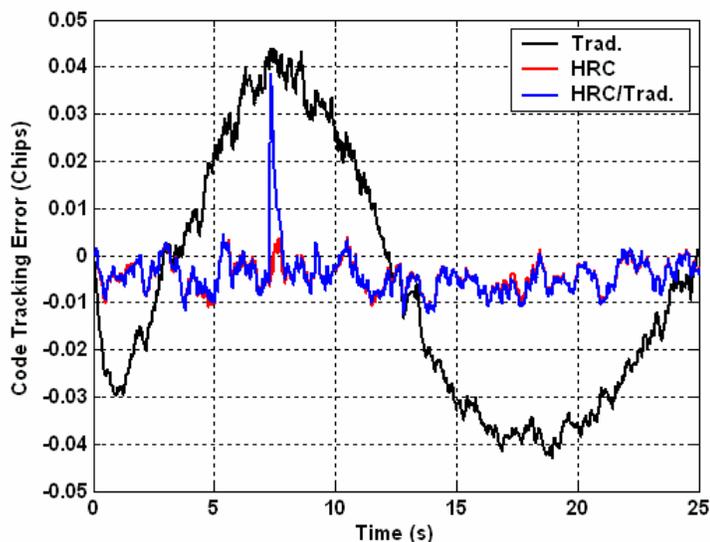


**Figure 5.28 – Code Delay Residual Estimation for the Three Techniques Tested in the Static Case**

As expected, the HRC technique mitigates most of the error due to the presence of multipath. A small variation is visible, in line with the expected error. The proposed combined method exhibits a behaviour that is remarkably similar to the HRC-alone method. However, after approximately 8 seconds, a slight divergence is noticed. This is due to the fact that the error between the traditional discriminator and the HRC

discriminator reaches the threshold (0.1 chips) due to a combination of the noise, a high amplitude (half of the direct signal) multipath, and the addition of the other multipath with less power (including two with amplitude of one third of the direct signal). This combination of multipath creates a code tracking envelope error that reaches 0.15 chips, so significantly higher than the expected 0.08 chips. This result clearly warrants two major remarks at this point. The first concerns the test conditions. It must be understood that the actual occurrence of such high amplitude and stable multipath is unlikely in actual conditions, however, it is possible and it is understood that it is in such cases (high tracking error), that the method should work. Yet, it has to be kept in mind that tracking robustness is probably of first importance. The second remark concerns the test in itself. As in the case of the combined PLL architecture, the use of the direct discriminator output can produce a very noisy measurement, since it is based on values taken before the loop filter. Consequently, it might be more relevant, once again to use a filtered version of these discriminator values. However, signal dynamics does not affect the DLL the same way as the PLL, especially in the case of carrier-aiding. Thus, under such circumstances two methods are realistic and encouraged: the use of a longer coherent integration time (in order to reduce the discriminator output noise) and the use of a low-lag filter, as already realized for the PLL data/pilot architecture. It is also possible to take the output of the DLL filter (in an open loop) since its lag is equal to loop response. Finally, since the Galileo L1F stability region is actually far larger than the specified linear tracking region (0.33 versus 0.05 chips) in the configuration chosen, it is possible to increase the threshold of the test in Equation (5.46). Indeed, even if the HRC loses lock, the loop will eventually reach this threshold, and the traditional robust

tracking will then start. Figure 5.29 shows the result using the same test conditions, but with a 0.15 chips threshold for the test in Equation (5.46) and a 10-epoch averaging filter on the discriminator output.

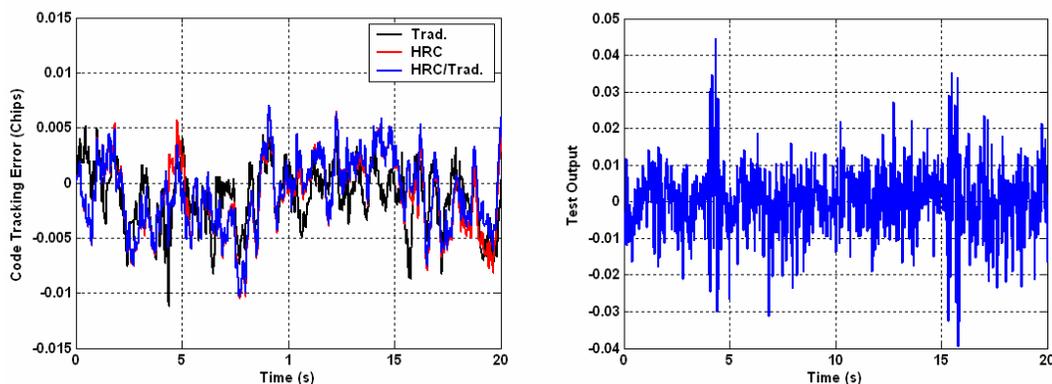


**Figure 5.29 – Code Delay Residual Estimation for the Three Technique Tested for in the Static Case (10 Epochs Average and Threshold of 0.15 Chips for the HRC/Traditional Technique)**

It can be seen that the degradation is reduced, but that there is still a spike corresponding to the moment with the maximum multipath-induced error. The drawback of this technique is that if such an event occurs, it means that the tracking will suddenly jump from one tracking technique to the next. However, this is, once again, the price to pay for obtaining robust measurements, especially if  $\delta_{HRC}$  is chosen very small. Finally, it has to be kept in mind that the event of a loss of lock of the HRC tracking technique, if  $\delta_{HRC}$  is not chosen very small, is minimal especially using long coherent integrations and carrier-aiding. So the use of the traditional tracking is only an insurance that likely will not be often used. Finally, note that the threshold used for the test in Equation (5.46)

has to be chosen according to the correlator spacing used by both the HRC and traditional tracking technique.

Figure 5.30 shows the code delay residual estimation result in the dynamic case. Because the simulated user was moving fairly fast (100 km/h) along a small circular route, the multipath consequently undergoes rapid change. As a result, multipath will have less impact on the final measurement error since their delay varies too quickly for the tracking loop to be able to respond correctly (1 Hz DLL loop bandwidth used).



**Figure 5.30 - Code Delay Residual Estimation for the Three Techniques Tested (Left) and Test Output for the Combined Implementation (Right) in the Dynamic Case**

This is the case when looking at Figure 5.30. Two spikes that correspond to the moment when the user is at the furthest location from the obstacle (and, thus, when the multipath delay is changing at a slower rate) are visible on the test output figure. These two spikes are slightly visible on the DLL tracking error for the traditional tracking approach. It can be seen that they are cancelled by both the HRC-only and combined tracking methods. However, since in general the impact of multipath is minimal, the traditional technique

shows a slightly better behaviour essentially due to superior mitigation of the thermal noise. This is the drawback when using the HRC method only.

The proposed method has shown its effectiveness in mitigating multipath. However, since it relies on only one channel for the tracking, and that the pilot channel uses the HRC method that induces an equivalent  $C/N_0$  degradation, the use of a narrow correlator technique will bring better noise mitigation. Moreover, it might suffer from jumps when extreme multipath-induced error occurs if not well calibrated. These jumps are however not expected to create loss of tracking. However, this is the price to pay to remove the impact of multipath but keep robust tracking, which can be significantly higher than the impact of noise on the final solution. It is important to mention also that this method should be calibrated according to the value of  $\delta_{HRC}$  and the type of traditional tracking desired (for the loop robustness).

A second method, not tested in the frame of this thesis, would consist in systematically relying on a robust traditional Galileo L1F tracking. Coupled with this robust tracking, the HRC method would be implemented in an open loop to obtain multipath-free measurements. This method allows always relying on the robust tracking, while it is still possible to check the consistency of the HRC measurements. Moreover, this implementation would bring robustness to the HRC measurements since it would always rely on the traditional Galileo L1F tracking technique (Garin 2004).

## 5.5 Synthesis

It has been observed that the realisation of the Galileo L1F code tracking loop was a function of many parameters: correlator spacing, front-end filter, discriminator, discriminator normalization, and DLL loop filter bandwidth. The choice of each of them is mainly dependent upon the target user in each application.

Galileo L1F code tracking contains several drawbacks as compared to GPS C/A code tracking. The stability domain of the Galileo L1F discriminators is reduced from  $\pm 1$  chip for a BPSK(1) down to  $\pm 0.33$  chips. This means that Galileo L1F signal will be more susceptible to sudden code delay error jumps (due to a short loss of lock for instance) than GPS C/A code. This is particularly true if no carrier aiding is used. However, this problem does not really concern loss of lock due to high dynamics, since the Galileo L1F discriminator output has been seen to be more advantageous than that of the GPS C/A for correlator spacing values lower than 0.3 chips.

Yet, the Galileo L1F signal structure also exhibits many advantages as compared to the GPS C/A code signal structure. First of all, it affords, through the use of a very similar tracking technique, an improvement of the code delay tracking accuracy standard deviation of approximately 2 to 3 dBs compared to currently available methods, assuming the same loop settings. Moreover, The sBOC(1,1) modulation also exhibits a more favourable discriminator output than BPSK(1) modulation for narrow correlator spacing values using either the EMLP and DP discriminator. This means that, in case of code estimation errors greater than the linear tracking domain (but within the stability

domain), the response of the loop will be more relevant for the Galileo L1F tracking than for the GPS C/A. All this makes the Galileo L1F signal a very compelling option in terms of tracking accuracy, sensitivity and robustness compared to the GPS C/A signal. This conclusion is enhanced by the possibility of realizing very long coherent integrations to further improve the correlation gain. The resistance to multipath is also significantly improved compared to use of the GPS C/A signal. Indeed, the effects of multipath with a delay greater than 0.5 chips are significantly mitigated using an sBOC(1,1) modulation, as compared to a BPSK(1) modulation.

It has also been seen that Galileo L1F, thanks to its pilot channel, was very well suited to incorporating an efficient multipath tracking scheme - the HRC technique - which usually lacks robustness when only short coherent integrations are possible. Combined with a more robust traditional tracking technique, it has been shown to be very well suited to providing accuracy (through multipath mitigation) and robustness. This design seems particularly suited for high-end receivers at this time since it requires more available power.

## CHAPTER 6

### ASPeCT: A NEW UNAMBIGUOUS CODE TRACKING TECHNIQUE FOR GALILEO L1F

Chapter 5 briefly introduced the possible threat of bias code tracking that is intrinsic to the sBOC(1,1) signal structure when a classical tracking architecture is used. The intention of the present chapter is first to demonstrate and analyze the reality of this problem and its intolerable result on signal acquisition and code delay measurements. Two commonly referenced solutions are then studied for a specific use with Galileo L1F signals, and their advantages and drawbacks are investigated. An innovative tracking technique, dedicated to sBOC(1,1) signals, and referred to as *Autocorrelation Side-Peak Cancellation Technique* (ASPeCT), is then introduced. After a thorough description of its theoretical basis, its complete mitigation of potential biased tracking is described. ASPeCT's ability to undergo minimal degradation in terms of Gaussian noise- and multipath-induced code tracking error compared to the traditional sBOC(1,1) signal tracking is then demonstrated, showing this new tracking technique as a strong candidate for any Galileo L1F receiver. Finally, an adaptation of ASPeCT for purposes of unambiguous acquisition is introduced, and its superior performance with respect to traditional acquisition scheme is demonstrated.

## 6.1 Galileo L1F Ambiguous Acquisition and Tracking

The shape of the sBOC(1,1) discriminator output shown in Chapter 5 helped demonstrate its expected improved code tracking accuracy in comparison to BPSK(1) modulation. However, there remain two serious problems associated with the presence of the secondary peaks of the sBOC(1,1) autocorrelation function within the range of  $\pm 1$  chip that have been briefly mentioned, but not yet discussed: (1) potential incorrect acquisition; and (2) potential biased tracking.

### 6.1.1 Acquisition

The sBOC(1,1) has an autocorrelation function that has secondary peaks with a magnitude of 0.5 relative to its main peak, as shown by Equation (2.30). This constitutes an important threat of acquiring on the secondary peak, since there is a significant amount of signal energy located at the secondary peaks. This will of course have an impact on acquisition performance since unlike the cross-correlation peaks, this relative magnitude will remain constant regardless of the signal's  $C/N_0$  value. In order to visualize the probability of acquisition on the secondary peak, an analysis based upon the acquisition theory described by Bastide *et al.* (2002) has been followed. The traditional acquisition test criterion is given by:

$$T_B = \sum_{k=1}^M \left( I_{P,k}^2 + Q_{P,k}^2 \right) \quad (6.1)$$

The traditional sequential approach tests each possible code delay and expected Doppler values one by one using Equation (6.1). A certain two-dimensional bin size, in order to

limit the code delay and Doppler uncertainty, is set with a width that varies from receiver to receiver. The test  $T_B$  is at its maximum when the pair “code delay - Doppler” is the closest from the true value. Understandably, when looking at the correlator output expression in Equations (2.12) and (2.13), a large uncertainty on both the code delay and Doppler can significantly decrease the post-correlation SNR, and the value of  $T_B$ . Obviously, a lower post-correlation SNR will result in less sensitive acquisition. In order to limit this type of loss, it is appropriate to choose small bin sizes; then again, very small bin sizes induce a larger number of cells to search, and thus a longer mean acquisition time (Holmes 1990). Usually, the width of the code delay bins is chosen to be of the size of the spreading symbol autocorrelation width, while the Doppler bin size is one half of the pre-detection bandwidth (Bastide *et al.* 2002). The associated uncertainty induces a maximum loss of SNR of 2.5 dB due to the maximum code delay error, and 1 dB due to the maximum Doppler error.

It has to be noted that by selecting these parameters, it follows that the code delay search bin size for the Galileo L1F will be 0.165 chips wide, while the corresponding quantity for the GPS C/A signal is 0.5 chips wide. This implies a longer acquisition time, especially considering that Galileo L1F spreading codes are four times longer than the GPS C/A spreading code. This is equivalent to what happens when a high code rate is used in the ranging signal; however, the increased acquisition time is balanced by a more accurate code delay estimate. This implies that the number of code delay search cells is 12 times higher for Galileo L1F than for GPS C/A. Similarly, the necessary 4 ms integration time for Galileo L1F means that the Doppler bin size is 125 Hz, and that it is

4 times smaller than for the GPS C/A signal when only a 1 ms coherent integration time is used. Both cases show that, to achieve a comparable mean acquisition time, Galileo L1F will require 48 times more correlators.

In order to arrive at realistic values, an interfering signal is assumed to cause a cross-correlation peak that must be taken into account in the acquisition process (Bastide *et al.* 2002). This is usually used to construct and anticipate a worst case scenario. It allows calculation of the detection threshold from a false detection rate specification. For a given false alarm probability  $P_{fa}$ , the detection threshold  $Th$  is given by:

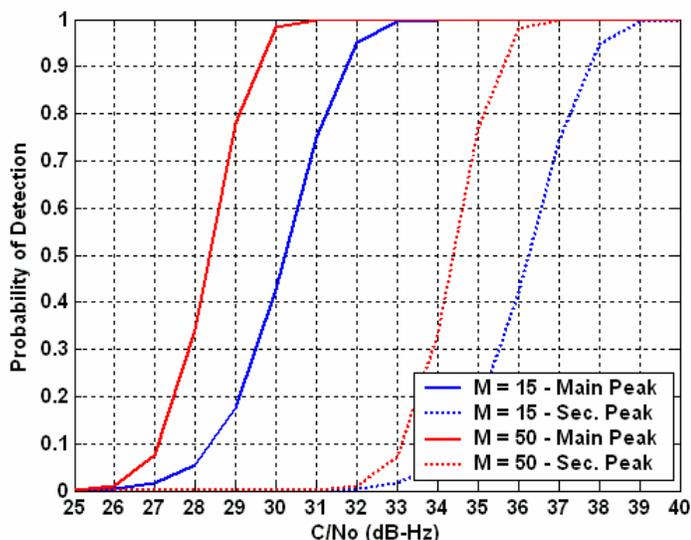
$$P(T|_{NoSignal+1Interf} > Th) = P_{fa} \quad (6.2)$$

Once this threshold has been determined, the probability of detection of the useful signal  $P_d$  is given by:

$$P_d = P(T|_{Signal} > Th) \quad (6.3)$$

This method, valid for signals spread by a standard PRN code, can be fully re-used when a sub-carrier modulates the code because the cross-correlation peaks of the BOC signals have the same magnitude as those of the spreading code. In order to set the acquisition thresholds, a probability of false alarm ( $P_{fa}$ ) of  $10^{-3}$  has been chosen and an interfering signal with a  $C/N_0$  of 50 dB-Hz has been assumed as the worst case. However, it has been shown in Chapter 2 that Galileo L1F spreading codes have not yet been selected. Consequently, the isolation of Galileo L1F spreading sequence autocorrelation over potential cross-correlation peaks has not been established. By comparison to the well-

known GPS C/A and GPS L5 isolation, it was assumed to be equal to 25 dB in the present case. Figure 6.1 shows the probability of detection of the main and secondary peaks of the sBOC(1,1) signals assuming no Doppler or code delay, for a coherent integration time of 4 ms and for 15 and 50 non-coherent summations.



**Figure 6.1 – Probability of Detection of the Main and Secondary Peaks of the sBOC(1,1) Signals for 15 and 50 Non-Coherent Summations and a Coherent Integration Time of 4 ms**

As expected, the probabilities of detection of the secondary peaks are offset from those of the main peak by 6 dB, which is the difference between the correlation power of the two peaks. It can be observed from Figure 6.1 that when the  $C/N_0$  reaches between 35 and 40 dB-Hz, the secondary peaks can be considered as real threats for acquisition due to their non-trivial probability of detection. It must be stressed that the 6 dB difference between the curves corresponding to the main and secondary peaks is not the worst case. Indeed, since the search bins induce an uncertainty into the estimation of the code delay, it is possible that the code delay initially estimated may fall slightly on the side of the main

peak, but directly on the secondary peak, thereby reducing the difference between the two cases. To cancel that possibility, the code delay search bin should be taken as a divider of the distance between the main and secondary peaks (0.167 chips, for instance).

### 6.1.2 Tracking

It can be seen that all of the discriminators studied for the Galileo L1F signal have false lock points located around  $\pm 0.55$  chips, as apparent in the graph of Figure 5.5 for instance. There, it can be observed that the false lock point stability does not appear to be affected by the choice of the discriminator. These false lock points are a direct consequence of the existence of the secondary peaks of the sBOC(1,1) autocorrelation function. Indeed, peaks pointing upward will lead to stable false lock points when using an “Early-Minus-Late”-type of discriminator (while peaks pointing downward lead to unstable lock points).

It is also interesting to note that the false lock points are not situated exactly on the secondary peak location since the slopes forming the secondary peaks have different values. Computation of the exact location of the false lock  $\tau_{FL}$  point can be done by solving:

$$\tilde{R}\left(\tau_{FL} + \frac{\delta}{2}\right) = \tilde{R}\left(\tau_{FL} - \frac{\delta}{2}\right) \quad (6.4)$$

with  $|\tau_{FL}| > 0$  chips.

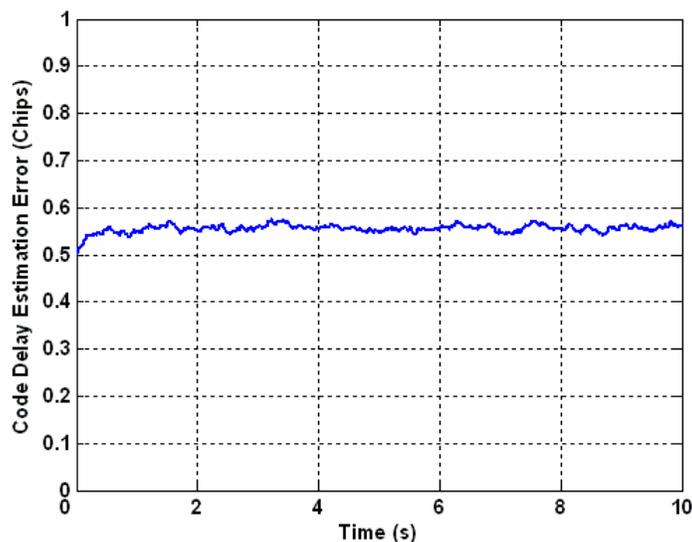
Assuming an infinite front-end filter, this gives:

$$\tau_{FL} = \frac{2 + \delta}{4} \text{ chips} \quad (6.5)$$

Consequently, the wider the correlator spacing, the greater the tendency for the false lock point to approach  $\pm 1$  chip. Since, as shown in Section 5.1.2, the maximum correlator spacing is limited to 0.33 chips for optimal Galileo L1F tracking, the false lock point will be within [0.5; 0.583] chips. This implies that the resulting code measurements would be biased by 146.5 to 170.9 metres. However, this error level is intolerable for a navigation system that purports to deliver maximum accuracy.

Figure 6.2 shows the result of a simple test where the initial code delay error was assumed to be 0.5 chips away from the true code delay (to simulate a false acquisition), the signal had a  $C/N_0$  of 40 dB-Hz. A 4 ms integration time was used, along with a correlator spacing of 0.2 chips. It can be seen that the DLL locks on the false lock point, and maintains lock throughout the test. The stable point seems to be around 0.55 chips, which corresponds to the relation described in Equation (6.5).

False lock can result from different causes, for instance false acquisition, high noise, or short loss of lock with rapid re-acquisition. In any case, it is of major importance to prevent the occurrence of such ambiguous tracking. Two main methods are often referenced to solve this problem of biased code tracking and are presented hereafter.



**Figure 6.2 – Example of False Lock Occurring after False Acquisition on a Secondary Peak**

### 6.1.3 Main Indicated Solutions

Two main solutions to the ambiguity problem have been described in the literature: the *Single Side-Lobe* (Martin *et al.* 2002) and the *Bump and Jump* (Fine & Wilson 1999) techniques. These two methods were suggested for the processing of any BOC family. They are described herein with particular reference to the case of the sBOC(1,1) modulation.

#### 6.1.3.1 Single Side-Lobe Technique

The *Single Side-Lobe* (SSL) technique is based on the consideration of the BOC spectrum as the sum of two BPSK spectrums shifted by  $\pm f_c$ . Indeed, one BOC PSD side-lobe is reminiscent, in theory, of a classical BPSK PSD shifted by  $\pm f_c$ , the only difference

being that a sine square-wave is used instead of a normal sine function. Consequently, when the incoming signal is correlated with the PRN code only (no sub-carrier) shifted by  $f_c$ , it should give a correlation function close to a BPSK(1) correlation function and thus an unambiguous position solution. Following this idea and the notations from Chapter 2, the local replica will be:

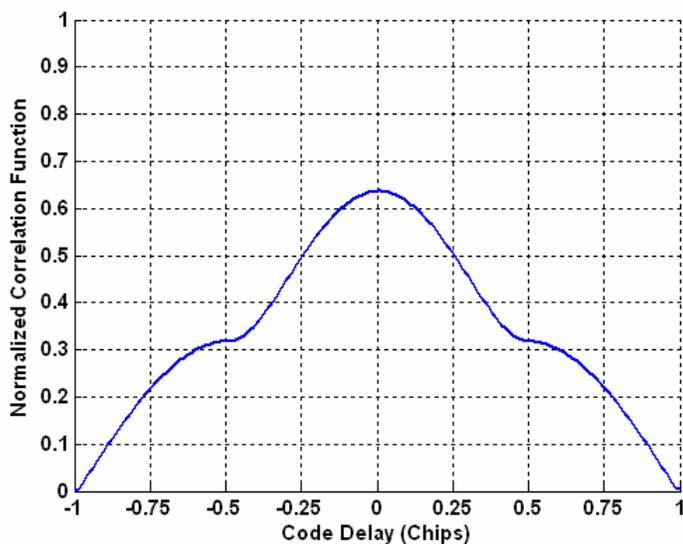
$$LR_{I,SSL}(t) = c(t)\sin(2\pi f_c t) \quad (6.6)$$

$$LR_{Q,SSL}(t) = c(t)\cos(2\pi f_c t) \quad (6.7)$$

The resulting correlation function is given by Winkel (2003) and is shown in Figure 6.3. As expected, the correlation function does not present any side-peaks, meaning that it is fully unambiguous. However, this method suffers from two main drawbacks:

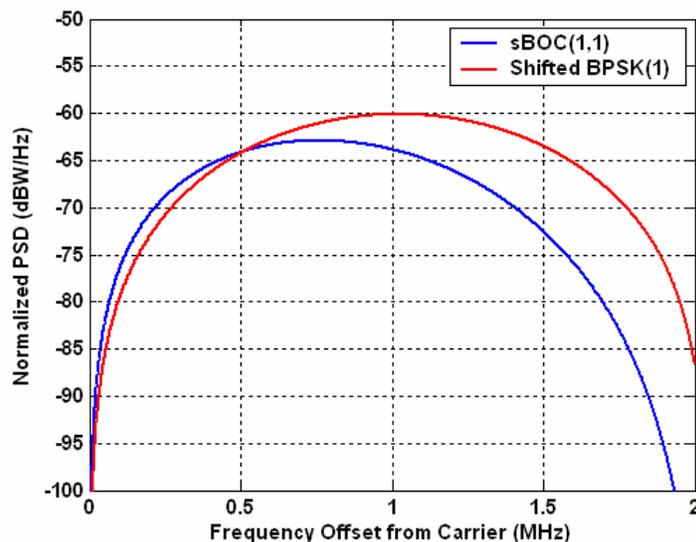
- Since neither of the sBOC(1,1) side-lobes perfectly match a BPSK(1) main lobe, as shown in Figure 6.4, it creates a correlation function that does not have straight slopes on each side, which will clearly impact the possibility of ideal tracking. Moreover, it also induces an associated power loss of approximately 2 dB, as compared to conventional sBOC(1,1) autocorrelation.
- Also, by comparison to conventional sBOC(1,1) tracking, the imitation of BPSK(1) tracking induces weaknesses associated with BPSK(1) tracking (compared to traditional sBOC(1,1) tracking) in terms of Gaussian noise and multipath mitigation.

Consequently, the expected tracking performance using the SSL technique will be poor compared to classical BOC tracking. However, it offers reliable measurements with a wide stability domain (i.e., equivalent to a BPSK(1)).



**Figure 6.3 – Normalized Correlation Function between the Incoming sBOC(1,1) Signal and a PRN Local Replica (no Sub-Carrier) Shifted by  $f_c$**

Because of the power loss experienced during the correlation process, it seems that this method is not viable in terms of tracking compared to what is expected from an sBOC(1,1) signal with respect to accuracy and sensitivity. However, it can be an attractive option as an acquisition tool, since its wide main correlation peak allows incorporation of larger code delay cells, which can be precious to gain in mean acquisition time (see Section 6.1.1).



**Figure 6.4 – Normalized Power Spectrum Densities of sBOC(1,1) and BPSK(1) shifted by  $f_c$**

The SSL method has also been proposed for transition from acquisition to tracking since for an acquisition anywhere on the main peak, tracking with the SSL method should unambiguously converge toward the correct code delay (Martin *et al.* 2002). Once within the sBOC(1,1) traditional discriminator’s tracking region, it is then possible to switch to an alternative and more optimal tracking method.

#### 6.1.3.2 “Bump and Jump” Technique

The “*Bump and Jump*” (BJ) technique is not defined as a tracking technique, but more as a “bias checking” technique. Indeed, its goal is to constantly check that a traditional BOC tracking technique is locked on the correct (main) peak of the BOC autocorrelation function and not on one of the side-peaks. To do so, the BJ technique uses extra-correlators located at the theoretical location of the side-peaks. They are referred to as Very Early (VE) and Very Late (VL) correlators. Using the output of these correlators, a

statistical test is then performed to compare the magnitude of the VE, Prompt and VL correlation values. If the Prompt value is the greatest, it means that tracking is likely done on the main peak. If not, it means that tracking might be biased, and that the DLL should ‘jump’ on the highest peak.

There are several ways of checking the relative magnitude of each side-peak and to implement the jumping process. A counter related to each correlator is usually employed to ensure a certain dependence upon the correlators’ output noise and thus to avoid frequent irrelevant jumps (Fine & Wilson 1999). However, it is not the intention of this thesis to study each of these methods in detail, but simply to give the essence of each technique. Consequently, a probabilistic approach to assessing the reliability of this method will be given based on correlation values using only one coherent integration.

In such a scenario, the following two cases are possible:

- Tracking is unbiased and, consequently, the VE and VL correlators are located on the secondary peaks. This desired configuration means that the sBOC(1,1) signal is tracked optimally as described in Chapter 5, taking advantage of the good performance of the sBOC modulation;
- The DLL is locked on one of the side peaks, and either the VE or the VL correlator will measure the magnitude of the main peak. In this case, it must be kept in mind that the false lock point is located in  $\tau_{FL}$ , as calculated in Equation (6.5). Consequently, if one wants to compute the magnitude of the main autocorrelation peak, the distance between the VE, Prompt and VL correlators must equal  $\tau_{FL}$ , and not the theoretical (0.5 chips) distance between each peak - a

point not discussed in Fine & Wilson (1999). Taking  $\tau_{FL}$  instead of 0.5 chips should also be beneficial to the BJ approach in the case of unbiased tracking since in this case, the correlation value in  $\pm\tau_{FL}$  is lower than the magnitude of the secondary peaks. Note that this has to be adapted when a limited front-end filter bandwidth is used, since it tends to move the location of the sBOC(1,1) autocorrelation side-peaks (see Figure 2.8) and thus the location of the false lock points as well.

The approach taken herein to assess the performance of the BJ technique is based on Monte Carlo simulations. Therefore, it is important to remember that the values obtained from these simulations are statistical, and are aimed only at showing the general type of performance expected from the BJ, and not describing the exact performance. Using the same method as in Equation (5.36) to generate correlated Gaussian noises, it is possible to closely generate the typical VE, Prompt and VL correlation values using Equations (2.14), (2.15) and (2.16). The correlation matrix must also be changed to:

$$C = \begin{pmatrix} R_n(0) & R_n(\tau_{FL}) & R_n(2\tau_{FL}) \\ R_n(\tau_{FL}) & R_n(0) & R_n(\tau_{FL}) \\ R_n(2\tau_{FL}) & R_n(\tau_{FL}) & R_n(0) \end{pmatrix} \quad (6.8)$$

Using a pure theoretical approach, Fine & Wilson (1999) did not consider this facet of noise correlation.

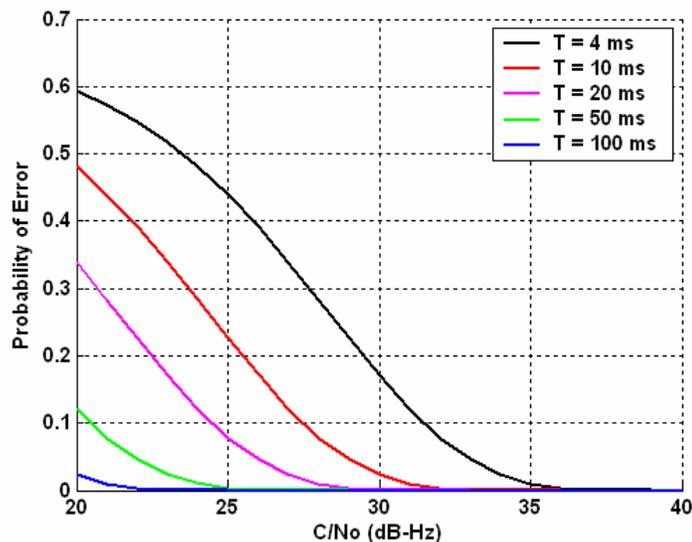
In order to assess the magnitude of each VE, Prompt and VL correlation points, the following power estimate was used:

$$M_X = I_X^2 + Q_X^2 \quad (6.9)$$

On this basis, it is interesting to test the two tracking cases (unbiased and biased). In the first case, assuming that tracking is correct, the figure of merit is the probability that one of the two side-peaks is higher than the main peak. It is expressed by:

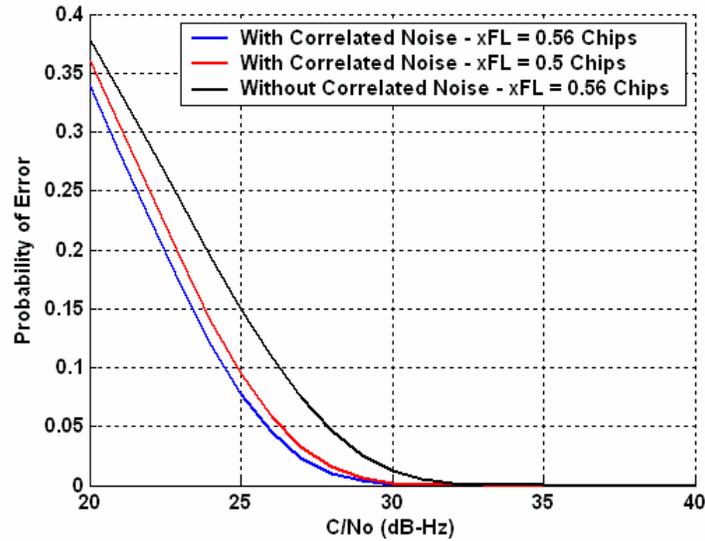
$$p((M_{VE} \geq M_P) \cup (M_{VL} \geq M_P)) \quad (6.10)$$

This figure of merit is shown in Figure 6.5. For typical  $C/N_0$  values ( $> 30$  dB-Hz), it seems that the BJ technique is satisfactory under Gaussian noise; yet, it also shows that the test must be carefully calibrated in order to minimize any possibility of error: a short coherent integration would clearly lead to a very high probability of error, and it is likely that the tracking loop would randomly jump on the side-peaks. On the other hand, a long coherent integration seems feasible and produces a reliable result. The problem encountered with long integration times is that the conditions can be variable during the correlation process, and this can affect the test significantly. Assuming a good external Doppler estimation, integration times of 100 ms should not be problematic. However, as seen in Section 5.3 for a 100 ms integration time, the Doppler accuracy should be better than 2.5 Hz in order to secure a correlation power loss lower than 1 dB. If only carrier-aiding is available, maintaining this may seem problematic, especially for low  $C/N_0$ . Consequently, this should induce a lower performance of the BJ technique as compared to the theoretical situation depicted in Figure 6.5. Indeed, as seen in Equation (2.12) and (2.13), a frequency error could be assimilated, from a correlation point-of-view, to a decrease in the signal  $C/N_0$  value.



**Figure 6.5 – Probability that the Magnitude of one of the Secondary Peaks is Higher than the Magnitude of the Main Peak, Assuming Unbiased Tracking**

The impact of the assumption of correlated noise (versus assuming uncorrelated noise as in Fine & Wilson (1999)) is shown in Figure 6.6. If no correlation between the correlators' output noise is assumed, a large underestimation of the performance of the technique is induced. This means that the results given by Fine & Wilson (1999) demonstrate a lower bound of the expected performance of their version of the BJ technique. Figure 6.6 also shows the impact of the distance between the prompt and VE/VL correlators. As explained previously, it can be seen that using the actual false lock point separation  $\tau_{FL}$ , instead of 0.5 chips improves the statistical test in order to avoid the loop mistakenly jumping on a side-peak.

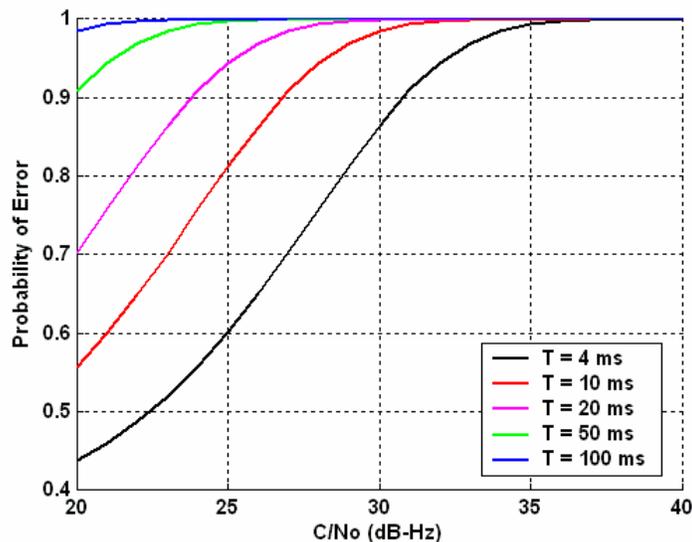


**Figure 6.6 – Probability that the Magnitude of one of the Secondary Peaks is Higher than the Magnitude of the Main Peak, Assuming Unbiased Tracking, for a 20 ms Integration Time, With and Without the Correlated Noise Assumption, and with Different VE – Prompt Separations**

In the second configuration, tracking is assumed to be biased, and so the lock point is assumed to be located at  $\pm \tau_{FL}$ . The figure of merit in this case is the probability that the main peak (at VE or VL) is higher than the magnitude of the prompt correlator output.

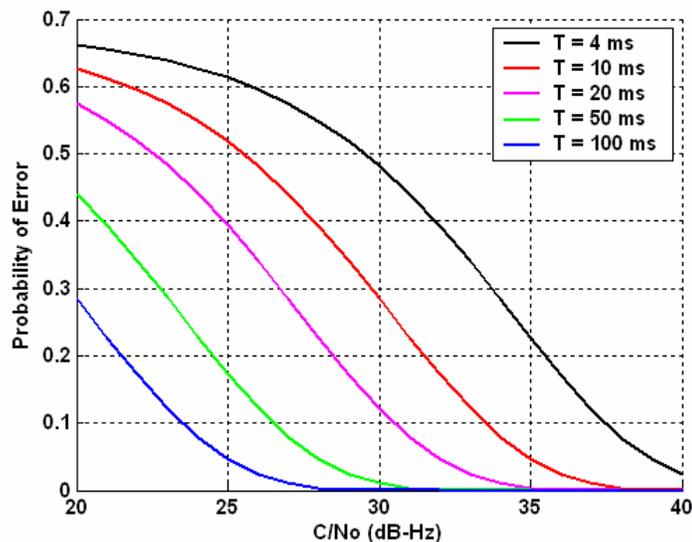
$$P((M_P \geq M_{VE}) | \tau_{FL}) \quad (6.11)$$

The results are shown in Figure 6.7. The conclusions are the same as in the first case; i.e., that only long coherent integrations could satisfy users with high sensitivity receivers.



**Figure 6.7 - Probability that the Magnitude of the Main Peak is Greater than the Magnitude of the Secondary Peak, Assuming Biased Tracking**

The second main problem with the BJ method is the presence of multipath. Strong multipath will greatly change the shape of the autocorrelation function, as discussed in Section 3.1.5. This will have a significant influence on the test, especially if the multipath originates from a large smooth surface (specular multipath). Indeed, in such a case, this would attribute stable characteristics to the multipath that may significantly impact the statistical test, even during long coherent integration times. Figure 6.8 shows an example of such an event with stable multipath with half the amplitude of the direct signal in phase opposition, and with a very short code delay ( $\sim 0$  chips). As compared to Figure 6.5, it can be seen that even for longer coherent integration times, the probability of error is high. Even though multipath tends to be unstable (especially in kinematic conditions) this is an event to consider as part of the reliability analysis of this method.



**Figure 6.8 - Probability that the Magnitude of one of the Secondary Peaks is Higher than the Magnitude of the Main Peak, Assuming Unbiased Tracking and One Stable Multipath in Phase Opposition with an Amplitude Half of the Direct Signal**

Another problem is multipath mixing with the direct signal while the tracking is biased. Indeed, in such an event, the correlation function associated with the multipath will have an amplitude that will be relatively closer to the sBOC(1,1)'s autocorrelation function secondary peak than to the main peak. This means that its effect will influence potential tracking of the secondary peak considerably; furthermore, if the magnitude of the multipath correlation is very high (around the secondary peak magnitude), it could potentially deceive the DLL into tracking it instead of the main peak.

In conclusion, the BJ method seems appropriate for typical conditions when using long integration times. Moreover, specific algorithms, such as the one presented in Fine and Wilson (1999), based on a counter for each correlator output, can improve the overall technique but two significant problems remain:

- If a mistake is made, the longer the integration time, the longer the loop will take to correct it. In some applications, this might be extremely harmful, and
- It is a statistical test, and the chance of occurrence of an error is never zero. Consequently, all tests in the presence of noise may fail.

Consequently, it would be preferable to employ a tracking technique that would be inherently reliable for critical applications, but also for the majority of users since blunders can significantly harm the final position estimate. This is especially true when conditions are poor and the redundancy of measurements precludes effective blunder detection in the navigation algorithm used (MacGougan 2003, Lachapelle *et al.* 2003).

As discussed in the introductory chapter, the two techniques mentioned to remove the sBOC(1,1) tracking ambiguity problem can be used with any of the BOC families. It is known that the various BOC(n,m) families have distinct characteristics (Betz 2002), and trying to find a generic solution might not lead to an optimal solution for each family. Consequently, it was decided herein to seek a solution that would be ‘customized’ for Galileo L1F, since it is the signal of interest of this thesis. However, after the analysis of the main drawbacks and advantages of the SSL and BJ techniques, several criteria were selected in order to develop a tracking solution as optimal as possible. These criteria were given in Section 1.1, and are repeated here for greater emphasis:

- Tracking architecture close to current tracking techniques in order to be easily incorporated in any receiver;
- Total reliability (no false lock possible); and
- Close-to-optimal tracking accuracy under Gaussian noise and multipath.

The research done in this direction resulted in the development of an innovative tracking technique dedicated to sBOC(n,n)-family tracking, and referred to as ASPeCT, that is described in the next section in the specific case of the Galileo L1F signal.

## 6.2 ASPeCT

The essence of ASPeCT is to remove the side-peaks of an sBOC(1,1) autocorrelation function since they are at the origin of the false lock tracking points. To do so, it is interesting to calculate the correlation of the sBOC(1,1) modulated spreading sequence (PRN code  $\times$  sine square-wave sub-carrier) with the PRN code only. This sBOC(1,1)/PRN (that will be referred to as BOC/PRN) correlation function, assuming band-limiting by a rectangular filter is given by:

$$\tilde{R}_{BP}(\tau) = \int_{-\infty}^{+\infty} F_B(f) F_P^*(f) H(f) e^{2i\pi f \tau} df \quad (6.12)$$

where  $F_B$  is the Fourier Transform of the sBOC(1,1) signal, and

$F_P$  is the Fourier Transform of the PRN sequence (that is a BPSK(1) signal).

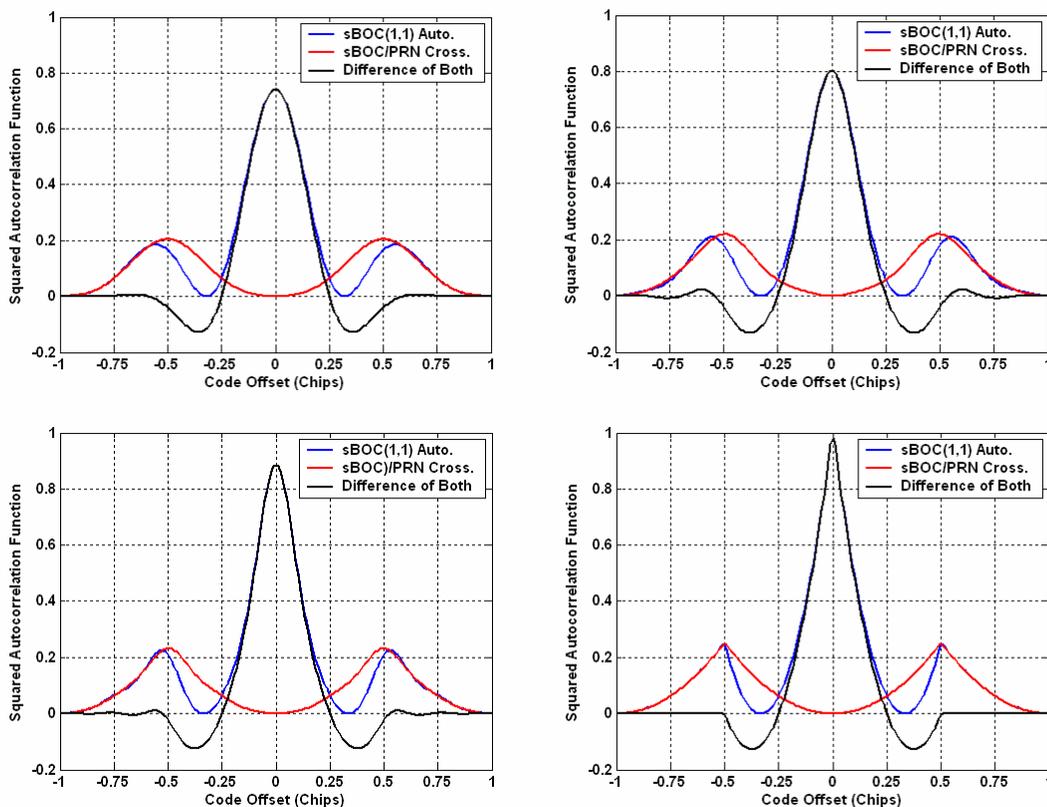
Assuming infinite front-end filtering, the BOC/PRN correlation function equals [Julien *et al.* 2004]:

$$R_{BP}(x) = \frac{1}{2} \left( \text{tri} \left( \frac{x - \frac{1}{2}}{\frac{1}{2}} \right) - \text{tri} \left( \frac{x + \frac{1}{2}}{\frac{1}{2}} \right) \right) \quad \text{for } x \leq 1 \text{ chip} \quad (6.13)$$

Equation (6.13) shows that  $R_{BP}$  consists of two triangles perfectly located on the side-peaks of the sBOC(1,1) autocorrelation function, and having exactly the same magnitude. As a consequence, the idea on which ASPeCT is based is to form a synthesized correlation function by subtracting  $R_{BP}^2$  from  $R_B^2$  to remove the undesired side-peaks. However, it must be realized that the front-end filter limited bandwidth will probably degrade the side-peak cancellation process since, as seen in Figure 2.8, a narrow filter can slightly move the location of the sBOC(1,1) side-peaks. The impact of the front-end filter is depicted in Figure 6.9. It shows how the respective side-peaks of each correlation function move in relation to each other, and how this might affect the expected synthesized correlation function. It can be seen that, for a certain front-end filter bandwidth, the synthesized correlation function can have small remaining side-peaks that may pose a threat.

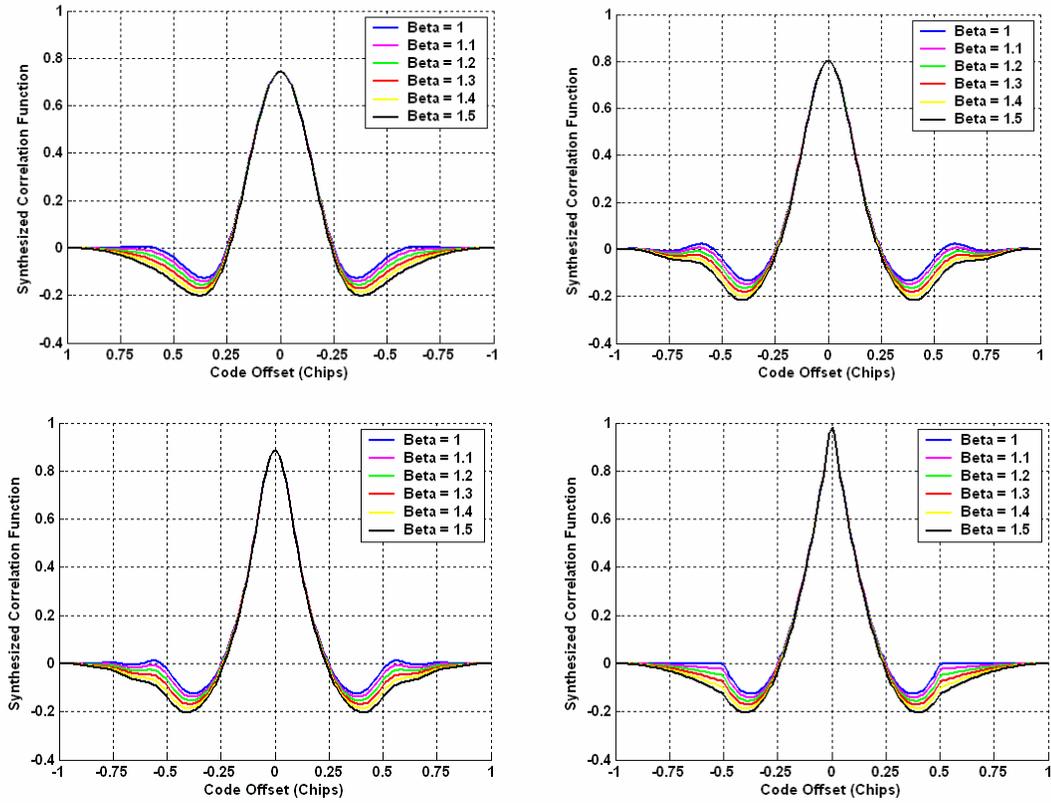
These remaining side-peaks can translate into false lock points. Thus, to ensure fully reliable tracking, this problem must be taken into account. A possible solution is to use a coefficient  $\beta$  in the combination of the two squared correlation functions in order to eliminate any small remaining peaks. This can be modeled as:

$$R_{ASPeCT}(\tau) = \tilde{R}_B^2(\tau) - \beta \tilde{R}_{BP}^2(\tau) \quad (6.14)$$



**Figure 6.9 – sBOC(1,1)/sBOC(1,1), sBOC(1,1)/PRN Squared Normalized Correlation Functions and the Difference of Both Squared Correlation Functions for 2 (Top Left), 3 (Top Right), 5 (Bottom Left), and 20 (Bottom Right) MHz One-Sided Front-End Filter**

From the shape of the correlation functions shown in Figure 6.9, it can be seen that the value of  $\beta$  will probably be dependent upon the front-end filter bandwidth. Figure 6.10 shows ASPECT's synthesized correlation function using different values of  $\beta$  (between 1 and 1.5) using a 2, 3, 5, and 20 MHz one-sided front-end filter bandwidth. It can be seen that the use of the coefficient  $\beta$  helps to remove the potential remaining side-peaks.



**Figure 6.10 - ASPeCT Synthesized Correlation Function For Different Beta Values Using a 2 (Top Left), 3 (Top Right), 5 (Bottom Left), and 20 (Bottom Right) MHz One-Sided Front-End Filter.**

It can be understood that since this synthesized correlation function attenuates the side-peaks, it is an excellent starting point for defining an unambiguous discriminator. Two discriminators are proposed: a modified version of the conventional EMLP discriminator and a modified version of the DP discriminator, both presented in Chapter 5. Their expressions are as follows:

$$D_{EMLP}^{ASPeCT}(\varepsilon_\tau) = \begin{pmatrix} \left[ (I_E^2 - I_L^2) + (Q_E^2 - Q_L^2) \right]_B \\ -\beta \left[ (I_E^2 - I_L^2) + (Q_E^2 - Q_L^2) \right]_{BP} \end{pmatrix} \quad (6.15)$$

and

$$D_{DP}^{ASPeCT}(\varepsilon_\tau) = \begin{pmatrix} \left[ (I_E - I_L)I_P + (Q_E - Q_L)Q_P \right]_B \\ -\beta \left[ (I_E - I_L)I_P + (Q_E - Q_L)Q_P \right]_{BP} \end{pmatrix} \quad (6.16)$$

where the subscript  $B$  refers to the sBOC(1,1) autocorrelation function, and the subscript  $BP$  refers to the BOC/PRN correlation function.

The discriminators' model, assuming no frequency error and no external disturbance, can be written as:

$$D_{EMLP}^{ASPeCT}(\varepsilon_\tau) = \frac{P}{2} \begin{bmatrix} \left( \tilde{R}_B^2\left(\varepsilon_\tau - \frac{d}{2}\right) - \tilde{R}_B^2\left(\varepsilon_\tau + \frac{d}{2}\right) \right) \\ -\beta \left( \tilde{R}_{BP}^2\left(\varepsilon_\tau - \frac{d}{2}\right) - \tilde{R}_{BP}^2\left(\varepsilon_\tau + \frac{d}{2}\right) \right) \end{bmatrix} \quad (6.17)$$

and

$$D_{DP}^{ASPeCT}(\varepsilon_\tau) = \frac{P}{2} \begin{bmatrix} \left( \tilde{R}_B\left(\varepsilon_\tau - \frac{d}{2}\right) - \tilde{R}_B\left(\varepsilon_\tau + \frac{d}{2}\right) \right) \tilde{R}_B(\varepsilon_\tau) \\ -\beta \left( \tilde{R}_{BP}\left(\varepsilon_\tau - \frac{d}{2}\right) - \tilde{R}_{BP}\left(\varepsilon_\tau + \frac{d}{2}\right) \right) \tilde{R}_{BP}(\varepsilon_\tau) \end{bmatrix} \quad (6.18)$$

Assuming an infinite front-end filter, and an absolute code delay error within  $\pm \delta/2$

chips, the discriminators' output can be written as:

$$D_{EMLP}^{ASPeCT}(\varepsilon_\tau) = P(\alpha(2 - \alpha\delta) + \beta\delta)\varepsilon_\tau \quad (6.19)$$

and

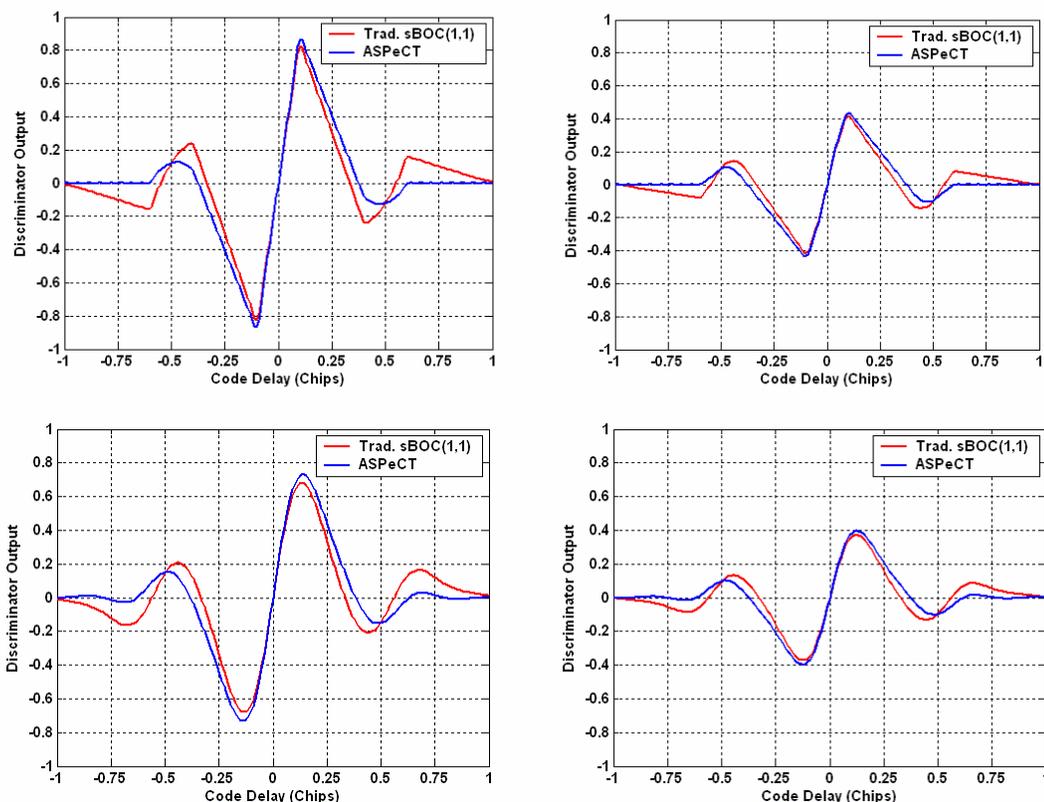
$$D_{DP}^{ASPeCT}(\varepsilon_\tau) = P\left(\alpha(1 - \alpha|\varepsilon_\tau|) + \beta\frac{\delta}{2}\right)\varepsilon_\tau \quad (6.20)$$

It should be noted that the gain produced by each ASPeCT discriminator is slightly greater than the gain of their traditional counterparts. This is important since it positively influences the DLL tracking error variance.

It must also be noted that the proposed discriminators - since they are based upon the combination of two correlation functions - use more complex correlators. The ASPeCT EMLP discriminator, taking into account the need for the prompt correlator for phase tracking, uses five complex correlators. On the other hand, the DP discriminator, since the local 'Early-Minus-Late' replicas can be directly generated, uses only four complex correlators. This is equivalent to traditional DP tracking (two complex correlators) with the BJ technique (two complex correlators) to check the tracking ambiguity. Consequently, in its DP version, ASPeCT does not require additional power compared to the BJ technique.

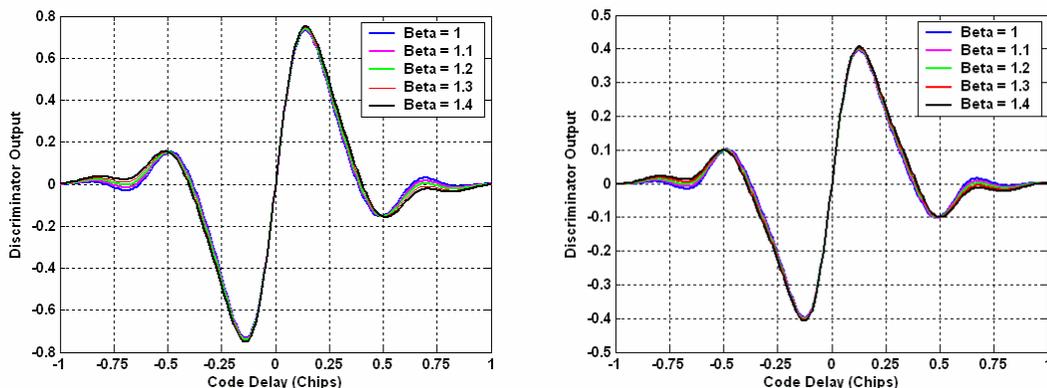
Figure 6.11 shows the traditional and ASPeCT EMLP and DP discriminator outputs for the Galileo L1F signal for a 20 and 3 MHz one-sided front-end filter for  $\beta = 1$ . In the wide front-end filter case, there are no false lock points present since, after approximately 0.6 chips, ASPeCT discriminators' outputs are zero. This means that tracking will never lock at the location of the sBOC(1,1) autocorrelation side-peaks. However, upon examination of the narrow front-end filter case, it can be seen that there is a slight chance of false lock around  $\pm 0.6$  chips. This is in line with the synthesized correlation function analysis done previously, yet the risk is greatly reduced as compared to the classical sBOC(1,1) discriminators since these lock points have an extremely narrow stability region. This means that these false lock points will very likely be unstable for low post-

correlation SNR values. However, it is possible that the DLL would lock on them when the post-correlation SNR is high.



**Figure 6.11 – Traditional and ASPeCT EMLP (Left) and DP (Right) Discriminator Output for using a 20 (Top) and 3 (Bottom) MHz One-Sided Front-End Filter for a 0.2 Chips Correlator Spacing**

This is a good first conclusion since it can be anticipated that it is easier to detect biased tracking for high post-correlation SNR values than for low ones. It has been observed, however, that it is possible to tweak the parameter  $\beta$  to try to make the discriminators fully unambiguous. Figure 6.12 shows the influence of the parameter  $\beta$  on discriminator output. It can be seen that the greater the value of  $\beta$  the more likely that the ASPeCT discriminators will be unambiguous.



**Figure 6.12 – ASPeCT EMLP (Left) and DP (Right) Discriminator Output for different values of Beta using a 3 MHz One-Sided Front-End Filter, and a 0.2 Chips Correlator Spacing**

It is also interesting to see from Figure 6.11 that by using ASPeCT, the discriminators' stability domains are slightly increased as compared to traditional discriminators. The gain is approximately 0.08 chips (one-sided) for  $\beta = 1$ , or approximately 12%, which means that ASPeCT will be slightly more resistant to large errors. Figure 6.12 shows that the greater the value of  $\beta$ , the wider the discriminator stability domain. For a  $\beta$  value of 1.4, the gain is 0.14 chips over the traditional EMLP and DP discriminators.

It has been seen in Chapter 5 that the EMLP and DP discriminators must be normalized in order to cancel their dependency upon the signal power. It has also been concluded that the choice of discriminator did not have a significant impact on loop performance. Consequently, it has been decided to use the instantaneous power estimation ( $N_3$  given in Equation (4.8)) to normalize both ASPeCT discriminators. This results in the following normalized discriminators:

$$V_{EMLP}^{ASPeCT}(\varepsilon_\tau) = \frac{D_{EMLP}^{ASPeCT}}{2(\alpha(2-\alpha\delta) + \beta\delta)N_3} = \frac{\varepsilon_\tau}{(1-\alpha|\varepsilon_\tau|)^2} \xrightarrow{\varepsilon_\tau \rightarrow 0} \varepsilon_\tau \quad (6.21)$$

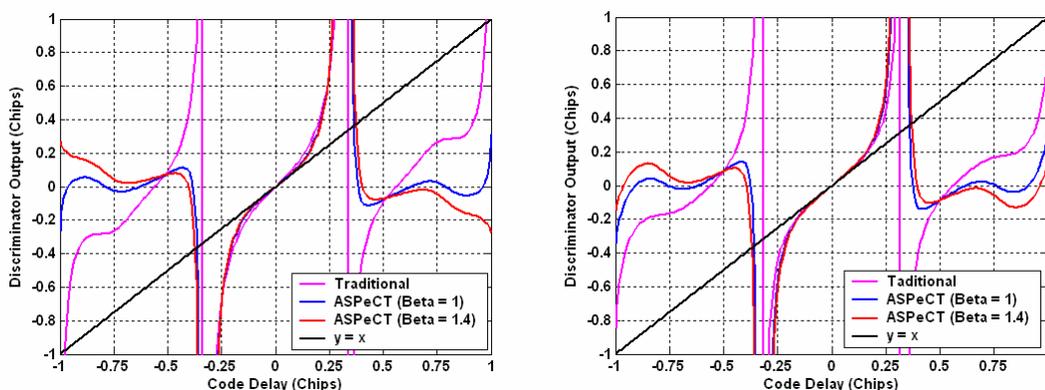
and

$$V_{DP}^{ASPeCT}(\varepsilon_\tau) = \frac{D_{DP}^{ASPeCT}}{2\left(\alpha + \beta\frac{\delta}{2}\right)N_3} = \frac{\left(\alpha(1-\alpha|\varepsilon_\tau|) + \beta\frac{\delta}{2}\right)\varepsilon_\tau}{\left(\alpha + \beta\frac{\delta}{2}\right)(1-\alpha|\varepsilon_\tau|)^2} \xrightarrow{\varepsilon_\tau \rightarrow 0} \varepsilon_\tau \quad (6.22)$$

Figure 6.13 shows the discriminator output associated with Equations (6.21) and (6.22) for a 0.2 chip correlator spacing, a 3 MHz one-sided front-end filter and different values of  $\beta$ . It also shows the traditional EMLP and DP discriminators normalized by  $N_3$ . It can be shown that within the stability domain, the synthesized EMLP and DP discriminators are much like the traditional ones. Consequently, as in the case of traditional tracking, it appears that when normalized by  $N_3$ , the DP discriminator achieves more optimal results than the EMLP discriminator since it more closely follows the first diagonal. At the boundaries though, it can be noted that the stability domain is still extended to the values given previously, and dependent upon the value of  $\beta$ .

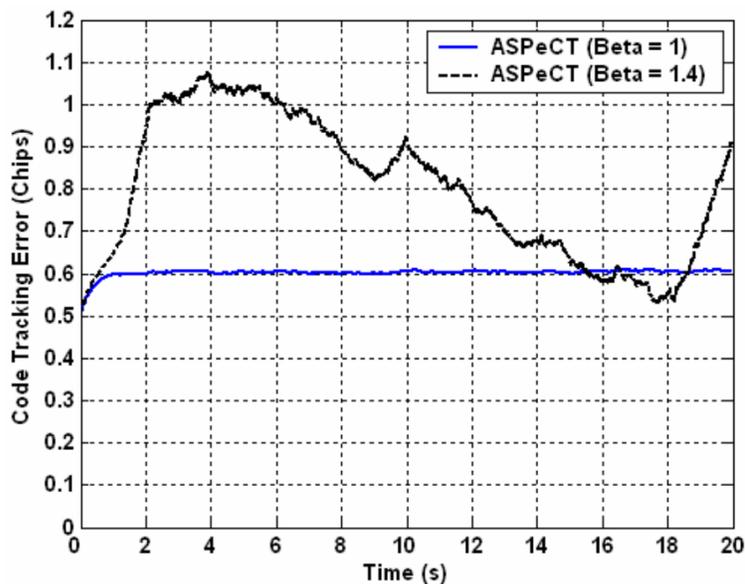
The value of 1.4 for  $\beta$  seems to ensure complete tracking reliability for the synthesized EMLP discriminator. For the synthesized DP, it appears that a false lock point is present around  $\pm 0.96$  chips. This could be considered a problem, but one factor makes such a false lock point very unstable: the normalized prompt correlation value for an error of 0.96 chips is approximately  $\tilde{R}_{ASPeCT}(0.96) = 0.06$ , which is extremely small. Consequently, code tracking will not be possible at this location. In any case, it would be

straightforward to determine if the tracking is erroneous. In order to avoid this possibility, it is also possible to use ASPeCT's EMLP discriminator with  $\beta = 1.4$  since no false lock points are present in this case.



**Figure 6.13 – Traditional and ASPeCT (Beta = 1 and Beta = 1.4) Normalized EMLP (Left) and DP (Right) Discriminator Output**

Figure 6.14 shows examples of code tracking using ASPeCT DP with a  $\beta$  value equal to 1 and 1.4. The  $C/N_0$  has been purposely chosen to be high (50 dB-Hz) to try to emphasize the problem of false lock. The one-sided front-end filter bandwidth has been set to 3 MHz in order to reproduce the conditions of Figure 6.13. The initial code delay error was set to 0.5 chips to simulate false acquisition. It can be seen that for  $\beta$  equal to 1, the DLL using ASPeCT locks around 0.6 chips, which is as expected from Figure 6.13. However, when looking at the graph corresponding to  $\beta = 1.4$ , the DLL simply varies without showing any proof of lock. This means that the DLL does not track any signal. It shows the reliability of ASPeCT when the correct value is given to  $\beta$ .



**Figure 6.14 – Code Delay Error for a Sine-BOC(1,1) Signal Tracking Starting with an Initial Code Delay Error of -0.5 Chips (2 Hz DLL), an Integration Time of 1 ms, and for a  $C/N_0$  of 50 dB-Hz for ASPeCT DP (Beta = 1 and 1.4)**

The architecture of an ASPeCT-based DLL is shown in Figure 6.15. Note that the branches of the figure associated with code generation represent the Early, Prompt and Late replicas, respectively.

Now that ASPeCT's principles have been explained in detail and its unambiguous property has been expounded, it is important to study the impact of the main sources of error on code tracking performance to ensure that it does not imply significant drawbacks. To this end, the effect of thermal noise and multipath are investigated in the following section.

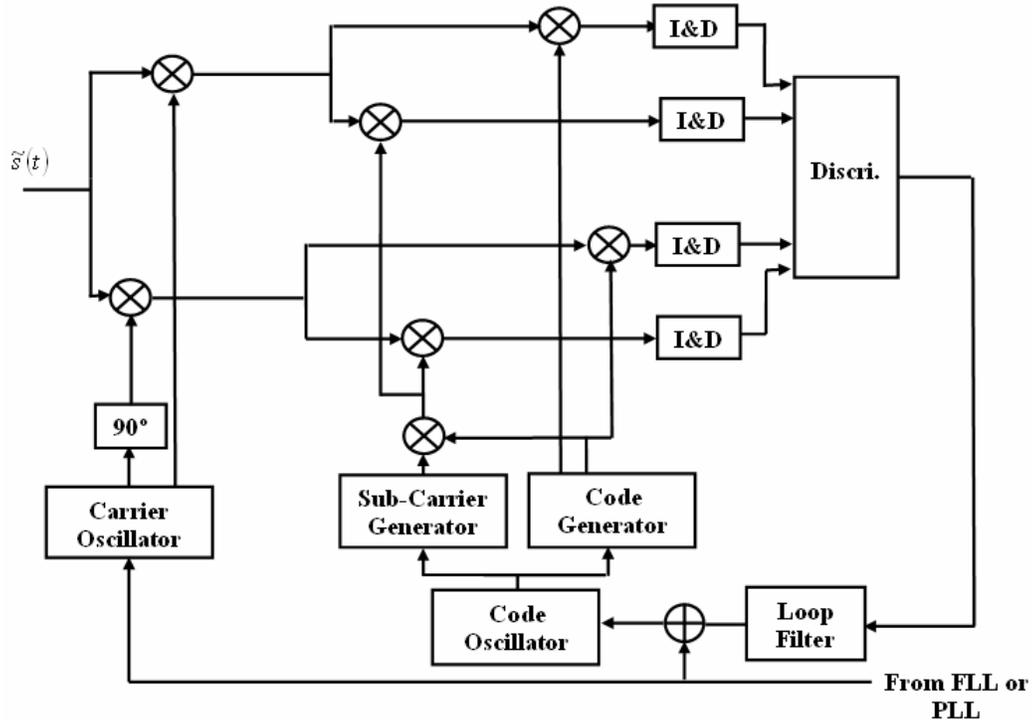


Figure 6.15 – ASPeCT-based DLL Architecture

## 6.3 Impact of Thermal Noise and Multipath on ASPeCT

### 6.3.1 Thermal Noise

ASPeCT code tracking error variance using the DP and EMLP discriminators have been derived in Appendices D and E. They equal, assuming a rectangular front-end filter:

$$\begin{aligned}
\sigma_{EMLP, \varepsilon_\tau}^2 = & \left( \left( B_L(1 - 0.5B_L T_I) \left( \left( \int_{-\infty}^{+\infty} |F_B(f)|^2 \cos(\pi f d) df \right)^2 \left( \int_{-\infty}^{+\infty} |F_B(f)|^2 \sin^2(\pi f d) df \right) \right. \right. \right. \\
& \left. \left. \left. + \beta^2 \left( \int_{-B}^{+B} jF_B(f) F_P^*(f) \sin(\pi f d) df \right)^2 \left( \int_{-B}^{+B} |F_P(f)|^2 \cos^2(\pi f d) df \right) \right) \right) \\
& \frac{P}{N_0} (2\pi)^2 \left( \left( \int_{-B}^{+B} |F_B(f)|^2 \cos(\pi f d) df \right) \times \left( \int_{-B}^{+B} f |F_B(f)|^2 \sin(\pi f d) df \right) - \right. \\
& \left. \beta \left( \int_{-B}^{+B} F_B(f) F_P^*(f) \sin(\pi f d) df \right) \times \left( \int_{-B}^{+B} f F_B(f) F_P^*(f) \cos(\pi f d) df \right) \right)^2 \\
& \times \left( \left( \int_{-B}^{+B} |F_B(f)|^2 \sin^2(\pi f d) df \right) \left( \int_{-B}^{+B} |F_B(f)|^2 \cos^2(\pi f d) df \right) \right. \\
& \left. + \beta^2 \left( \int_{-B}^{+B} |F_P(f)|^2 \sin^2(\pi f d) df \right) \left( \int_{-B}^{+B} |F_P(f)|^2 \cos^2(\pi f d) df \right) \right. \\
& \left. + \frac{\beta}{2} \left( \int_{-B}^{+B} jF_B(f) F_P^*(f) \cos(2\pi f d) df \right)^2 \right) \\
& \times \left( 1 + \frac{P}{N_0} T_I \left( \left( \int_{-\infty}^{+\infty} |F_B(f)|^2 \cos(\pi f d) df \right)^2 \left( \int_{-\infty}^{+\infty} |F_B(f)|^2 \sin^2(\pi f d) df \right) \right. \right. \\
& \left. \left. + \beta^2 \left( \int_{-B}^{+B} jF_B(f) F_P^*(f) \sin(\pi f d) df \right)^2 \left( \int_{-B}^{+B} |F_P(f)|^2 \cos^2(\pi f d) df \right) \right) \right)
\end{aligned} \tag{6.23}$$

and

$$\sigma_{DP, \varepsilon_\tau}^2 = \left( \begin{array}{c} B_L(1 - 0.5B_L T_I) \left( \begin{array}{c} \left( \int_{-B}^{+B} |F_B|^2 df \right)^2 \left( \int_{-B}^{+B} |F_B|^2 \sin^2(\pi f d) df \right) \\ + 2\beta^2 \left( \int_{-B}^{+B} jF_B(f)F_P^*(f)\sin(\pi f d) df \right)^2 \left( \int_{-B}^{+B} |F_P(f)|^2 df \right) \end{array} \right) \\ \hline \frac{P}{N_0} \left( \begin{array}{c} \left( \int_{-B}^{+B} f |F_B(f)|^2 \sin(\pi f d) df \right) \left( \int_{-B}^{+B} |F_B(f)|^2 df \right) - \\ \beta \left( \int_{-B}^{+B} F_B(f)F_P^*(f)\sin(\pi f d) df \right) \left( \int_{-B}^{+B} f F_B(f)F_P^*(f) df \right) \end{array} \right) \right)^2 \\ \times \left( \begin{array}{c} \left( \int_{-B}^{+B} |F_B|^2 df \right) \left( \int_{-B}^{+B} |F_B|^2 \sin^2(\pi f d) df \right) \\ + \beta^2 \left( \int_{-B}^{+B} |F_P(f)|^2 df \right) \left( \int_{-B}^{+B} |F_P(f)|^2 \sin^2(\pi f d) df \right) \\ + 2\beta \left( \int_{-B}^{+B} jF_B(f)F_P^*(f)\sin(\pi f d) df \right)^2 \end{array} \right) \\ \times \left( 1 + \frac{P}{N_0} T_I \left( \begin{array}{c} \left( \int_{-B}^{+B} |F_B|^2 df \right)^2 \left( \int_{-B}^{+B} |F_B|^2 \sin^2(\pi f d) df \right) \\ + 2\beta^2 \left( \int_{-B}^{+B} jF_B(f)F_P^*(f)\sin(\pi f d) df \right)^2 \left( \int_{-B}^{+B} |F_P(f)|^2 df \right) \end{array} \right) \right) \end{array} \right)$$

(6.24)

Assuming an infinite front-end filter, these equations can be reduced to:

$$\sigma_{EMLP, \varepsilon_\tau}^2 = \left( \frac{B_L(1 - 0.5B_L T_I) \delta (12 + 2\delta(\beta^2 - 18)) - \delta^2(27 - \beta^2)}{2 \frac{C}{N_0} (3(2 - 3\delta) + \beta\delta)^2} \right) \left( \text{Chips}^2 \right) \quad (6.25)$$

$$\times \left( 1 + \frac{2(2(3 + \beta^2) + \delta(2\beta - (9 + \beta^2)))}{\frac{C}{N_0} T_I (12 + 2\delta(\beta^2 - 18)) - \delta^2(27 - \beta^2)} \right)$$

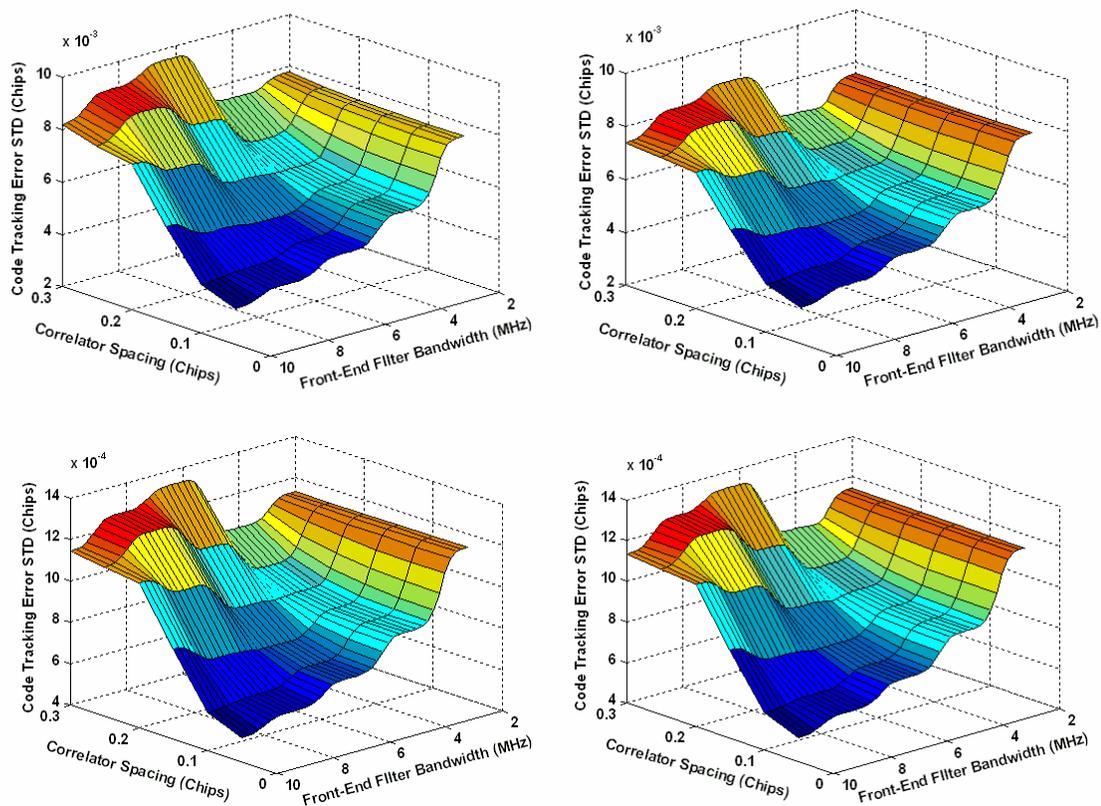
and

$$\sigma_{DP, \varepsilon_\tau}^2 = \frac{B_L(1 - 0.5B_L T_I) \delta (6 + \beta^2 d)}{\frac{C}{N_0} (6 + \beta\delta)^2} \left( 1 + \frac{2(\beta\delta + 3 + \beta^2)}{\frac{C}{N_0} T_I (6 + \beta^2 \delta)} \right) \left( \text{Chips}^2 \right) \quad (6.26)$$

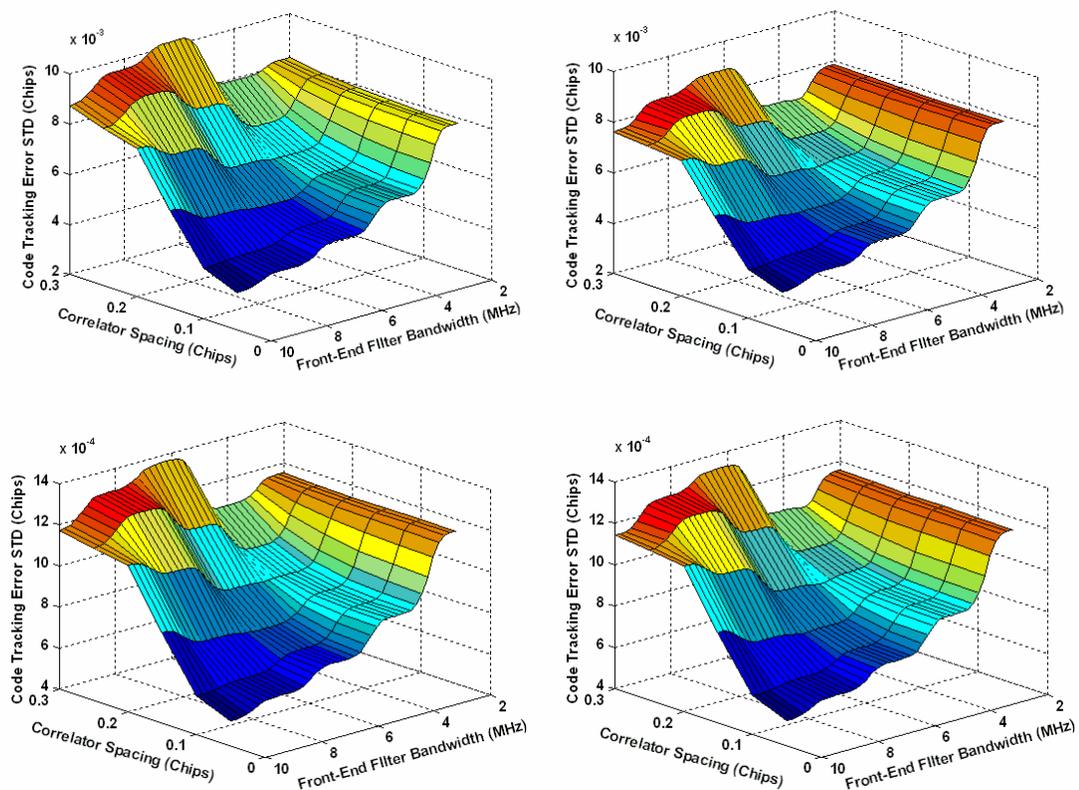
ASPeCT's code tracking error standard deviation using the proposed DP or the EMLP discriminator is shown in Figure 6.16 ( $\beta = 1$ ) and Figure 6.17 ( $\beta = 1.4$ ) for a low (30 dB-Hz) and a high (45 dB-Hz)  $C/N_0$  assuming a 4 ms coherent integration time for different front-end filter bandwidths and correlator spacing values. It can be seen that the general code tracking error standard deviation is very similar to the classical sBOC(1,1) tracking shown in Figure 5.7 and Figure 5.8. This shows that the contribution of the BOC/PRN correlation is limited in the error budget. The reason for this can be analyzed following the calculations realized in Appendices D and E. The main reason for this is that the BOC/PRN correlation has low values around zero code delay error and since the sBOC(1,1) correlator spacing is low, it will not produce a lot of extra-power.

It can be observed that ASPeCT's EMLP discriminator seems to suffer from a higher squaring loss, as in the case of the traditional discriminators. A comparison of Figure

6.16 and Figure 6.17 also shows that the impact of the parameter  $\beta$  is limited, and just slightly decreases the code tracking accuracy.



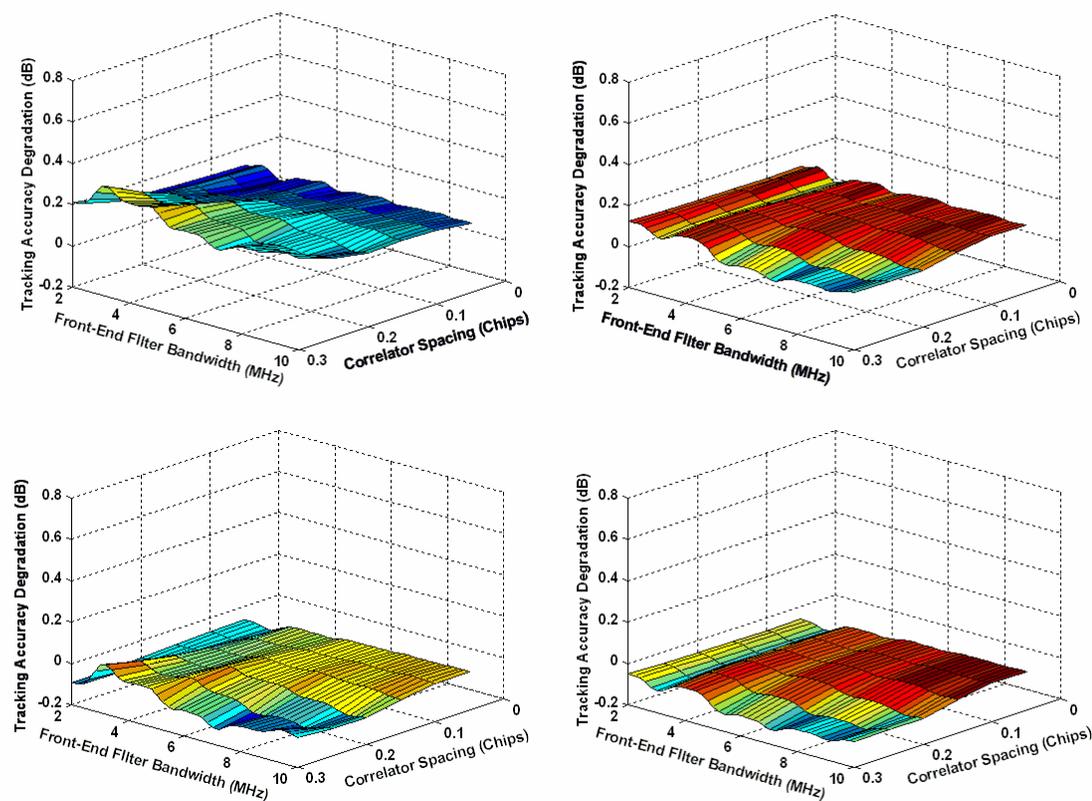
**Figure 6.16 – EMLP (Left) and DP (Right) Code Tracking Error Standard Deviation using ASPeCT (Beta = 1) and a 30 (Top) and 45 (Bottom) C/N<sub>0</sub> and a 4 ms Coherent Integration Time**



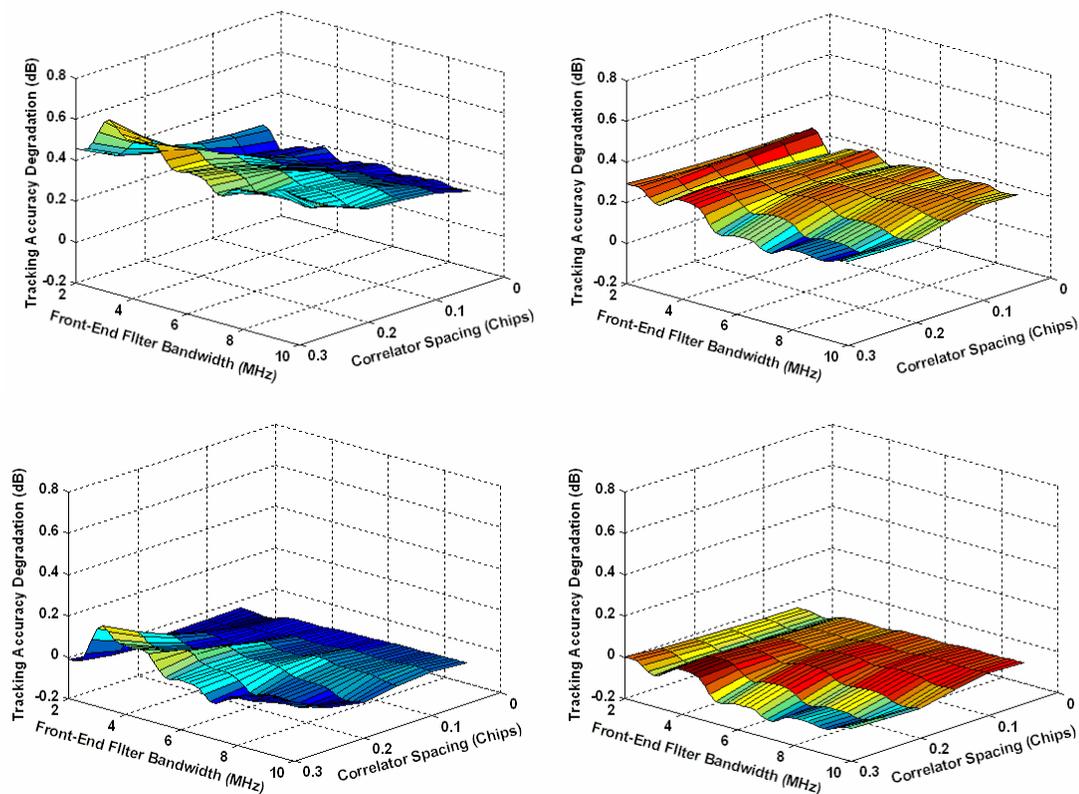
**Figure 6.17 - EMLP (Left) and DP (Right) Code Tracking Error Standard Deviation using ASPeCT ( $\beta = 1.4$ ) and a 30 (Top) and 45 (Bottom)  $C/N_0$  and a 4 ms Coherent Integration Time**

The degradation produced through the use of ASPeCT over traditional tracking using an EMLP and a DP discriminator is shown in Figure 6.18 ( $\beta = 1$ ) and Figure 6.19 ( $\beta = 1.4$ ) for  $C/N_0$  values of 30 and 45 dB-Hz. It can be seen that the degradation is minimal, and dependent upon the  $C/N_0$  value. Indeed, for low  $C/N_0$  values, the impact of the noise imparted by the BOC/PRN correlation will start to be significant, and will thus impact the code tracking error more seriously. Of course, the higher the value of  $\beta$ , the greater the significance of this effect. However, the degradation is still small compared to the use of a BPSK(1) modulation which, as shown in Section 5.2.1.1, exhibits a degradation of the

tracking error standard deviation on the order of 2 to 3 dBs. With ASPeCT, this degradation is, for a 30 dB-Hz incoming signal, only around 0.2-0.3 dBs for  $\beta = 1$  and around 0.3-0.5 dBs for  $\beta = 1.4$ . In the 45 dB-Hz case, it can even be seen that ASPeCT tracking will be slightly more accurate or equal to the traditional sBOC(1,1) tracking for both  $\beta = 1$  and  $\beta = 1.4$ . This is due to the fact that as seen in Equations (6.19) and (6.20), both the ASPeCT DP and EMLP discriminators entail a higher gain than their traditional counterparts.



**Figure 6.18 – DP and EMLP Code Tracking Error Standard Deviation Degradation Brought from ASPeCT (Beta = 1) Compared to Traditional Tracking for a 30 (Top) and 45 (Bottom)  $C/N_0$  and a 4 ms Coherent Integration Time**



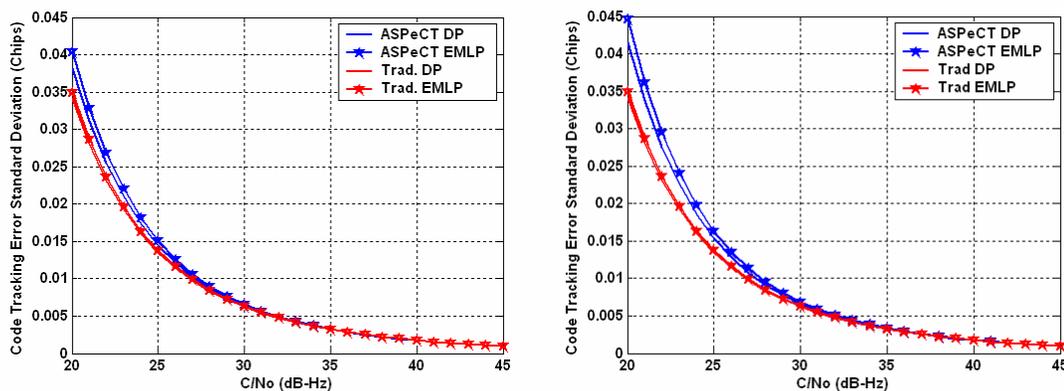
**Figure 6.19 - DP and EMLP Code Tracking Error Standard Deviation Degradation Brought from ASPeCT (Beta = 1.4) Compared to Traditional Tracking for a 30 (Top) and 45 (Bottom)  $C/N_0$  and a 4 ms Coherent Integration Time**

Consequently, choosing a higher value for  $\beta$  to ensure that the tracking is unambiguous does not produce too much extra-noise. This is very important, since it implies that ASPeCT can provide reliable unambiguous tracking with an accuracy that is significantly higher than the SSL technique, and which is comparable to the performance of the BJ technique.

It is also interesting to see that ASPeCT's DP discriminator suffers less from a large correlator spacing than ASPeCT's EMLP discriminator. This results from the fact that it uses the prompt BOC/PRN correlation value that should equal zero if tracking is

perfectly centred. It thus appears that it would be the best candidate for implementing ASPeCT in a Galileo L1F receiver.

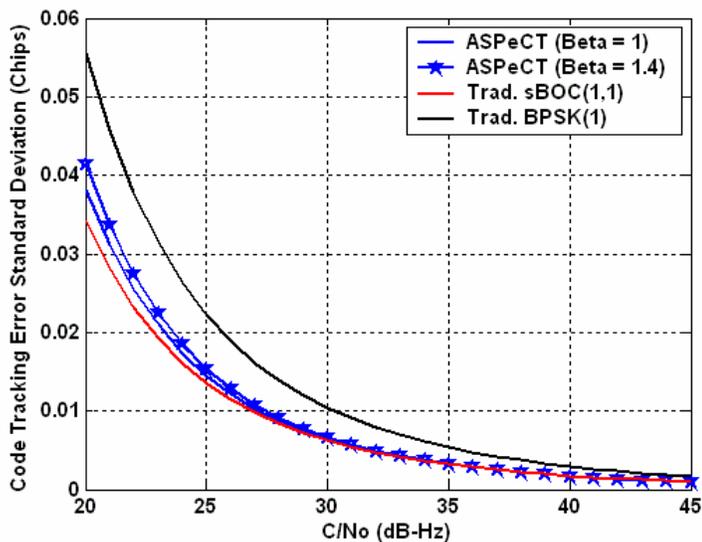
In order to get a better view of the actual code tracking error standard deviation as a function of the  $C/N_0$ , Figure 6.20 shows the ASPeCT DP and EMLP code tracking error standard deviations for  $\beta = 1$  and  $\beta = 1.4$  for a 3 MHz one-sided front-end filter and a 0.2 chip correlator spacing. As it appears that the ASPeCT DP implementation slightly outperforms the EMLP implementation, and that it is more favourable in terms of complex correlator requirements, the next part of this thesis will consist of a consideration of only ASPeCT DP implementation, unless otherwise stated.



**Figure 6.20 – Code Tracking Error Standard Deviation for ASPeCT DP and EMLP Discriminators (Beta = 1 on Left and Beta = 1.4 on Right) using a 0.2 Chip Correlator Spacing, a 3 MHz one-sided front-end filter and a 4 ms Coherent Integration Time, and a 1 Hz Loop Bandwidth**

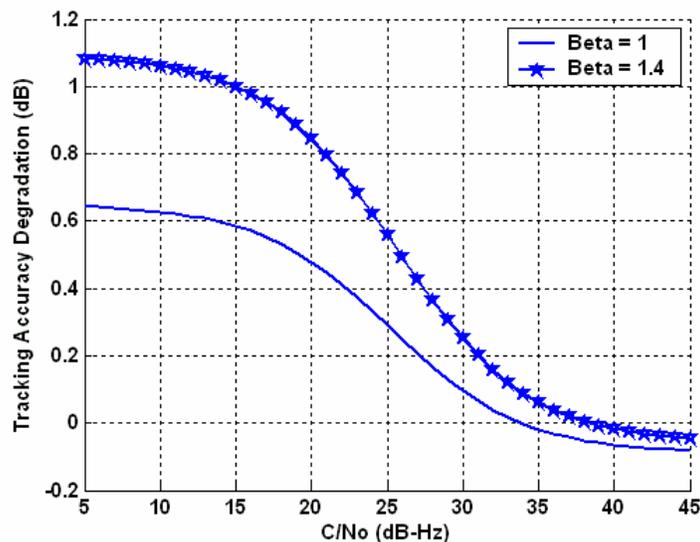
Figure 6.21 compares the ASPeCT DP, traditional SBOC(1,1), and traditional BPSK(1) tracking techniques for a coherent integration time of 4 ms, a 0.2 chip correlator spacing, and a 3 MHz one-sided front-end filter. As previously explained, the tracking degradation

using ASPeCT appears to be small and dependent upon the value of  $\beta$ . The lower the value of  $\beta$ , the higher the accuracy of ASPeCT's tracking performance.



**Figure 6.21 – Code Tracking Error Standard Deviation vs. the  $C/N_0$  for the Traditional Sine-BOC( $n,n$ ) Tracking and ASPeCT DP (Beta = 1 and Beta = 1.4) with a Correlator Spacing of 0.2 Chips, a Coherent Integration Time of 4 ms, a 3 MHz Front-End Filter Bandwidth, and a 1 Hz Loop Bandwidth**

ASPeCT's code tracking error degradation, with respect to traditional sBOC(1,1) tracking, is shown in Figure 6.22. It seems to stabilize for very low  $C/N_0$  around 0.6 dB for  $\beta = 1$  and 1 dB for  $\beta = 1.4$ , which would still allow a tracking accuracy significantly better than using a BPSK(1) modulation. It is interesting to confirm that ASPeCT slightly outperforms conventional sBOC(1,1) tracking for high  $C/N_0$  (great than 35 and 40 dB-Hz for  $\beta = 1$  and  $\beta = 1.4$ , respectively).

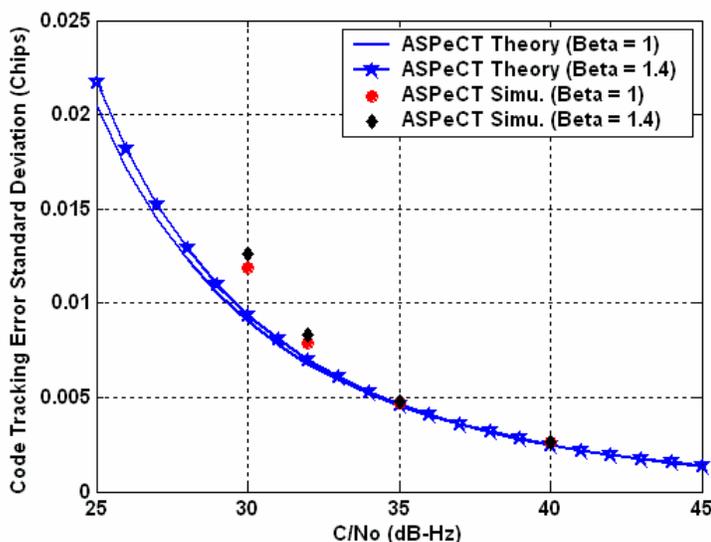


**Figure 6.22 – ASPeCT DP Code Tracking Error Standard Deviation Degradation against Traditional sBOC(1,1) Tracking (DP) vs.  $C/N_0$  (Beta = 1 and Beta = 1.4) using a 3 MHz Front-End Filter Bandwidth, a 0.2 Chip Correlator Spacing, a Coherent Integration Time of 4 ms**

Finally, Figure 6.23 shows the results of a 20-second simulation using ASPeCT DP implementation for  $\beta = 1$  and  $\beta = 1.4$ . The simulation used a 0.4 chip discriminator output hard limiter, as described in Section 5.2.1.2, and the discriminator was normalized using  $N_3$  (see Section 6.2). The figure shows an excellent agreement of the theory and the simulation results for  $C/N_0$  values over 35 dB-Hz. A slight divergence is seen below this value. This divergence is the result of two main causes. First, the theoretical expression does not take into account the extra noise produced by normalization of the discriminator. A study of the effect of normalization on overall tracking performance in Gaussian noise could not be conducted as in Section 5.2.1.2 because the odd shape of the BOC/PRN correlation function makes the matrix C in Equation (5.36) non-symmetrical and thus it is not possible to generate the correlated noise of the various correlator

outputs. However, when looking at the shape of the simulated points in Figure 6.23, it seems to follow the trend observed for the results associated with traditional sBOC(1,1) tracking using normalization  $N_3$  shown in Figure 5.16.

Second, as the  $C/N_0$  decreases, the tracking error increases, the assumption of small tracking error used to reach the theoretical expression of the tracking error variance becomes weaker.



**Figure 6.23 – ASPeCT DP implementation Theoretical and Simulated Code Tracking Error using a 4 ms Coherent Integration Time, a 0.2 Chips Correlator Spacing, a 3 MHz Front-End Filter, a 2 Hz Loop Bandwidth and a 0.4 Chips Discriminator Hard Limiter**

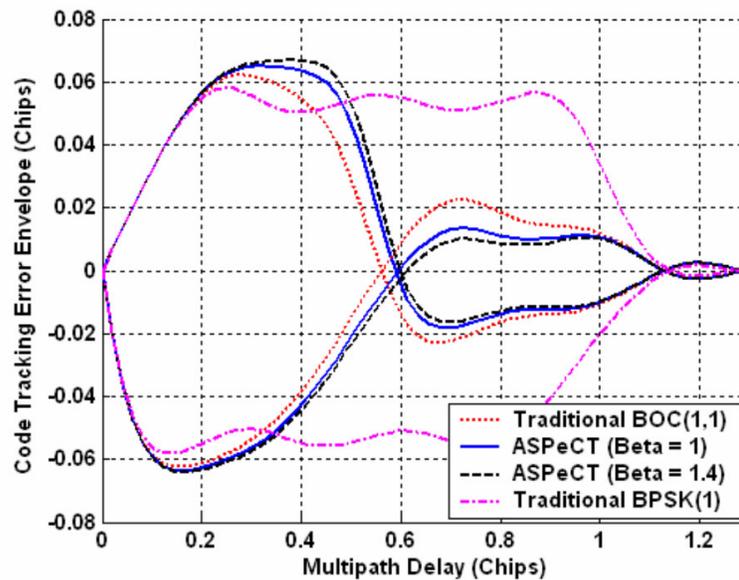
### 6.3.2 Multipath

Multipath-induced error on traditional sBOC(1,1) tracking has been studied in Section 5.2.2. It was seen that its impact on code tracking depends upon the relative delay and phase difference of the multipath with the direct signal, as well as its relative magnitude.

When only one multipath reflector is considered, its resulting error on code tracking is contained within the envelope of the error created by in-phase and out-of-phase multipath.

Figure 6.24 shows the multipath-induced error envelope for traditional sBOC(1,1) tracking, as well as for ASPeCT ( $\beta = 1$  and  $\beta = 1.4$ ) for a signal-to-multipath-amplitude ratio of 0.5 and a 3 MHz one-sided front-end filter and a 0.2 chip correlator spacing (the envelope is the same for the DP and EMLP version). The difference between the two tracking methods is minimal. For multipath delays within [0.25; 0.5] chips, the traditional method seems to mitigate multipath slightly better than ASPeCT while for multipath delays within [0.6; 1] chips, the opposite is true. The choice of  $\beta$  seems to have only a limited effect on the shape of the multipath envelope.

ASPeCT offers inherent multipath mitigation which is similar to traditional sine-BOC(1,1) tracking. This is extremely important, as the SSL technique has a multipath envelope close to that of BPSK(1) signals, which is significantly larger (see Figure 6.24). Moreover, it should be kept in mind that the BJ technique relies on a statistical test (comparing the magnitude of the side-peaks) that could be affected during the checking process when strong multipath is present, as seen in Figure 6.8. The problems that could be raised by high amplitude multipath when using the BJ technique were discussed in Section 6.1.3.2.



**Figure 6.24 – Code Tracking Multipath Envelope for Traditional sBOC(1,1), Traditional BPSK(1), and ASPeCT (Beta = 1 and Beta =1.4) for a 0.2 Chips Correlator Spacing, and a 3 MHz One-Sided Front-End Filter**

### 6.3.3 Conclusions on ASPeCT's Error Mitigation

The study of the thermal noise and multipath impact on ASPeCT's tracking performance has shown that it is close to traditional sBOC(1,1) tracking performance. The role of the parameter  $\beta$  in the magnitude of the resulting error in the presence of thermal noise and multipath seems limited. This is of critical importance as it implies that ASPeCT can be adapted to any receiver architecture (front-end filter and correlator spacing) by a simple calibration of the value given to  $\beta$  without significantly degrading its tracking performance. This means that in terms of tracking accuracy, ASPeCT is equivalent to the use of the BJ technique (provided that the BJ technique is tracking the main peak!), and is always significantly better than the SSL method. Moreover, ASPeCT offers only reliable measurements. Either the code delay estimation is correct, or the DLL has lost lock.

There is no chance of obtaining momentary biased tracking as in the case of the BJ technique. Finally, another advantage of ASPeCT is that its DP version requires exactly the same number of complex correlators as the BJ technique, so employing this method requires no extra power.

One of the potential drawbacks of ASPeCT is that it does not allow recovery from an absolute code delay error greater than approximately 0.38 chips (depending upon the value of  $\beta$ ) due to its reduced stability domain, unlike the SSL method, and the BJ technique (assuming first a false lock and then a jump on the main peak). This could be a drawback in case of short losses of lock. One way to compensate for this could be to use a method similar to the method implemented in Section 5.4 where an HRC tracking technique was backed up by a more robust tracking method. In the present case, ASPeCT would be bolstered by a method such as the SSL technique.

It is important to mention that since all the sBOC(n,n) family shares the same autocorrelation function, ASPeCT can easily be extended to all the family (Julien *et al.* 2004c). However, in such a case, the choice of the parameter  $\beta$  should be calibrated since different front-end filters should be used (the sBOC(n,n) PSD, with  $n > 1$ , is wider than that of the sBOC(1,1) PSD).

ASPeCT's discriminators' reduced stability domain, as compared to traditional BPSK(1) discriminators, also means that acquisition on a side-peak, even though it is an error of only 0.5 chips, will result in a loss of lock. To solve this problem, one solution is to use the SSL technique to acquire the signal and first have a convergence of the SSL tracking

loop toward the correct code delay before transferring to code tracking via ASPeCT.

This approach would be similar to that presented by Martin *et al.* (2002). Another solution is to use ASPeCT's synthesized correlation function in an acquisition scheme to remove any false acquisition, since it does not contain any side-peaks. This is investigated in the following section.

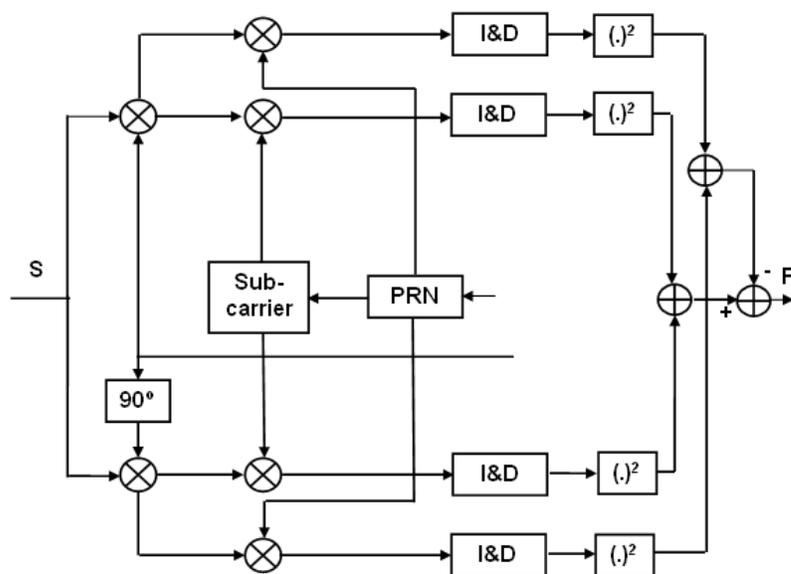
#### 6.4 Acquisition Using ASPeCT's Synthesized Correlation Function

As explained in Section 6.1.1, the side-peaks of the sBOC(1,1) autocorrelation function constitute a real threat for an sBOC(1,1) signal acquisition process due to their non-trivial magnitude. The essence of ASPeCT is to create a synthesized correlation function that would cancel the sBOC(1,1) side-peaks in order to eliminate the false tracking lock points. Figure 6.10 shows that ASPeCT's synthesized correlation function effectively has no positive side-peak. Moreover, as it will be seen later on, the negative side-peaks do not pose any problem to the statistical test since only positive values could pass the acquisition test. Consequently, it seems appropriate to try to use ASPeCT's synthesized correlation function in order to implement an unambiguous acquisition scheme.

A detailed description of the spread spectrum signal acquisition theory has already been briefly described in Section 6.1.1. When considering an acquisition scheme using ASPeCT's synthesized correlation function, it seems natural to introduce the following statistical test:

$$T_{ASPeCT} = \sum_{k=1}^M \left( \left( I_{P,k}^2 + Q_{P,k}^2 \right)_B - \beta \left( I_{P,k}^2 + Q_{P,k}^2 \right)_{BP} \right) \quad (6.27)$$

The synopsis of the resultant acquisition scheme is given in Figure 6.25.



**Figure 6.25 – Synopsis of ASPeCT Acquisition Architecture**

According to the shape of ASPeCT's synthesized correlation function given in Figure 6.10, the coefficient  $\beta$  does not seem to play an important role in the pure cancellation of the positive side-peak ( $\beta$  was intended to remove any peak pointing upward, even if they were negative in magnitude). As a result,  $\beta$  will be assumed to be equal to 1 herein since it ensures no major side-peaks in the synthesized correlation function, and it can be anticipated that it will provide a minimum extra-noise coming from the BOC/PRN correlation values. The choice of lower  $\beta$  (lower than 1) values could also be investigated, but this is not done herein. One drawback with the test criterion introduced by Equation (6.27) is that it requires two complex correlators (for BOC/BOC and BOC/PRN correlations) for each code delay/Doppler pair tried by the acquisition algorithm, while a traditional acquisition scheme would need only one (see Equation

(6.1)). This problem can be solved by increasing the number of correlators in the receiver, but this strategy would require more available power.

In order to assess ASPeCT's acquisition test criterion shown in Equation (6.27), it is important to be able to model the noise contribution. Assuming a Gaussian incoming noise, it has been demonstrated in Julien *et al.* (2004c) that the noise coming from the prompt BOC/BOC and BOC/PRN correlators were uncorrelated and could be assumed Gaussian. Moreover, according to Papoulis (1991), the difference between two independent random variables has a distribution that is the convolution between the distribution of the first random variable and the distribution of the opposite of the second random variable. This can be translated into:

$$p_{T_{ASPeCT}}(x) = p_{T_B}(x) \otimes p_{T_{B/P}}(-x) \quad (6.28)$$

where  $p_{T_B}$  is the distribution of the random variable  $\sum_{k=1}^M \left( I_{P,k}^2 + Q_{P,k}^2 \right)_B$ , and

$$p_{T_{BP}} \text{ is the distribution of the random variable } \sum_{k=1}^M \left( I_{P,k}^2 + Q_{P,k}^2 \right)_{BP}.$$

Assuming Gaussian noise only, it is well known that  $\frac{p_{T_B}}{\sigma_{n_B}^2}$  and  $\frac{p_{T_{BP}}}{\sigma_{n_{BP}}^2}$  have a Chi-

square distribution. Assuming a sufficiently wide front-end filter, it has been observed in Julien *et al.* (2004c) that the power of the noise originating in the BOC/BOC and BOC/PRN correlators are the same. Using Equation (6.28), it is then possible to model

the distribution of  $\frac{T_{ASPeCT}}{\sigma_{n_B}^2}$  as the convolution of two chi-square distributions. By

correctly modeling the distribution of  $\frac{T_{ASPeCT}}{\sigma_{n_B}^2}$ , it is then straightforward to

determine the probability of detection of the main peak, given a certain probability of false alarm, and a certain  $C/N_0$ , as discussed in Section 6.1.1.

Since the acquisition criterion is the difference between two Chi-square distributions, its expected value can be expressed as:

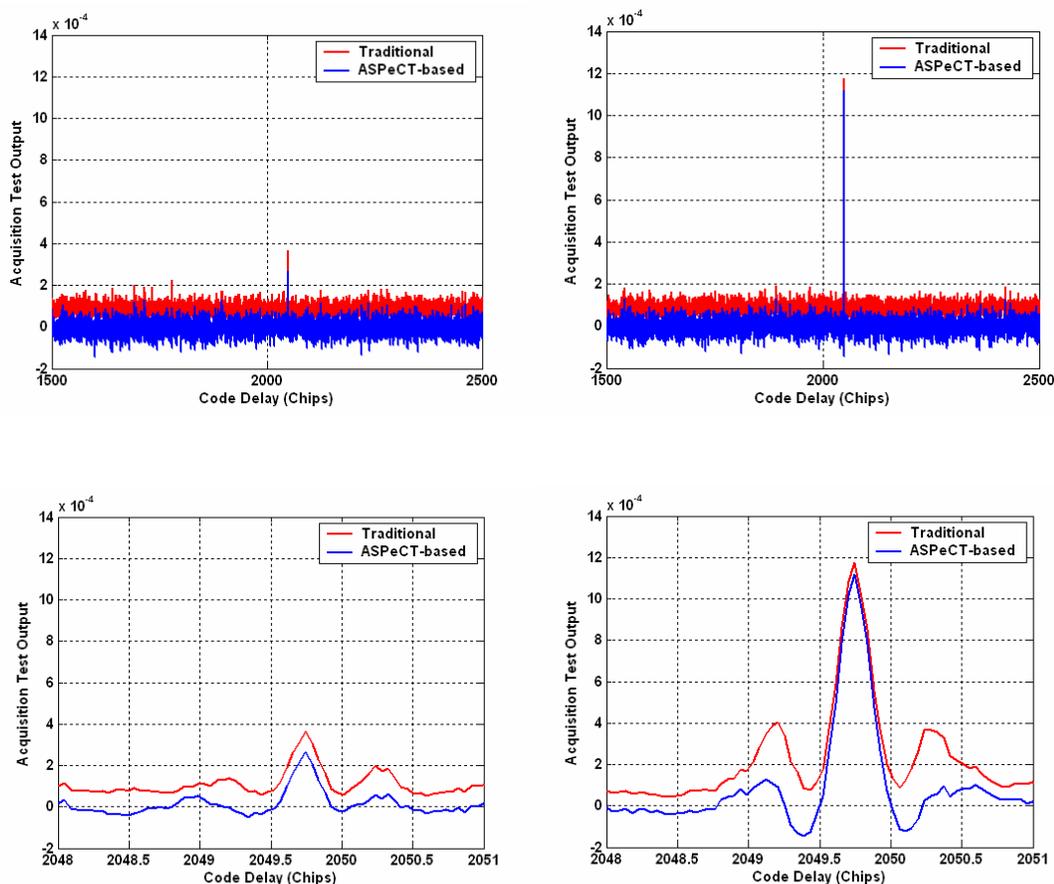
$$\langle T_{ASPeCT} \rangle = \langle T_B \rangle - \langle T_{BP} \rangle \quad (6.29)$$

From Equation (6.29), it is possible to compare the mean of the new acquisition criterion with the mean of the standard acquisition criterion (symbolized by  $T_B$ ). For this purpose, one figure of merit has been defined: the ratio of the means ( $FOM1$ ). From the theoretical expression of the mean of chi-square distributions (Papoulis 1991), this figure of merit can be expressed as:

$$FOM1 = \frac{\frac{CT_I}{N_0} \left( (R_B(\varepsilon_\tau))^2 - (R_{BP}(\varepsilon_\tau))^2 \right) \left( \frac{\sin(\pi \varepsilon_f T_I)}{\pi \varepsilon_f T_I} \right)^2}{\frac{CT_P}{N_0} (R_B(\varepsilon_\tau))^2 \left( \frac{\sin(\pi \varepsilon_f T_I)}{\pi \varepsilon_f T_I} \right)^2 + 2} \quad (6.30)$$

From Equation (6.30), it can be seen that  $FOM1$  favours the standard acquisition criterion. Indeed, the ratio of the means will always be smaller than 1, meaning that  $\langle T_{ASPeCT} \rangle$  will always be smaller than  $\langle T_B \rangle$  when the same acquisition parameters are used. Moreover, this is emphasized by the fact that  $R_{BP}^2(\varepsilon_\tau) = \varepsilon_\tau^2$ .

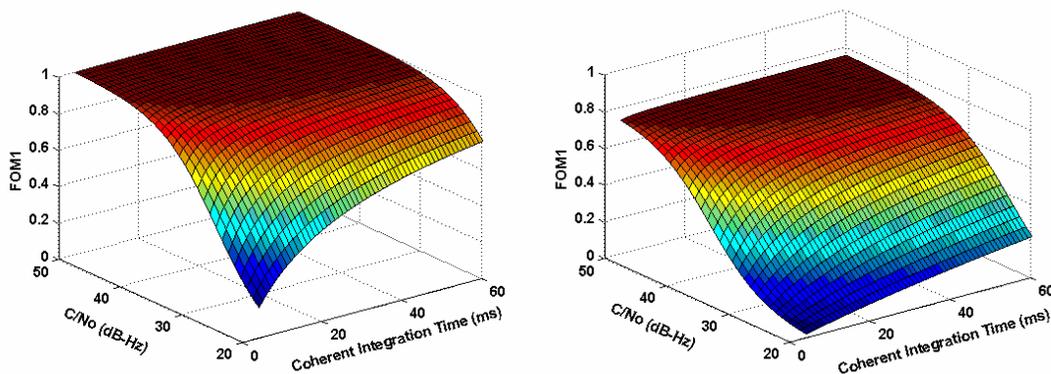
However, it is important to note that the distribution of  $T_{ASPeCT}$  noise will differ sharply from the classical  $T_B$  distribution. Indeed,  $T_{ASPeCT}$  will subtract two squared Gaussian noise values. Consequently, this will result in a reduction of the maximum positive magnitude of  $T_{ASPeCT}$  noise, as compared to  $T_B$  noise. This will have an important impact on ASPeCT's acquisition performance as seen in Figure 6.26 in the cases of a 30 and 35 dB-Hz signal using a 4 ms coherent integration time, and 15 non-coherent summations.



**Figure 6.26 – Example of the Output of the Traditional and ASPeCT-based Acquisition Test Criteria for a 30 (Left) and 35 (Right)  $C/N_0$ , a 4 ms Coherent integration, and 15 Non-Coherent Integrations (Wide (Top) and Close-up around the Main Peak (Bottom))**

Finally, it is interesting to note that the number of non-coherent summations has no impact on  $FOM1$ . This means that, *a priori*, the difference between the two criteria cannot be bridged using a large number of non-coherent summations. Equations (6.30) also show that the difference between the mean of  $T_{ASPeCT}$  and  $T_B$  is reduced when the value of  $\frac{CT_I}{N_0}$  increases. However, the uncertainty associated with the code delay and frequency used for the acquisition process must also be considered. For this purpose, it has been decided to assume that the code delay bin size was 0.4 chips, and the Doppler bin size was  $\frac{0.4}{T_I}$ , which is in line with the discussion in Section 6.1.1.

The impact of the  $C/N_0$  and  $T_I$  on  $FOM1$  is represented in Figure 6.27 for the case when there is no code delay/Doppler uncertainty and the case when the error corresponds to the boundary of the code and Doppler bin sizes. It can be seen that for increasing  $C/N_0$  and  $T_I$ ,  $FOM1$  tends quickly towards 1. However, the presence of the code delay and Doppler uncertainties makes  $FOM1$  always smaller than 1 (and even smaller than 0.8 at the chosen boundaries). This might constitute a drawback with respect to ASPeCT's acquisition performance. One possible way to reduce that problem could be to reduce the code delay/Doppler bin size.



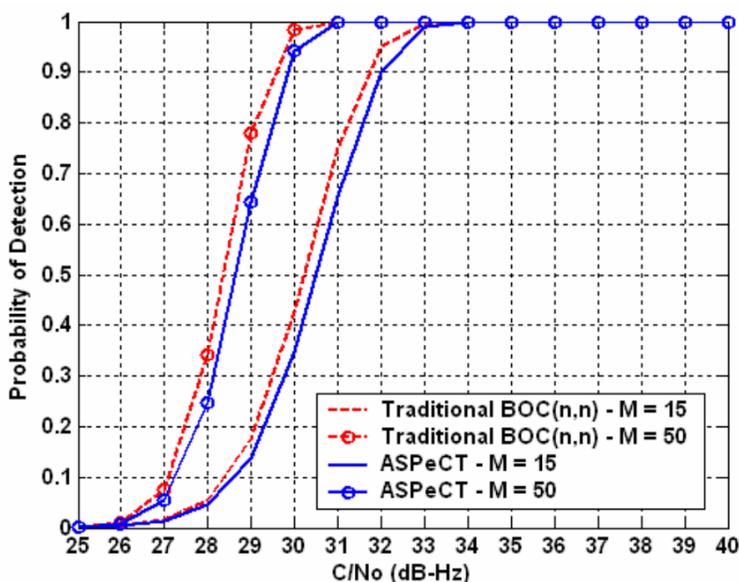
**Figure 6.27 – Impact of the Coherent Integration Time and the  $C/N_0$  on FOM1 for no (Left) and Maximum (Right) Code Delay/Doppler Uncertainty**

Now that ASPeCT's acquisition test function has been analyzed, and that the impact of different parameters on the mean of  $T_{ASPeCT}$  has been investigated, it is worthwhile to assess the actual performance of ASPeCT's acquisition scheme.

The proposed Galileo L1F, described in Section 2.3, was taken as an example.  $T_{ASPeCT}$  distribution was modeled using Equation (6.28). The coherent integration time was assumed to be 4 ms. No code delay or Doppler uncertainty is assumed in a first trial, and the same settings as in Section 6.1.1 are taken. The resulting  $P_d$  is shown in Figure 6.28 for a  $C/N_0$  ranging from 25 to 40 dB-Hz, 15 and 50 non-coherent summations, and using both ASPeCT and traditional acquisition schemes. It can be seen that the sensitivity of ASPeCT's acquisition process is only slightly degraded (by less than 0.5 dB in terms of equivalent  $C/N_0$ ), as compared to the traditional acquisition scheme. This can be understood by reference to Figure 6.27. Indeed, at the  $C/N_0$  when the probability of detection increases for the traditional acquisition method, it approximately corresponds to the moment when the ratio of the means (between the conventional and ASPeCT test

criteria) tends toward 1. Moreover, as has been emphasized above, the  $T_{ASPeCT}$  noise distribution does not have a magnitude greater than the traditional test criteria.

It must be recalled that the correlation on which ASPeCT's acquisition process is based does not contain side-lobes, making it more reliable than the traditional sBOC(1,1) acquisition scheme, which might detect one of the side-peaks only 6 dB in power under the main peak, as shown in Section 6.1.1.

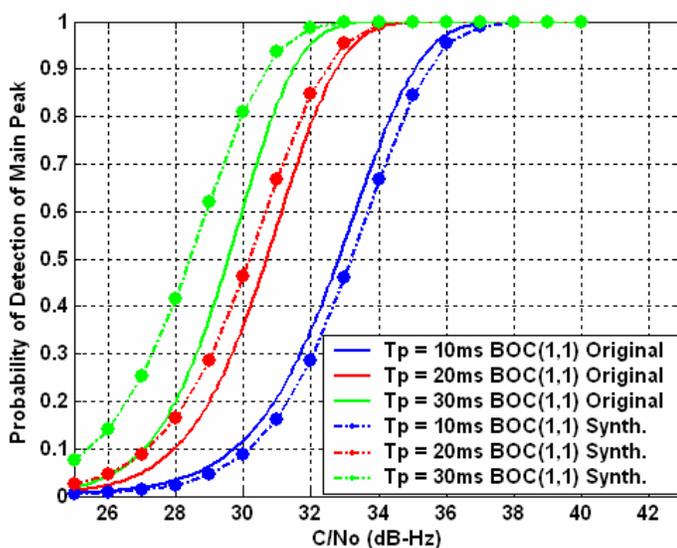


**Figure 6.28 – Probability of Detection for Proposed Galileo L1F Signal Assuming a Coherent Integration Time of 4 ms, 15 and 50 Non-Coherent Summations, no Code Delay/Doppler Uncertainty, and a Probability of False Alarm of  $10^{-3}$**

It is well known that the probability of detection influences the mean acquisition time (Holmes 1982). However, the 0.5 dB equivalent  $C/N_0$  degradation is not expected to significantly degrade this time.

Figure 6.29 shows the probability of detection of the sBOC(1,1) autocorrelation main peak for different integration times, assuming no non-coherent summations. It can be

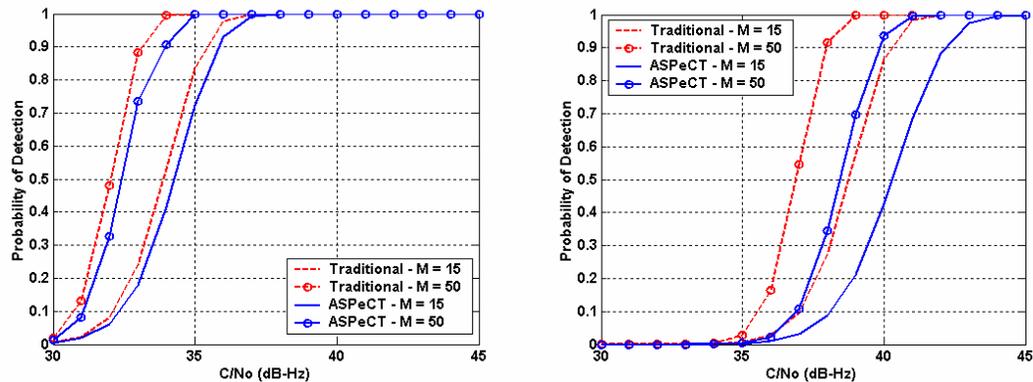
seen that for longer coherent integration times (greater than 20 ms), the proposed acquisition scheme would outperform the traditional acquisition scheme. This phenomenon derives mainly from the fact that, while the ratio of both test criteria equals 1, the noise component in the test criterion (Equation (6.27)) is not purely positive, as is the case in the classical acquisition criterion, as noted above. This can be interesting for acquisition on data-less channels where the coherent integration time is unlimited *a-priori*. However, for applications on Galileo L1F, the maximum coherent integration is limited to 4 ms, even on the pilot channel, due to the presence of the synchronization sequence (see Section 2.3).



**Figure 6.29 – Probability of Detection for Proposed Galileo L1F Signal Assuming a Coherent Integration Time of 10, 20, 30 ms, No Non-Coherent Summations, no Code Delay/Doppler Uncertainty, and a Probability of False Alarm of  $10^{-3}$**

Figure 6.30 shows acquisition performance when both the code delay and the Doppler uncertainty are considered. The code delay uncertainty was set to 0.1 chips and 0.2 chips (corresponding to code delay bins of 0.2 and 0.4 chips, respectively), while the Doppler

uncertainty was set to 50 Hz (corresponding to a 100 Hz Doppler bin size, which is typical considering a 4 ms coherent integration time). It can be seen that the uncertainty plays an important role in the degradation of ASPeCT's acquisition performance, as compared to the traditional acquisition scheme. Indeed, the greater the error, the greater the value of the BOC/PRN correlation will be and, consequently, the lower  $T_{ASPeCT}$  will become. The degradation is approximately equal to 1 dB for a code delay error of 0.1 chips, and of 2.5 dB for a 0.2 chips (in terms of equivalent  $C/N_0$ ). A 1 dB degradation in the acquisition performance is considered limited in view of the fact that the acquisition scheme using ASPeCT's test criterion will never lead to false acquisition. This degradation is even lower than that undertaken via the SSL correlation process (approximately 2 dB as seen in Section 6.1.3.1). Thus it might still be preferable to use the ASPeCT acquisition scheme rather than the SSL correlation function in terms of acquisition sensitivity, provided that the code delay uncertainty is sufficiently small. Of course, if using the SSL method for sBOC(1,1) signal acquisition, a larger code delay/Doppler bin size would be acceptable.



**Figure 6.30 - Probability of Detection for Proposed Galileo L1F Signal Assuming a Coherent Integration Time of 4 ms, 15 and 50 Non-Coherent Summations, Code Delay Uncertainty of 0.1 (Left) and 0.2 (Right) Chips, a Doppler Uncertainty of 50 Hz and a Probability of False Alarm of  $10^{-3}$**

## **CHAPTER 7**

# **CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK**

This chapter presents the conclusions from the research results obtain in the previous chapters and draws recommendations for future work.

### **7.1 Conclusions**

The purpose of this thesis was to investigate and assess, from a pure tracking point-of-view, the impact of the future Galileo L1F signal on the accuracy, reliability and sensitivity of code and carrier phase measurements. This assessment was done with respect to the only currently available signal, namely the GPS C/A signal, to gain a practical appreciation of the improvements promised by Galileo L1F to the general user. Two main characteristics of this new signal were studied: (1) the availability of a pilot channel in quadrature with a traditional data channel carrying the navigation message; and (2) the use of a new sBOC(1,1) modulation.

The pilot channel was shown to be an extremely important characteristic for improving the overall behaviour of both the PLL and DLL. The main asset for a PLL running on a

pilot channel is its use of more efficient phase discriminators than those used on a data channel since they are not constrained to be insensitive to data bit transitions. Therefore, they can provide significant gains in performance. The two pilot discriminators studied (coherent and extended arctangent) offer two improvements with respect to the data channel. First, they possess a wider stability domain ( $\pm\pi$  radians, instead of  $\pm\pi/2$  radians) and a linear tracking domain that ensures better resistance to dynamics and only an integer number of cycle slips, which is of major importance as far as carrier phase ambiguity resolution is concerned. Second, they have a higher PLL tracking sensitivity by approximately at least 5 to 6 dBs (in equivalent  $C/N_0$ ). The coherent discriminator offers optimal tracking in terms of mitigation of Gaussian noise; in addition, its performance is independent of the integration time duration. The extended arctangent discriminator has potentially the widest linear tracking domain, but has degraded performance in the case of low SNR values.

The pilot channel also allows long coherent integration durations. This can be of great help since this would actually increase PLL tracking sensitivity. However, it has been seen that the use of a long coherent integration time has to be calibrated according to the receiver FTS and the expected user motion. Consequently, in certain cases, the use of long coherent integration times might degrade phase tracking. Moreover, since phase tracking is also used to aid code tracking, it might be justified to have short coherent integration in order to better model signal dynamics during longer DLL coherent integration to limit correlation losses due to a large Doppler error. If it is decided to conduct only short coherent integration on the PLL, the use of the coherent discriminator

is recommended since its performance is not dependent upon the integration time. If long coherent integration is carried out, the use of the extended arctangent is then preferable due to its wide linear tracking domain that provides strong resistance to high dynamics.

For the above reasons, it is likely that Galileo L1F phase tracking will be carried out on the pilot channel. However, the data channel can still be used through a data/pilot discriminator combination in order to further mitigate phase measurement noise. Although the data channel cannot be used at a very low SNR due to its limited sensitivity, it still helps to provide more accurate phase measurements for precise positioning users under normal line-of-sight conditions as shown by the two new proposed data/pilot combined phase tracking techniques. Note also that the proposed data/pilot phase tracking architecture has robustness to dynamics equivalent to the use of pilot channel-only tracking.

The use of a pilot channel for Galileo L1F code delay tracking allows long coherent integration time duration and thus, greater sensitivity. However, although the dynamics could be a limiting factor for the choice of the integration time on the PLL, this is less the case for the DLL as long as carrier aiding (through the PLL or other sensors) is provided. Indeed, if the frequency error is accurately modeled, the correlation losses for long coherent integration should be minimal. Consequently, potential gains increase the sensitivity and accuracy of code tracking. Using a 1 s coherent integration duration, the Galileo L1F code tracking threshold could be close to a  $C/N_0$  of 5 dB-Hz.

The use of long coherent integration also allows the use of powerful multipath-mitigation techniques that inherently entail a high inherent correlation loss, as proposed in Section 5.4. Combined with traditional tracking, this new code tracking technique can be accurate and robust. If long integrations are strictly not possible, it is then better to use non-coherent integration to still have a significant post-correlation SNR value.

The use of an sBOC(1,1) modulation for Galileo L1F was studied in Chapter 5. It was seen to be advantageous for code tracking in two respects. Firstly, it provides a code tracking accuracy 2 to 3 dB better than currently available from the GPS C/A code in terms of Gaussian noise for the same loop settings as well as improved multipath mitigation. Secondly, common discriminators (EMLP and DP) exhibit a more favourable output shape for narrow correlator spacing. These two factors compensate for the reduced stability domain offered by sBOC(1,1) discriminators. However, an error greater than 0.33 chips would inevitably result in a loss of lock or in ambiguous tracking.

It was shown in Chapter 6 that the main problem faced by Galileo L1F code tracking is the presence of high autocorrelation side-peaks within the range  $\pm 1$  chip. This could result in a false acquisition or biased tracking, either of which could be extremely dangerous since they would strongly degrade the final position solution if not corrected. An innovative tracking solution dedicated to sBOC(1,1) signals (but that could be extended to any sBOC(n,n) modulation as shown by Julien *et al.* (2004c)), namely ASPeCT, was introduced for this purpose. It is based on a synthesized correlation function that removes the inherent sBOC(1,1) autocorrelation side-peaks. By doing so, it was shown to fully mitigate the tracking bias threat, while still providing a tracking

accuracy equivalent to traditional sBOC(1,1) tracking under Gaussian noise and multipath. Moreover, it does not require more correlator power compared to other proposed techniques to cancel the tracking ambiguity problem. An unambiguous acquisition technique, derived from ASPeCT's synthesized correlation function, was also introduced and its good performance (equivalent to traditional sBOC(1,1) signal acquisition) verified. It is thus an excellent tool to achieve unambiguous acquisition and is complementary with ASPeCT tracking (that cannot compensate for a false acquisition).

## **7.2 Recommendations for Future Work**

Within the framework of this thesis it was necessary to anticipate some limitations due to the tools used, or the information available, and to formulate compensatory strategies. Regarding the PLL tracking sensitivity, the study was limited by the lack of model of the impact of the oscillator phase noise on phase tracking (coming from natural frequency noise and vibrations) for high  $B_L T_I$  values (typically greater than 0.1). Consequently, it would be beneficial to study and assess its contribution on the overall loop performance when long coherent integrations are used.

It would also be valuable to assess the performance of an FLL, normally used to compensate for the PLL-limited sensitivity, on a pilot channel in order to quantify its level of performance with respect to the PLL.

The proposed PLL data/pilot implementation and the DLL HRC/traditional tracking implementation require a consolidation. Indeed, although their concepts have been

thoroughly investigated, they need some adjustments to reach optimal performance (especially in the choice of the discriminator output filter). More advanced testing under diverse environments would be valuable for this analysis.

The results shown using the traditional and ASPeCT code tracking methods were built on the assumption of an sBOC(1,1) modulation. As explained in the introductory chapter, the choice of the Galileo L1F signal modulation might be different, although close to an sBOC(1,1). It is then expected that the main conclusions should still hold. However, it is possible that some adjustments in the formulation of ASPeCT's synthesized correlation function might be required to ensure its unambiguous tracking.

Finally, although the simulation tools used in this thesis were designed to be as close as possible to actual conditions, it would be valuable to test the various algorithms proposed on real data. The first Galileo satellites will be launched at the end of 2005, which will be a good opportunity to validate the methods proposed in this thesis in different environments. Likewise, it is expected that hardware Galileo simulators will become commercially available by the end of 2005.

## REFERENCES

- Allan, D.W., N. Ashby, and C.C. Hodge (1997), *The Science of Time-Keeping*, Application Note 1289, Hewlett Packard.
- Alves, P. (2001), *The effect of Galileo on Carrier Phase Ambiguity Resolution*, Proceedings of the US Institute of Navigation GPS (Salt Lake City, UT, USA, Sept. 11-14), pp. 2086-2095.
- Axelrad, P., and R.G. Brown (1997), *GPS Navigation Algorithms in Global Positioning System: Theory and Applications Volume I*, Progress in Astronautics and Aeronautics Volume 164, AIAA, pp. 409-434.
- Bastide, F., O. Julien, C. Macabiau, B. Roturier (2002), *Analysis of L5/E5 Acquisition, Tracking and Data Demodulation Thresholds*, Proceedings of the US Institute of Navigation GPS (Portland, OR, USA, Sept. 24-27), pp. 2196-2207.
- Bastide, F. (2004), *Analysis of the Feasibility and Interests of Galileo E5a/E5b and GPS L5 Signals for Use with Civil Aviation*, Ph.D. Thesis, Institut National Polytechnique de Toulouse.
- Betz, J.W. (2002), *Binary Offset Carrier Modulations for Radionavigation*, Navigation: Journal of the Institute of Navigation, Vol. 48, Number 4, pp. 227-246.
- Betz, J., and D. B. Goldstein (2002), *Candidate Design for an Additional Civil Signal in GPS Spectral Bands*, Proceedings of the US Institute of Navigation NTM, (San Diego, CA, USA, Jan.28-30), pp. 622-631.
- Bradley, J., D. Guenter, J.A. Nixon, R.D. Sexton (2000), *LAAS Concept of Operations*, Proceedings of the US Institute of Navigation NTM (Anaheim, CA, USA, Jan. 26-28), pp. 200-209.
- Braasch, M. (1997), *Multipath Effects in Global Positioning System: Theory and Applications Volume I*, Progress in Astronautics and Aeronautics Volume 164, AIAA, pp. 547-568.
- Brenner, M., R. Reuters, and B. Schipper (1998), *GPS Landing System Multipath Evaluation Techniques and Results*, Proceedings of the US Institute of Navigation GPS (Nashville, TN, USA, Sept. 15-18), pp. 999-1008.

- Brown, R.G., and P.Y.C. Hwang (1992), *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley & Sons, Inc., Second Edition.
- Chansarkar, M.M., and L. Garin (2000), *Acquisition of GPS Signals at Very Low Signal to Noise Ratios*, Proceedings of the US Institute of Navigation NTM (Anaheim, CA, USA, Jan. 26-28), pp. 731-737.
- Clinton, W. (2000), *Improving the Civilian Global Positioning System (GPS)*, White House Fact Sheet, May 1.
- Coster, A., J. Foster, P. Erickson (2003), *Monitoring the Ionosphere with GPS*, GPS World, May 2003.
- Dafesh, P.A., T.M. Nguyen and S. Lazar (1999), *Coherent Adaptive Subcarrier Modulation (CASM) for GPS Modernization*, Proceedings of US the Institute of Navigation NTM (San Diego, CA, USA, Jan. 25-27), pp. 649-660.
- Dao, D.T.H. (2004), *Simulating GNSS Observables and Errors*, Lead Discussion for ENGO 625 Course, University of Calgary.
- Dong, L. (2003), *IF GPS Signal Simulator Development and Verification*, Master's Thesis, Department of Geomatics Engineering, University of Calgary, UCGE Report 20184, available at <http://www.geomatics.ucalgary.ca/links/GradTheses.html>.
- Dong, L., C. Ma, and G. Lachapelle (2004), *Implementation and Verification of a Software-Based IF GPS Signal Simulator*, Proceedings of the US Institute of Navigation NTM (San Diego, Jan 26-28), pp. 378-389.
- Döttling M., A. Jahn, D. Didascalou, and W Wiesbeck (2001), *Two- and Three-Dimensional Ray Tracing Applied to the Land Mobile Satellite (LMS) Propagation Channel*, IEEE Antennas and Propagation Magazine, Vol. 43, No. 6, Dec., pp. 27-37.
- Enge P., and A.J. Van Dierendonck (1997), *Wide Area Augmentation System*, in *Global Positioning System: Theory and Application Volume II*, Progress in Astronautics and Aeronautics Volume 164, AIAA, pp. 117-142.
- European Commission (1999), *GALILEO Involving Europe in a New Generation of Satellite Navigation Services*, COM 54 Final.
- Fine, P., and W. Wilson (1999), *Tracking Algorithm for GPS Offset Carrier Signals*, in Proceedings of the US Institute of Navigation NTM Conference, (San Diego, CA, Jan. 25-27), pp. 671-676.
- Galileo Joint Undertaking (2005), *L1 Band Part of Galileo Signal in Space ICD (SIS ICD)*, 32 pages.
- Garin, L. (2004), Personal discussions.

- Garin, L. (2005), *The "Shaping Correlator", Novel Multipath Mitigation Technique Applicable to GALILEO BOC(1,1) Modulation Waveforms in High Volume Markets*, CD-ROM Proceedings of the European Navigation Conference GNSS (Munich, Germany, July 19-22).
- Godet, J., J.C. de Mateo, P. Erhard, and O. Nouvel (2002), *Assessing the Radio-Frequency Compatibility between GPS and Galileo*, Proceedings of US. Institute of Navigation (Portland, OR, USA, Sept. 24-27), pp. 1260-1269.
- Gold, R. (1967), *Optimal Binary Sequences for Spread Spectrum Multiplexing*, IEEE Transactions on Information Theory, Vol. 13, Issue 4, Oct., pp. 619-621.
- GPS World (2005), *GNSS on Stage*, GPS World, Jan 1.
- Hegarty, C. (1997), *Analytical Derivation of the Maximum Tolerable In-Band Interference Levels for Aviation Applications of GNSS*, Navigation: Journal of the Institute of Navigation, Vol. 44, No. 1, Spring, pp. 25-34.
- Hegarty, C. (1999), *Evaluation of the Proposed Signal Structure for the New Civil GPS Signal at 1176.45 MHz*, WN99W0000034, The MITRE Corporation.
- Hegarty, C., M. Tran and J.W. Betz (2004), *Multipath Performance of the New GNSS Signals*, Proceedings of the US Institute of Navigation NTM (San Diego, CA, USA, Jan.), pp. 333-342.
- Hein, G., J. Godet, J.-L. Issler, J.-C. Martin, P. Erhard, R. Lucas-Rodriguez, and T. Pratt (2002), *Status of Galileo Frequency and Signal Design*, Proceedings of the US Institute of Navigation GPS/GNSS (Portland, OR, USA, Sept. 24-27), pp. 266-277.
- Hein, G., M. Irsigler, J.A. Avila Rodriguez, and T. Pany (2004), *Performance of Galileo L1 Signal Candidates*, CD-ROM Proceedings of the European Navigation Conference GNSS (Rotterdam, The Netherlands, May 16-19), Session 'Galileo System Design', 18 pages.
- Holmes, J.K. (1982), *Coherent Spread Spectrum Systems*, John Wiley and Sons, New York.
- Holmes, J.K. (2000), *Code Tracking Loop Performance Including the Effect of Channel Filtering and Gaussian Interference*, Proceedings of the US Institute of Navigation AM (San Diego, CA, USA, June 26-28), pp. 382-398.
- Hudnut, K., B. Titus, J. Taylor and T. Stansell (2004), *Improving the GPS L1 Signal – GPS III Offers the Opportunity*, Presentation by US Geological Survey, Inter-agency GPS Executive Board and US Air Force, 36 slides.
- Hurst, R.L., and R.E. Knop (1972), *Generation of Random Correlated Normal Variables*, Communications of the ACM, Vol. 15, No. 5, May, pp. 355-357.

- Irisigler, M., and B. Eissfeller (2002), *PLL Tracking Performance in Presence of Oscillator Phase Noise*, GPS Solutions, Vol. 5, No. 4, pp. 45-54.
- Jahn, A., H. Bischl, G. Heiss (1996), *Channel Characterisation for Spread Spectrum Satellite Communication*, Spread, IEEE 4th International Symposium on Spectrum Techniques and Applications Proceedings, Vol. 3, (Sept. 22-25), pp. 1221-1226.
- Jiang, Z. (2004), *Mitigation of Narrow Band Interference on Software Receivers Based on Spectrum Analysis*, Master's Thesis, Department of Geomatics Engineering, University of Calgary, UCGE Report 20202, available at <http://www.geomatics.ucalgary.ca/links/GradTheses.html>.
- Julien, O., P. Alves, M.E. Cannon, and G. Lachapelle (2004a) *Improved Triple-Frequency GPS/Galileo Carrier Phase Ambiguity Resolution Using a Stochastic Ionosphere Modeling*, Proceedings of the US Institute of Navigation NTM (San Diego, CA, USA, Jan. 24-28), pp. 441-452.
- Julien, O., M.E. Cannon, P. Alves, and G. Lachapelle (2004b), *Triple Frequency Ambiguity Resolution Using GPS/Galileo*. European Journal of Navigation, Vol. 2, Number 2, pp. 51-57.
- Julien, O., C. Macabiau, G. Lachapelle, M.E. Cannon, and C. Mongrédien (2004c), *A New Unambiguous BOC(n,n) Signal Tracking Technique*, in CD-ROM Proc. of the European Navigation Conference GNSS (Rotterdam, The Netherlands, May 16-19), Session 'Signal', 12 pages.
- Julien, O., G. Lachapelle and M.E. Cannon (2004d), *A New Multipath and Noise Mitigation Technique using Data/Data-less Navigation Signals*, Proceedings of the US Institute of Navigation GNSS (Long Beach, CA, USA, Sept. 24-27), pp. 8-19.
- Klobuchar, J.A. (1997), *Ionospheric Effects on GPS in Global Positioning System: Theory and Applications Volume I*, Progress in Astronautics and Aeronautics Volume 164, AIAA, pp. 485-516.
- Klukas, R., O. Julien, L. Dong, E. Cannon, and G. Lachapelle (2004), *Effects of Building Materials on UHF Ranging Signals*, GPS Solutions, Vol. 8, No. 1, pp. 1-8.
- Kovach, K., and K. Van Dyke (1997), *GPS in Ten Years*, Proceedings of the US Institute of Navigation GPS Conference (Kansas City, MS, USA, Sept. 16-19), pp. 1251-1259.
- Kuusniemi, H., G. Lachapelle, and J. Takala (2004), *Reliability in Personal Positioning*, CD-ROM Proceedings of the European Navigation Conference GNSS, (Rotterdam, The Netherlands, May 16-19), Session 'Land-Based Applications I', 12 pages.
- Lachapelle, G. (2004a), *NAVSTAR GPS: Theory and Applications*, ENGO 625 Lecture Notes, Department of Geomatics Engineering, University of Calgary.

- Lachapelle, G. (2004b), *GNSS Indoor Location Technologies*, CD-ROM Proceedings of the 2004 International Symposium on GNSS/GPS (Sydney, Australia, Dec. 6-8), 15 pages.
- Lin, V.S., P.A. Dafesh, A. Wu, and C.R. Cahn (2003), *Study of the Impact of False Lock Points on Subcarrier Modulated Ranging Signals and Recommended Mitigation Approaches*, in Proceedings of US Institute of Navigation AM Conference, (Albuquerque, NM, USA, June 23-25), pp. 156-165.
- Ma, C., G. Jee, G. MacGougan, G. Lachapelle, S. Bloebaum, G. Cox, L. Garin and J. Shewfelt (2001), *GPS Signal Degradation Modeling*, Proceedings of the US Institute of Navigation GPS, (Session C2, Salt Lake City, UT, USA, September 11-14), pp. 882-893.
- Ma, C., G. Lachapelle, and M.E. Cannon (2004), *Implementation of a Software GPS Receiver*, Proceedings of the US Institute of Navigation GNSS (Long Beach, CA, USA, Sept. 22-25), pp. 956-970.
- Macabiau, C. (2001), *Communications par Spectre Etalé*, Lecture Note, ENAC.
- Macabiau, C., O. Julien, and E. Chatre (2002), *Use of Multicorrelator Techniques for Interference Detection*, Proceedings of the US Institute of Navigation NTM (Long Beach, CA, USA, Jan. 22-24), pp. 353-363.
- Macabiau, C., L. Ries, F. Bastide, J.-L. Issler (2003), *GPS L5 Receiver Implementaion Issues*, Proceedings of the US Institute of Navigation GPS/GNSS (Portland, OR, USA, Sept. 9-12), pp. 153-164.
- MacGougan, G., G. Lachapelle, R. Klukas, K. Siu, L. Garin, J. Shewfelt, and G. Cox (2002), *Performance Analysis of A Stand-Alone High Sensitivity Receiver*, GPS Solutions, Springer Verlag, Vol. 6, No. 3, pp. 179-195.
- MacGougan, G.D. (2003), *High Sensitivity GPS Performance Analysis in Degraded Signal Environment*, Master's thesis, Department of Geomatics Engineering, University of Calgary, UCGE Report 20176, available at <http://www.geomatics.ucalgary.ca/links/GradTheses.html>.
- Martin, N., V. Leblond, G. Guillotel, and V. Heiries (2003), *BOC(x,y) Signal Acquisition Techniques and Performances*, in Proceedings of the US Institute of Navigation GPS/GNSS Conference, (Portland, OR, USA, Sept. 9-12), pp. 188-198.
- McDonald, K., J. Armor, G. Beutler, C. Boesentburg, J. Canny, W. Lechner, and S. Pace (1997), *The Civil and Military Issues Facing GPS and GNSS*, Proceedings of the US Institute of Navigation GPS Conference (Kansas City, MS, USA, Sept. 16-19), pp. 3-20.

- McGraw, G, and M. Braasch (1999), *GNSS Multipath Mitigation Using Gated and High Resolution Correlator Concept*, Proceedings of the US Institute of Navigation NTM (San Diego, CA, Jan. 25-27), pp. 333-342.
- Misra, P., and P. Enge (2001), *Global Positioning System: Signals, Measurements and Performance*, Ganga-Jamuna Press.
- Morrissey, T.N, K.W. Shallberg, A.J. Van Dierendonck, T. Kim, and M.J. Nicholson (2002), *GPS Receiver Performance Characterization under Simulated Ionospheric Scintillation Environments II*, Proceedings of the US Institute of Navigation NTM (San Diego, CA, USA, Jan. 28-30), pp. 682-693.
- NovAtel (2004), <http://www.novatel.ca> (July 2005).
- Nunes, F.D., F.M.G. Sousa, J.M.N. Leitão (2004), *Multipath Mitigation Techniques for BOC Signals Using Gated Functions*, CD-ROM Proceedings of Navitech 2004 (Noordwijk, The Netherlands, Dec. 8-10), 8 pages.
- Olynik, M (2003), *Temporal Characteristics of GPS Error Sources and Their Impact on Relative Positioning*, Master's Thesis, Department of Geomatics Engineering, University of Calgary, UCGE Report 20162, available at <http://www.geomatics.ucalgary.ca/links/GradTheses.html>.
- Onidi, O. (2002), *Launching Galileo*, Galileo's World, Nov. 1.
- Papoulis, A. (1991), *Probability, Random Variables and Stochastic Processes*, Third Edition, McGraw Hill International Editions.
- Parkinson, B.W. (1997), *Introduction and Heritage of NAVSTAR, the Global Positioning System in Global Positioning System: Theory and Applications Volume I*, Progress in Astronautics and Aeronautics Volume 164, AIAA, pp. 3-28.
- Peterson, R.L., R.E. Ziemer, and D. E. Borth (1995), *Introduction to Spread Spectrum Communications*, Prentice Hall.
- Pratt, A.R. (2002), *A New Class of Spreading Codes Exhibiting Low Cross-Correlation Properties*, Proceedings of the US Institute of Navigation GPS (Portland, OR, USA, Sept. 24-27), pp. 1554-1564.
- PriceWaterHouseCoopers (2001), *Inception Study to Support the Development of a Business Plan for the GALILEO Program*, Report TREN/B5/23-2001, Prepared for the European Community under contract ETU-B6613-E4-PWC-2001-S12.321824.
- Raghavan, S.H., J.K. Holmes, S. Lazar, and M. Bottjer (1997), *Tricode Hexaphase Modulation for GPS*, Proceedings of the US Institute of Navigation GPS (Kansas City, MI, USA, Sept. 16-19), pp. 385-397.

- Ray, J. (2002), *Advanced Receiver Design*, ENGO 699.73 Lecture Notes, The University of Calgary.
- Ries, L., L. Lestarquit, E. Armangou-Miret, F. Legrand, W. Vigneau, C. Bourga, P. Erhard, and J.-L. Issler (2002), *A Software Simulation Tool for GNSS2 BOC Signal Analysis*, Proceedings of the US Institute of Navigation GPS (Portland, OR, USA, Sept. 24-27), pp. 2225-2239.
- Ries, L., F. Legrand, L. Lestarquit, W. Vigneau, and J.-L. Issler (2003), *Tracking and Multipath Performance of Assessments of BOC Signals Using a Bit-Level Signal Processing Simulator*, Proceedings of the US Institute of Navigation GPS/GNSS Conference (Portland, OR, Sept. 9-12), pp. 1996-2010
- Rooney, E., J. Paffet, M. Unwin, and A. Garutti (2004), *Status of GSTB-V2/A Satellite Programme*, Proceedings of the US Institute of Navigation GNSS (Long Beach, CA, USA, Sept. 21-24), pp. 650-653.
- Simsky, A., and J.M. Sleewaegen (2004), *C/A Code Bias in High-End Receivers*, Proceedings of the European Navigation Conference GNSS (Rotterdam, The Netherlands, May 17-19), 7 pages.
- Spilker, J.J. (1997a), *GPS Signal Structure and Theoretical Performance in Global Positioning System: Theory and Applications Volume I*, Progress in Astronautics and Aeronautics Volume 164, AIAA, pp. 57-120.
- Spilker, J.J. (1997b), *Fundamentals of Signal Tracking Theory in Global Positioning System: Theory and Applications Volume I*, Progress in Astronautics and Aeronautics Volume 164, AIAA, pp. 245-328.
- Spilker J.J. (1997c), *GPS Navigation Data in Global Positioning System: Theory and Applications Volume I*, Progress in Astronautics and Aeronautics Volume 164, AIAA, pp. 121-176.
- Steingass, A., A. Lehner, F. Pérez-Fontán, E. Kubista, M.J. Martín, and B. Arbesser-Rastburg (2004), *The High Resolution Aeronautical Multipath Navigation Channel*, Proceedings of the US Institute of Navigation NTM (San Diego, CA, USA, Jan. 24-28), pp. 793-804.
- Stephens S.A., and J. B. Thomas (1995), *Controlled-Root Formulation for Digital Phase-Locked Loops*, in IEEE Transactions on Aerospace and Electronic Systems, Vol. 31, No. 1, pp. 78-95.
- Sullivan, D.B., S.W. Allan, D.A. Howe, and F.L. Walls (1990), *Characterization of Clocks and Oscillators*, National Institute of Standards and Technology, US Department of Commerce, Technical Note 1337.

- Toran-Marti F., and J. Ventura-Traveset (2004), *The ESA EGNOS Project: The First Step of the European Contribution to the Global Navigation Satellite System (GNSS)*, NAVIGARE Conference (Winterthur, Switzerland, June 18).
- US Office of Science and Technology Policy (2004), *U.S. Space-Based Positioning, Navigation and Timing Policy – Fact Sheet*.
- Tran, M, and C. Hegarty (2002), *Receiver Algorithms for the New Civil GPS Signals*, Proceedings of the US Institute of Navigation NTM (San Diego, CA, USA, Jan. 28-30), pp. 778-789.
- Van Dierendonck, A.J. (1997), *GPS Receivers in Global Positioning System: Theory and Application Volume I*, Progress in Astronautics and Aeronautics Volume 164, AIAA, pp. 329-408.
- Van Dierendonck, A.J., and J.J. Spilker (1999), *Proposed Civil GPS Signal 1176.45 MHz: In-Phase/Quadrature Codes at 10.23 MHz Chip Rate*, Proceedings of the US Institute of Navigation AM (Cambridge, MA, USA, June 28-30), pp. 761-770.
- Ward, P. (1996), *Satellite Signal Acquisition and Tracking in Understanding GPS: Principles and Applications*, Artech House Inc.
- Ward, P. (2004), *A Design Technique to Remove the Correlation Ambiguity in Binary Offset Carrier (BOC) Spread Spectrum Signals (Revised Version)*, Proceedings of US Institute of Navigation NTM Conference, (San Diego, CA, USA, Jan. 26-28), pp. 886-896.
- Watson, R. (2005), *High-Sensitivity GPS L1 Signal Analysis for Indoor Channel Modeling*, Master's thesis, Department of Geomatics Engineering, University of Calgary, UCGE Report 20216, available at <http://www.geomatics.ucalgary.ca/links/GradTheses.html>.
- Weill, L. (1997), *GPS Multipath Mitigation by Means of Correlator Reference Waveform Design*, Proceedings of the US Institute of Navigation NTM (Santa Monica, CA, Jan. 14-16), pp. 197-206.
- Winkel, J.O. (2003), *Modeling and Simulating GNSS Signal Structures and Receivers*, PhD Thesis, University FAF Munich, Neubiberg.

## APPENDIX A

### CORRELATOR OUTPUT NOISE POWER AND CORRELATION

The goal of this appendix is to assess the correlation and the power of the noise at the output of the correlator. As an example, only the noise from the in-phase correlator is calculated herein. The same calculation is valid for the noise of the quadra-phase correlator

Using the notations introduced in Chapter 2, the correlation noise can be written as:

$$n_I(t) = (n_{FE}(t)c(t)\cos(2\pi f_{IF}t)) * h_{ID}(t) \quad (\text{A.1})$$

where  $*$  represents the convolution operator,

$n_{FE}(t)$  is the incoming noise after front-end filtering and equals  $n(t) * h(t)$ , and

In order to simplify the expression, it is convenient to introduce

$$n_b(t) = n_{FE}(t)c(t)\cos(2\pi f_{IF}t) \quad (\text{A.2})$$

Assume two correlators using the code replica with different delays (with a relative delay  $\tau$ ). The correlation of these two noise representations is obtained through:

$$R_{n_b}(x, \tau) = E[n_b(t, u)n_b(t-x, u-\tau)] = E \begin{bmatrix} n_{FE}(t)c(t+u)\cos(2\pi f_{IF}t)n_{FE}(t-x) \\ \times c(t-x+u-\tau)\cos(2\pi f_{IF}(t-x)) \end{bmatrix} \quad (\text{A.3})$$

$$R_{n_b}(x, \tau) = \frac{1}{2} \cos(2\pi f_{IF}x) R_{n_{FE}}(x) R_c(x) * \delta(x+\tau) \quad (\text{A.4})$$

Consequently, using the Wiener-Kirchine formula, the PSD of the correlator noise can be calculated as:

$$S_{n_b}(f, \tau) = FFT(R_{n_b}(x, \tau)) = \frac{1}{4} (\delta(f - f_{IF}) + \delta(f + f_{IF})) * (S_{n_{FE}}(f) * G(f) e^{-2i\pi f\tau}) \quad (\text{A.5})$$

Assuming a front-end filter with a one-sided bandwidth of  $B$  Hz and centered around  $f_{IF}$ , the PSD of the incoming noise can be modeled as:

$$S_{n_{FE}}(f) = \frac{N_0}{2} |H(f)|^2 * (\delta(f - f_{IF}) + \delta(f + f_{IF})) \quad (\text{A.6})$$

Consequently, considering that the components situated at  $\pm 2f_{IF}$  will be filtered out by the I&D filter

$$S_{n_b}(f, \tau) = \frac{N_0}{4} (|H(f)|^2 * G(f) e^{-2i\pi f\tau}) \quad (\text{A.7})$$

The I&D filter is just an integrator, and can be represented by:

$$h_{ID}(t) = \frac{1}{T_I} \text{rect}\left(\frac{t}{T_I}\right) \quad (\text{A.8})$$

Spectrally, its FFT is:

$$H_{ID}(f) = \frac{\sin(\pi f T_I)}{\pi f T_I} \quad (\text{A.9})$$

The effect of the filter on the noise can be represented by:

$$S_{n_I}(f, \tau) = |H_{ID}(f)|^2 S_{n_b}(f, \tau). \quad (\text{A.10})$$

The I&D filter can be considered as very narrow compared to the PSD of  $n_b$ .

Consequently, it can be written that:

$$S_{n_I}(f, \tau) = |H_{ID}(f)|^2 S_{n_b}(0, \tau). \quad (\text{A.11})$$

$S_{n_b}(0, \tau)$  corresponds to the convolution in 0 of Equation (A.8), so:

$$S_{n_b}(0, \tau) = \frac{N_0}{4} \int_{-\infty}^{+\infty} |H(f)|^2 G(f) e^{-2i\pi f \tau} df \quad (\text{A.12})$$

Finally,

$$S_{n_I}(f, \tau) = \frac{N_0}{4} |H_{ID}(f)|^2 \int_{-\infty}^{+\infty} |H(f_1)|^2 G(f_1) e^{-2i\pi f_1 \tau} df_1. \quad (\text{A.13})$$

The correlator noise power is then given by:

$$P_{n_I}(\tau) = \int_{-\infty}^{+\infty} S_{n_I}(f, \tau) df = \frac{N_0}{4} \int_{-\infty}^{+\infty} |H(f_1)|^2 G(f_1) e^{-2i\pi f_1 \tau} df_1 \int_{-\infty}^{+\infty} |H_{ID}(f)|^2 df \quad (\text{A.14})$$

$$P_{n_I}(\tau) = \frac{N_0}{4T_I} \int_{-\infty}^{+\infty} |H(f)|^2 G(f) e^{-2i\pi f \tau} df \quad (\text{A.15})$$

It can be concluded that the power of each correlator noise equals:

$$P_{n_I} = P_{n_I}(0) = \frac{N_0}{4T_I} \int_{-B}^{+B} |H(f)|^2 G(f) df = \frac{N_0}{4T_I} \tilde{R}_{F,c}(0) \quad (\text{A.16})$$

The correlation of the noise of two correlators using two different code sequence replicas separated by  $\tau$  can then be extrapolated as:

$$R_{n_I}(\tau) = P_{n_I}(\tau) = \frac{N_0}{4T_I} \int_{-\infty}^{+\infty} |H(f)|^2 G(f) e^{2i\pi f\tau} df = \frac{N_0}{4T_I} \tilde{R}_{F,c}(\tau) \quad (\text{A.17})$$

## APPENDIX B

### PLL FREQUENCY ERROR VARIANCE

This appendix is meant to evaluate the frequency error variance of the PLL. This is of interest since the PLL frequency error estimation is usually fed to the DLL for carrier-phase aiding.

If we take the usual model of a PLL, but design it as a frequency loop, it gives the following equation, as represented in its Laplace form:

$$\hat{f}(s) = KH_{LF}(s)D(s) \quad (\text{B.1})$$

where

$f$  represents the frequency error

$K$  is the loop filter gain

$H_{LF}$  is the loop filter, and

$D$  is the discriminator output

The discriminator output is assumed unbiased around  $\varepsilon_\varphi = 0$  and consequently it can be modeled for a small phase error as:

$$D(s) = K_D \varepsilon_\varphi(s) + N(s) \quad (\text{B.2})$$

where

$\varepsilon_\varphi$  represents the input phase error

$K_D$  is the discriminator gain, and

$N$  is the discriminator output noise.

Since the frequency error can be considered as the derivative of the phase error, this gives, using the Laplace transform:

$$\varepsilon_\varphi(s) = \frac{\varepsilon_f(s)}{s} \quad (\text{B.3})$$

It can then be deduced that:

$$D(s) = K_D \frac{\varepsilon_f(s)}{s} + N(s) \quad (\text{B.4})$$

Using Equation (B.1) and Equation (B.4), one obtains:

$$\hat{f}(s) = KK_D \frac{F(s)}{s} \left( \varepsilon_f(s) + \frac{sN(s)}{K_D} \right). \quad (\text{B.5})$$

Since  $\varepsilon_f = f - \hat{f}$ , one can write:

$$\varepsilon_f(s) \left( 1 + \frac{KK_D H_{PLL}(s)}{s} \right) = f_I - KK_D H_{PLL}(s) \left( \frac{sN(s)}{K_D} \right). \quad (\text{B.6})$$

This leads to:

$$\varepsilon_f(s) = \frac{f_I}{\left( 1 + \frac{KK_D H_{PLL}(s)}{s} \right)} - \frac{KK_D H_{PLL}(s)}{\left( 1 + \frac{KK_D H_{PLL}(s)}{s} \right)} \left( \frac{sN(s)}{K_D} \right). \quad (\text{B.7})$$

This is the closed loop expression of the frequency error. The effect of the noise is only contained in the second term that will be the object of the following analysis. For convenience,  $H_L(s)$  is defined as the closed loop transfer function so that:

$$\varepsilon_f(s) = (1 - H_L(s))f_i - H_L(s) \left( \frac{sN(s)}{K_D} \right) \quad (\text{B.8})$$

$$\text{with } H_{PLL}(s) = \frac{\frac{KK_D H_{PLL}(s)}{s}}{1 + \frac{KK_D H_{PLL}(s)}{s}}$$

The noise power can then be deduced through the following equation:

$$\sigma_{\varepsilon_f}^2 = \int_{-\infty}^{+\infty} \varepsilon_f(s) \varepsilon_f^*(s) ds \quad (\text{B.9})$$

and  $\varepsilon_f(s) \varepsilon_f^*(s)$  can be decomposed as:

$$\varepsilon_f(s) \varepsilon_f^*(s) = \frac{1}{K_D^2} |H_{PLL}(s)|^2 s s^* (N(s) N^*(s)) \quad (\text{B.10})$$

It can be reasonably assumed that the bandwidth of the loop filter  $H_{PLL}(s)$  is far smaller than the bandwidth of  $N(s)$ . In the frequency domain, this gives:

$$\varepsilon_f(f) \varepsilon_f^*(f) = (2\pi)^2 \frac{1}{K_D^2} |H_L(2i\pi f)|^2 f^2 S_N(0) \quad (\text{B.11})$$

and

$$\sigma_{\varepsilon_f}^2 = \frac{4\pi^2}{K_D^2} S_N(0) \int_{-\infty}^{+\infty} |H_L(2i\pi f)|^2 f^2 df. \quad (\text{B.12})$$

This expression can be approximated as

$$\sigma_{\varepsilon_f}^2 = \frac{\kappa 8\pi^2 B_L^3 S_N(0)}{3K_D^2} \quad (\text{B.13})$$

where the coefficient  $\kappa$  is defined as:

$$\kappa = \frac{\int_{-\infty}^{+\infty} |H_L(2i\pi f)|^2 f^2 df}{\int_{-B_L}^{B_L} f^2 df} = \frac{\int_{-\infty}^{+\infty} |H_L(2i\pi f)|^2 f^2 df}{\frac{2}{3} B_L^3} \quad (\text{B.14})$$

The value  $\kappa$  is dependent upon  $B_L$ .

Since  $N$  represents the noise of the PLL discriminator and it is known that for a usual PLL that:

$$\sigma_{\varepsilon_\varphi}^2 = \frac{2B_L S_N(0)}{K_D^2} \quad (\text{B.15})$$

One obtains

$$\sigma_{\varepsilon_f}^2 = \frac{\kappa(B_L) 4\pi^2 B_L^2}{3} \sigma_{\varepsilon_\varphi}^2 \quad (\text{B.16})$$

## APPENDIX C

### DOT-PRODUCT DISCRIMINATOR VARIANCE IN THE PRESENCE OF GAUSSIAN NOISE

#### C.1 Background

Following the notation introduced in Chapter 2 and Chapter 5, the traditional DP discriminator is given by:

$$D_{DP} = (I_E - I_L)I_P + (Q_E - Q_L)Q_P \quad (\text{C.1})$$

Assuming that Gaussian noise is the only disturbance entering the receiver front-end, the above can be re-written as:

$$D_{DP} = \left( \begin{array}{l} \left( \sqrt{\frac{P}{2}} \cos(\varepsilon_\phi) \left( \tilde{R}\left(\varepsilon_\tau + \frac{\delta}{2}\right) - \tilde{R}\left(\varepsilon_\tau - \frac{\delta}{2}\right) \right) \right) \left( \sqrt{\frac{P}{2}} \cos(\varepsilon_\phi) \tilde{R}(\varepsilon_\tau) + n_{I_P} \right) \\ + n_{I_E} - n_{I_L} \\ \left( \sqrt{\frac{P}{2}} \sin(\varepsilon_\phi) \left( \tilde{R}\left(\varepsilon_\tau + \frac{\delta}{2}\right) - \tilde{R}\left(\varepsilon_\tau - \frac{\delta}{2}\right) \right) \right) \left( \sqrt{\frac{P}{2}} \sin(\varepsilon_\phi) \tilde{R}(\varepsilon_\tau) + n_{Q_P} \right) \\ + n_{Q_E} - n_{Q_L} \end{array} \right) \quad (\text{C.2})$$

and

$$D_{DP} = \left( \begin{array}{l} \frac{P}{2} \left( \tilde{R} \left( \varepsilon_\tau + \frac{\delta}{2} \right) - \tilde{R} \left( \varepsilon_\tau - \frac{\delta}{2} \right) \right) \tilde{R}(\varepsilon_\tau) \\ + \sqrt{\frac{P}{2}} \cos(\varepsilon_\phi) \left( \tilde{R} \left( \varepsilon_\tau + \frac{\delta}{2} \right) - \tilde{R} \left( \varepsilon_\tau - \frac{\delta}{2} \right) \right) n_{I_P} \\ + \sqrt{\frac{P}{2}} \cos(\varepsilon_\phi) \tilde{R}(\varepsilon_\tau) (n_{I_E} - n_{I_L}) + (n_{I_E} - n_{I_L}) n_{I_P} \\ + \sqrt{\frac{P}{2}} \sin(\varepsilon_\phi) \left( \tilde{R} \left( \varepsilon_\tau + \frac{\delta}{2} \right) - \tilde{R} \left( \varepsilon_\tau - \frac{\delta}{2} \right) \right) n_{Q_P} \\ + \sqrt{\frac{P}{2}} \sin(\varepsilon_\phi) \tilde{R}(\varepsilon_\tau) (n_{Q_E} - n_{Q_L}) + (n_{Q_E} - n_{Q_L}) n_{Q_P} \end{array} \right) \quad (C.3)$$

Without loss of generality, assuming that the carrier-phase tracking is perfect, we can then simplify the expression as:

$$D_{DP} = \left( \begin{array}{l} \frac{P}{2} \left( \tilde{R} \left( \varepsilon_\tau + \frac{\delta}{2} \right) - \tilde{R} \left( \varepsilon_\tau - \frac{\delta}{2} \right) \right) \tilde{R}(\varepsilon_\tau) \\ + (n_{I_E} - n_{I_L}) n_{I_P} + (n_{Q_E} - n_{Q_L}) n_{Q_P} \\ + \sqrt{\frac{P}{2}} \left( \tilde{R}(\varepsilon_\tau) (n_{I_E} - n_{I_L}) + \left( \tilde{R} \left( \varepsilon_\tau + \frac{\delta}{2} \right) - \tilde{R} \left( \varepsilon_\tau - \frac{\delta}{2} \right) \right) n_{I_P} \right) \end{array} \right) \quad (C.4)$$

We can then divide the expression as follows:

$$D_{DP} = \left( \begin{array}{l} \frac{P}{2} \left( \tilde{R} \left( \varepsilon_\tau + \frac{\delta}{2} \right) - \tilde{R} \left( \varepsilon_\tau - \frac{\delta}{2} \right) \right) \tilde{R}(\varepsilon_\tau) \\ + N_1 + \sqrt{\frac{P}{2}} N_2 \end{array} \right) \quad (C.5)$$

with

$$N_1 = (n_{I_E} - n_{I_L}) n_{I_P} + (n_{Q_E} - n_{Q_L}) n_{Q_P} \quad (C.6)$$

and

$$N_2 = \left( \tilde{R}(\varepsilon_\tau)(n_{I_E} - n_{I_L}) + \left( \tilde{R}\left(\varepsilon_\tau + \frac{\delta}{2}\right) - \tilde{R}\left(\varepsilon_\tau - \frac{\delta}{2}\right) \right) n_{I_P} \right) \quad (\text{C.7})$$

In order to facilitate the calculations and the easier use of notation, it is convenient to introduce the notation extrapolated from Equation (2.15):

$$\tilde{R}_F(x) = \int_{-\infty}^{+\infty} |H(f)|^2 G(f) e^{-2i\pi fx} df \quad (\text{C.8})$$

Let us recall that the correlation value of the signal is given by Equation (2.17):

$$\tilde{R}(x) = \int_{-\infty}^{+\infty} H(f) G(f) e^{-2i\pi fx} df \quad (\text{C.9})$$

## C.2 Loop Gain

According to Equation (4.12), the loop gain is given by:

$$K_{DP} = \left. \frac{dDP}{d\varepsilon_\tau} \right|_{\varepsilon_\tau=0} = \left. \frac{d\left( \frac{P}{2} \left( \tilde{R}\left(\varepsilon_\tau + \frac{\delta}{2}\right) - \tilde{R}\left(\varepsilon_\tau - \frac{\delta}{2}\right) \right) \tilde{R}(\varepsilon_\tau) \right)}{d\varepsilon_\tau} \right|_{\varepsilon_\tau=0} \quad (\text{C.10})$$

which is equivalent to:

$$K_{DP} = \frac{P}{2} \left( \left( \frac{d\tilde{R}\left(\varepsilon_\tau - \frac{\delta}{2}\right)}{d\varepsilon_\tau} - \frac{d\tilde{R}\left(\varepsilon_\tau + \frac{\delta}{2}\right)}{d\varepsilon_\tau} \right) \tilde{R}(\varepsilon_\tau) + \left( \tilde{R}\left(\varepsilon_\tau - \frac{\delta}{2}\right) - \tilde{R}\left(\varepsilon_\tau + \frac{\delta}{2}\right) \right) \frac{d\tilde{R}(\varepsilon_\tau)}{d\varepsilon_\tau} \right) \Bigg|_{\varepsilon_\tau=0} \quad (\text{C.11})$$

Assuming that the front-end filter is symmetric and using Equation (6.9), one gets:

$$\tilde{R}\left(\varepsilon_\tau + \frac{\delta}{2}\right) - \tilde{R}\left(\varepsilon_\tau - \frac{\delta}{2}\right) = \int_{-\infty}^{+\infty} H(f)G(f) \left( e^{-2i\pi f\left(\varepsilon_\tau + \frac{\delta}{2}\right)} - e^{-2i\pi f\left(\varepsilon_\tau - \frac{\delta}{2}\right)} \right) df \quad (\text{C.12})$$

Assuming that  $\varepsilon_\tau \approx 0$ , one can then write:

$$\tilde{R}\left(\varepsilon_\tau + \frac{\delta}{2}\right) - \tilde{R}\left(\varepsilon_\tau - \frac{\delta}{2}\right) = 2i \int_{-\infty}^{+\infty} H(f)G(f) \sin(\pi f \delta) df . \quad (\text{C.13})$$

Since  $G(f)$  is symmetric (PSD of the spreading sequence) and  $\sin(\pi f \delta)$  is asymmetric, then, for  $\varepsilon_\tau \approx 0$ , one can extrapolate as follows:

$$\tilde{R}\left(\varepsilon_\tau + \frac{\delta}{2}\right) - \tilde{R}\left(\varepsilon_\tau - \frac{\delta}{2}\right) = 0 \quad (\text{C.14})$$

Equation (C.11) can be simplified as:

$$K_{DP} = \frac{P}{2} \left( \int_{-B}^{+B} j2\pi f H(f)G(f) \left( e^{j\pi f \delta} - e^{-j\pi f \delta} \right) df \right) \tilde{R}(0) \quad (\text{C.15})$$

and

$$K_{DP} = 2\pi P \left( \int_{-\infty}^{+\infty} f H(f)G(f) \sin(\pi f \delta) df \right) \tilde{R}(0) . \quad (\text{C.16})$$

As shown by Holmes (2000) and as per Equation (5.28):

$$\sigma_{DP, \varepsilon_\tau}^2 = \frac{2B_L(1 - 0.5B_L T_I) S_{N_{DP}}(0)}{K_{DP}^2}$$

In order to be able to compute the DLL tracking error variance, one needs to assess the value of the PSD of the discriminator output noise in  $f = 0$ . To do so, the methodology is to use the well-known formula:

$$S_{N_{DP}}(0) = \int_{-\infty}^{+\infty} R_{n_{DP}}(x) dx \quad (\text{C.17})$$

Consequently, the autocorrelation of the discriminator noise has to be realized. Using Equation (C.5), one can write:

$$R_{n_{DP}}(x) = E \left[ \left( N_1(t) + \sqrt{\frac{P}{2}} N_2(t) \right) \left( N_1(t-x) + \sqrt{\frac{P}{2}} N_2(t-x) \right) \right] \quad (\text{C.18})$$

or,

$$R_{n_{DP}}(x) = E[N_1(t)N_1(t-x)] + \frac{P}{2} E[N_2(t)N_2(t-x)] + \sqrt{\frac{P}{2}} \left( E[N_1(t)N_2(t-x)] + E[N_2(t)N_1(t-x)] \right) \quad (\text{C.19})$$

Each of these components has to be calculated.

### C.3 Cross-Correlation between $N_1$ and $N_2$

The cross-correlation between  $N_1$  and  $N_2$  is given by:

$$R_{N_1 N_2}(x) = E[N_1(t)N_2(t-x)] \quad (\text{C.20})$$

Using Equation (C.6) and Equation (C.7), this can be expanded into:

$$R_{N_1 N_2}(x) = \tilde{R}(\varepsilon_\tau) E \left[ \begin{array}{l} \left( \begin{array}{l} (n_{I_E}(t)n_{I_P}(t)n_{I_E}(t-x) - n_{I_L}(t)n_{I_P}(t)n_{I_E}(t-x)) \\ + (n_{Q_E}(t)n_{Q_P}(t)n_{I_E}(t-x) - n_{Q_L}(t)n_{Q_P}(t)n_{I_E}(t-x)) \end{array} \right) \\ - \left( \begin{array}{l} (n_{I_E}(t)n_{I_P}(t)n_{I_L}(t-x) - n_{I_L}(t)n_{I_P}(t)n_{I_L}(t-x)) \\ + (n_{Q_E}(t)n_{Q_P}(t)n_{I_L}(t-x) - n_{Q_L}(t)n_{Q_P}(t)n_{I_L}(t-x)) \end{array} \right) \end{array} \right) \right] \quad (C.21)$$

It can be seen that all the terms between brackets are the product of three Gaussian noise types with zero mean. This means that the expected value of these products will be zero.

Then:

$$R_{N_1 N_2}(x) = 0 \quad (C.22)$$

Similarly,

$$R_{N_2 N_1}(x) = 0 \quad (C.23)$$

## C.4 $N_1$ PSD in $\mathbf{f=0}$

Using Equation (C.6), the autocorrelation of the noise  $N_1$  is given by:

$$R_{N_1}(x) = E[N_1(t)N_1(t-x)] = E \left[ \begin{array}{l} \left( \begin{array}{l} (n_{I_E}(t) - n_{I_L}(t))n_{I_P}(t) \\ + (n_{Q_E}(t) - n_{Q_L}(t))n_{Q_P}(t) \end{array} \right) \\ \times \left( \begin{array}{l} (n_{I_E}(t-x) - n_{I_L}(t-x))n_{I_P}(t-x) \\ + (n_{Q_E}(t-x) - n_{Q_L}(t-x))n_{Q_P}(t-x) \end{array} \right) \end{array} \right) \right] \quad (C.24)$$

It can be expanded into:

$$R_{N_1}(x) = \left( \begin{array}{l}
E[n_{I_P}(t)n_{I_E}(t)n_{I_P}(t-x)n_{I_E}(t-x)] \\
+ E[n_{I_P}(t)n_{I_L}(t)n_{I_P}(t-x)n_{I_L}(t-x)] \\
- 2E[n_{I_P}(t)n_{I_E}(t)n_{I_P}(t-x)n_{I_L}(t-x)] \\
+ E[n_{I_P}(t)n_{I_E}(t)n_{Q_P}(t-x)n_{Q_E}(t-x)] \\
+ E[n_{I_P}(t)n_{I_L}(t)n_{Q_P}(t-x)n_{Q_L}(t-x)] \\
- E[n_{I_P}(t)n_{I_E}(t)n_{Q_P}(t-x)n_{Q_L}(t-x)] \\
- E[n_{I_P}(t)n_{I_L}(t)n_{Q_P}(t-x)n_{Q_E}(t-x)] \\
+ E[n_{Q_P}(t)n_{Q_E}(t)n_{I_P}(t-x)n_{I_E}(t-x)] \\
+ E[n_{Q_P}(t)n_{Q_L}(t)n_{I_P}(t-x)n_{I_L}(t-x)] \\
- E[n_{Q_P}(t)n_{Q_E}(t)n_{I_P}(t-x)n_{I_L}(t-x)] \\
- E[n_{Q_P}(t)n_{Q_L}(t)n_{I_P}(t-x)n_{I_E}(t-x)] \\
+ E[n_{Q_P}(t)n_{Q_E}(t)n_{Q_P}(t-x)n_{Q_E}(t-x)] \\
+ E[n_{Q_P}(t)n_{Q_L}(t)n_{Q_P}(t-x)n_{Q_L}(t-x)] \\
- 2E[n_{Q_P}(t)n_{Q_E}(t)n_{Q_P}(t-x)n_{Q_L}(t-x)]
\end{array} \right) \quad (C.25)$$

Using the symmetry of the spreading code autocorrelation function, it can be simplified

into:

$$R_{N_1}(x) = \left( \begin{array}{l}
E[n_{I_P}(t)n_{I_E}(t)n_{I_P}(t-x)n_{I_E}(t-x)] \\
+ E[n_{I_P}(t)n_{I_L}(t)n_{I_P}(t-x)n_{I_L}(t-x)] \\
- 2E[n_{I_P}(t)n_{I_E}(t)n_{I_P}(t-x)n_{I_L}(t-x)] \\
+ 2E[n_{I_P}(t)n_{I_E}(t)n_{Q_P}(t-x)n_{Q_E}(t-x)] \\
+ 2E[n_{I_P}(t)n_{I_L}(t)n_{Q_P}(t-x)n_{Q_L}(t-x)] \\
- 2E[n_{I_P}(t)n_{I_E}(t)n_{Q_P}(t-x)n_{Q_L}(t-x)] \\
- 2E[n_{I_P}(t)n_{I_L}(t)n_{Q_P}(t-x)n_{Q_E}(t-x)] \\
+ E[n_{Q_P}(t)n_{Q_E}(t)n_{Q_P}(t-x)n_{Q_E}(t-x)] \\
+ E[n_{Q_P}(t)n_{Q_L}(t)n_{Q_P}(t-x)n_{Q_L}(t-x)] \\
- 2E[n_{Q_P}(t)n_{Q_E}(t)n_{Q_P}(t-x)n_{Q_L}(t-x)]
\end{array} \right) \quad (C.26)$$

It has been seen that the noise component of each correlator output can be considered as Gaussian, consequently, it is possible to use (Brown & Hwang 1992):

$$E[X_1X_2X_3X_4] = \left( \begin{array}{l} E[X_1X_2]E[X_3X_4] + E[X_1X_3]E[X_2X_4] \\ + E[X_1X_4]E[X_2X_3] \end{array} \right) \quad (\text{C.27})$$

For simplicity, the  $n^{\text{th}}$  factor composing Equation (C.26) will be referred to as  $C_n$ .

Consequently:

$$C_1(x) = \left( \begin{array}{l} E[n_{I_P}(t)n_{I_E}(t)]E[n_{I_P}(t-x)n_{I_E}(t-x)] \\ + E[n_{I_P}(t)n_{I_P}(t-x)]E[n_{I_E}(t)n_{I_E}(t-x)] \\ + E[n_{I_P}(t)n_{I_E}(t-x)]E[n_{I_E}(t)n_{I_P}(t-x)] \end{array} \right) \quad (\text{C.28})$$

There is a difference to notice between this expression, and the one used in Appendix A. Indeed, in Appendix A, the noise power was calculated through the computation of the autocorrelation function of the incoming noise. However, in the present case, the I&D filter has to be taken into account as well since:

$$n_{I_P}(t) = n_{FE}(t)c(t)\cos(2\pi f_{IF}t) * h_{ID}(t) \quad (\text{C.29})$$

Consequently, this results in:

$$E[n_{I_P}(t)n_{I_E}(t-x)] = \frac{N_0}{4} \tilde{R}_F\left(\frac{\delta}{2}\right) R_{h_{ID}}(x) \quad (\text{C.30})$$

Using Appendix A and Equation (C.8), and taking into account that the autocorrelation of the spreading code is symmetric (assuming a symmetric front-end filter), this can be simplified as:

$$\int_{-\infty}^{+\infty} C_1(x) dx = \left| \begin{aligned} & \left( \frac{N_0}{4T_I} \tilde{R}_F \left( \frac{\delta}{2} \right) \right)^2 \left( \int_{-\infty}^{+\infty} R_{hID}^2(0) dx \right) \\ & + \left( \left( \frac{N_0}{4T_I} \tilde{R}_F(0) \right)^2 + \left( \frac{N_0}{4T_I} \tilde{R}_F \left( \frac{\delta}{2} \right) \right)^2 \right) \left( \int_{-\infty}^{+\infty} R_{hID}^2(x) dx \right) \end{aligned} \right| \quad (C.31)$$

By doing so with all the components of  $N_1$ , and adding these up, one gets:

$$S_{N_1}(0) = \int_{-\infty}^{+\infty} R_{N_1}(x) dx = 4 \left( \frac{N_0}{4T_I} \right)^2 \tilde{R}_F(0) (\tilde{R}_F(0) - \tilde{R}_F(\delta)) \int_{-\infty}^{+\infty} R_{hID}^2(x) dx \quad (C.32)$$

From the expression of the I&D filter given in Appendix A, and taking the 3-dB filter into account (Holmes 1990), one can write:

$$\int_{-\infty}^{+\infty} R_{hID}^2(x) dx = T_I \quad (C.33)$$

And finally:

$$S_{N_1}(0) = \frac{N_0^2}{4T_I} \tilde{R}_F(0) (\tilde{R}_F(0) - \tilde{R}_F(\delta)) \quad (C.34)$$

## C.5 $N_2$ PSD in $\mathbf{f}=0$

From Equation (C.7), and assuming that the code delay error is small (it is then possible to use Equation (C.13)),  $N_2$  can be reduced to:

$$N_2 = \tilde{R}(0) (n_{I_E} - n_{I_L}) \quad (C.35)$$

and

$$R_{N_2}(x) = \tilde{R}^2(0) E \left[ \left( n_{I_E}(t) - n_{I_L}(t) \right) \left( n_{I_E}(t-x) - n_{I_L}(t-x) \right) \right] \quad (\text{C.36})$$

This expression is equal to:

$$R_{N_2}(x) = \tilde{R}^2(0) \left( \begin{array}{l} E \left[ n_{I_E}(t) n_{I_E}(t-x) \right] - E \left[ n_{I_L}(t) n_{I_E}(t-x) \right] \\ - E \left[ n_{I_E}(t) n_{I_L}(t-x) \right] + E \left[ n_{I_L}(t) n_{I_L}(t-x) \right] \end{array} \right) \quad (\text{C.37})$$

Using the same method as for the computation of  $R_{N_1}(x)$ , one gets:

$$R_{N_2}(x) = \frac{N_0}{2} \tilde{R}^2(0) \left( \tilde{R}_F(0) - \tilde{R}_F(\delta) \right) R_{h_{ID}}(x) \quad (\text{C.38})$$

and

$$S_{N_2}(0) = \int_{-\infty}^{+\infty} R_{N_2}(x) dx = \frac{N_0}{2} \tilde{R}^2(0) \left( R_F(0) - R_F(\delta) \right) \int_{-\infty}^{+\infty} R_{h_{ID}}(x) dx \quad (\text{C.39})$$

and

$$S_{N_2}(0) = \frac{N_0}{2} \tilde{R}^2(0) \left( \tilde{R}_F(0) - \tilde{R}_F(\delta) \right) \quad (\text{C.40})$$

## C.6 Conclusion

Using Equations (C.19), (C.22), (C.23), (C.34), and (C.40), one can write:

$$S_N(0) = \frac{N_0^2}{4T_I} \tilde{R}_F(0) \left( \tilde{R}_F(0) - \tilde{R}_F(\delta) \right) + \frac{PN_0}{4} \tilde{R}^2(0) \left( \tilde{R}_F(0) - \tilde{R}_F(\delta) \right) \quad (\text{C.41})$$

which can be re-written as:

$$S_N(0) = \frac{PN_0}{4} \tilde{R}^2(0) \left( \tilde{R}_F(0) - \tilde{R}_F(\delta) \right) \left( 1 + \frac{\tilde{R}_F(0) N_0}{T_I \tilde{R}^2(0)} \right) \quad (\text{C.42})$$

Using Equations (5.28), (C.16), and (C.42), one concludes that:

$$\sigma_{DP, \varepsilon_\tau}^2 = \frac{B_L(1-0.5B_L T_I)(\tilde{R}_F(0) - \tilde{R}_F(\delta))}{2 \frac{P}{N_0} \left( 2\pi \int_{-\infty}^{+\infty} f H(f) G(f) \sin(\pi f \delta) df \right)^2} \left( 1 + \frac{\tilde{R}_F(0)}{\frac{P}{N_0} \tilde{R}^2(0) T_I} \right) \quad (\text{C.43})$$

It can be shown that:

$$\tilde{R}_F(0) - \tilde{R}_F(\delta) = \int_{-\infty}^{+\infty} |H(f)|^2 G(f) \sin^2(\pi f \delta) df \quad (\text{C.44})$$

Consequently,

$$\sigma_{DP, \varepsilon_\tau}^2 = \frac{B_L(1-0.5B_L T_I) \left( \int_{-\infty}^{+\infty} |H(f)|^2 G(f) \sin^2(\pi f \delta) df \right)}{2 \frac{P}{N_0} \left( 2\pi \int_{-\infty}^{+\infty} f H(f) G(f) \sin(\pi f \delta) df \right)^2} \left( 1 + \frac{\tilde{R}_F(0)}{\frac{P}{N_0} \tilde{R}^2(0) T_I} \right) \quad (\text{C.45})$$

Finally, assuming that the front-end filter has a unity gain within  $\pm B$  Hz, and is null elsewhere, it can be concluded that:

$$\sigma_{DP, \varepsilon_\tau}^2 = \frac{B_L(1-0.5B_L T_I) \left( \int_{-B}^{+B} G(f) \sin^2(\pi f \delta) df \right)}{2 \frac{P}{N_0} \left( 2\pi \int_{-B}^{+B} f G(f) \sin(\pi f \delta) df \right)^2} \left( 1 + \frac{1}{\frac{P}{N_0} T_I \int_{-B}^{+B} G(f) df} \right) \quad (\text{C.46})$$

## APPENDIX D

### ASPeCT DOT-PRODUCT DISCRIMINATOR STANDARD DEVIATION IN THE PRESENCE OF GAUSSIAN NOISE

The following calculations are based on in the results of Appendix C.

#### D.1 Background

Following the notations introduced in Chapter 2 and Chapter 5, the ASPeCT DP discriminator is given by:

$$D_{DP}^{ASPeCT} = \begin{pmatrix} [(I_E - I_L)I_P + (Q_E - Q_L)Q_P]_B \\ -\beta[(I_E - I_L)I_P + (Q_E - Q_L)Q_P]_{BP} \end{pmatrix} \quad (D.1)$$

Assuming that Gaussian noise is the only disturbance and that the carrier-phase tracking is perfect, one can then simplify the expression as:

$$D_{DP}^{ASPeCT} = \begin{pmatrix} \left( \begin{aligned} & \frac{P}{2} \left( \tilde{R}_B \left( \varepsilon_\tau + \frac{\delta}{2} \right) - \tilde{R}_B \left( \varepsilon_\tau - \frac{\delta}{2} \right) \right) \tilde{R}_B(\varepsilon_\tau) \\ & + (n_{I_{E,B}} - n_{I_{L,B}}) \mathbf{h}_{I_{P,B}} + (n_{Q_{E,B}} - n_{Q_{L,B}}) \mathbf{h}_{Q_{P,B}} \\ & + \sqrt{\frac{P}{2}} \left( \tilde{R}_B(\varepsilon_\tau) (n_{I_{E,B}} - n_{I_{L,B}}) + \left( \tilde{R}_B \left( \varepsilon_\tau + \frac{\delta}{2} \right) - \tilde{R}_B \left( \varepsilon_\tau - \frac{\delta}{2} \right) \right) n_{I_{P,B}} \right) \end{aligned} \right) \\ -\beta \left( \begin{aligned} & \frac{P}{2} \left( \tilde{R}_{BP} \left( \varepsilon_\tau + \frac{\delta}{2} \right) - \tilde{R}_{BP} \left( \varepsilon_\tau - \frac{\delta}{2} \right) \right) \tilde{R}_{BP}(\varepsilon_\tau) \\ & + (n_{I_{E,BP}} - n_{I_{L,BP}}) \mathbf{h}_{I_{P,BP}} + (n_{Q_{E,BP}} - n_{Q_{L,BP}}) \mathbf{h}_{Q_{P,BP}} \\ & + \sqrt{\frac{P}{2}} \left( \tilde{R}_{BP}(\varepsilon_\tau) (n_{I_{E,BP}} - n_{I_{L,BP}}) + \left( \tilde{R}_{BP} \left( \varepsilon_\tau + \frac{\delta}{2} \right) - \tilde{R}_{BP} \left( \varepsilon_\tau - \frac{\delta}{2} \right) \right) n_{I_{P,BP}} \right) \end{aligned} \right) \end{pmatrix} \quad (\text{D.2})$$

This expression can be re-written as:

$$D_{DP}^{ASPeCT} = \begin{pmatrix} \left( \begin{aligned} & \frac{P}{2} \left( \tilde{R}_B \left( \varepsilon_\tau + \frac{\delta}{2} \right) - \tilde{R}_B \left( \varepsilon_\tau - \frac{\delta}{2} \right) \right) \tilde{R}_B(\varepsilon_\tau) \\ & + N_1(t) + \sqrt{\frac{P}{2}} N_2(t) \end{aligned} \right) \\ -\beta \left( \begin{aligned} & \frac{P}{2} \left( \tilde{R}_{BP} \left( \varepsilon_\tau + \frac{\delta}{2} \right) - \tilde{R}_{BP} \left( \varepsilon_\tau - \frac{\delta}{2} \right) \right) \tilde{R}_{BP}(\varepsilon_\tau) \\ & + N_3 + \sqrt{\frac{P}{2}} N_4(t) \end{aligned} \right) \end{pmatrix} \quad (\text{D.3})$$

with

$$N_1(t) = (n_{I_{E,B}} - n_{I_{L,B}}) \mathbf{h}_{I_{P,B}} + (n_{Q_{E,B}} - n_{Q_{L,B}}) \mathbf{h}_{Q_{P,B}} \quad (\text{D.4})$$

$$N_2(t) = \tilde{R}_B(\varepsilon_\tau) (n_{I_{E,B}} - n_{I_{L,B}}) + \left( \tilde{R}_B \left( \varepsilon_\tau + \frac{\delta}{2} \right) - \tilde{R}_B \left( \varepsilon_\tau - \frac{\delta}{2} \right) \right) n_{I_{P,B}} \quad (\text{D.5})$$

$$N_3(t) = (n_{I_{E,BP}} - n_{I_{L,BP}}) \mathbf{h}_{I_{P,BP}} + (n_{Q_{E,BP}} - n_{Q_{L,BP}}) \mathbf{h}_{Q_{P,BP}} \quad (\text{D.6})$$

$$N_4(t) = \tilde{R}_{BP}(\varepsilon_\tau) (n_{I_{E,BP}} - n_{I_{L,BP}}) + \left( \tilde{R}_{BP} \left( \varepsilon_\tau + \frac{\delta}{2} \right) - \tilde{R}_{BP} \left( \varepsilon_\tau - \frac{\delta}{2} \right) \right) n_{I_{P,BP}} \quad (\text{D.7})$$

Assuming that  $\varepsilon_\tau \approx 0$ , and knowing that  $\tilde{R}_B$  is even and that  $\tilde{R}_{BP}$  is odd,  $N_2$  and

$N_4$  can be simplified as:

$$N_2(t) = \tilde{R}_B(0) (n_{I_{E,B}} - n_{I_{L,B}}) \quad (\text{D.8})$$

$$N_4(t) = \left( \tilde{R}_{BP}\left(\frac{\delta}{2}\right) - \tilde{R}_{BP}\left(-\frac{\delta}{2}\right) \right) n_{I_{P,BP}} \quad (\text{D.9})$$

In order to be able to use Equation (5.28), it is necessary to calculate the PSD of the ASPeCT DP discriminator noise in  $f = 0$ . As explained in Appendix C, the calculation of the discriminator output autocorrelation can be done as:

$$R_N(x) = E \left[ \left( N_1(t) + \sqrt{\frac{P}{2}} N_2(t) - \beta \left( N_3(t) + \sqrt{\frac{P}{2}} N_4(t) \right) \right) \times \left( N_1(t-x) + \sqrt{\frac{P}{2}} N_2(t-x) - \beta \left( N_3(t-x) + \sqrt{\frac{P}{2}} N_4(t-x) \right) \right) \right] \quad (\text{D.10})$$

and

$$R_N(x) = \left( \begin{aligned} & E[N_1(t)N_1(t-x)] + \frac{P}{2} E[N_2(t)N_2(t-x)] + \beta^2 E[N_3(t)N_3(t-x)] \\ & + \frac{P}{2} \beta^2 E[N_4(t)N_4(t-x)] \\ & + \sqrt{\frac{P}{2}} E[N_1(t)N_2(t-x)] - \beta E[N_1(t)N_3(t-x)] - \sqrt{\frac{P}{2}} \beta E[N_1(t)N_4(t-x)] \\ & + \sqrt{\frac{P}{2}} E[N_2(t)N_1(t-x)] - \sqrt{\frac{P}{2}} \beta E[N_2(t)N_3(t-x)] - \frac{P}{2} \beta E[N_2(t)N_4(t-x)] \\ & - \beta E[N_3(t)N_1(t-x)] - \sqrt{\frac{P}{2}} \beta E[N_3(t)N_2(t-x)] + \sqrt{\frac{P}{2}} \beta^2 E[N_3(t)N_4(t-x)] \\ & - \sqrt{\frac{P}{2}} \beta E[N_4(t)N_1(t-x)] - \frac{P}{2} \beta E[N_4(t)N_2(t-x)] + \sqrt{\frac{P}{2}} \beta^2 E[N_4(t)N_3(t-x)] \end{aligned} \right) \quad (\text{D.11})$$

It can shown, using the same model as in Section C.3, that

$$E[N_1(t)N_2(t-x)] = E[N_2(t)N_1(t-x)] = 0 \quad (\text{D.12})$$

$$E[N_3(t)N_4(t-x)] = E[N_4(t)N_3(t-x)] = 0 \quad (\text{D.13})$$

$$E[N_1(t)N_4(t-x)] = E[N_4(t)N_1(t-x)] = 0 \quad (\text{D.14})$$

$$E[N_2(t)N_3(t-x)] = E[N_3(t)N_2(t-x)] = 0 \quad (\text{D.15})$$

Equation (D.11) can then be reduced to:

$$R_N(x) = \left( \begin{array}{l} E[N_1(t)N_1(t-x)] + \frac{P}{2} E[N_2(t)N_2(t-x)] + \beta^2 E[N_3(t)N_3(t-x)] \\ + \frac{P}{2} \beta^2 E[N_4(t)N_4(t-x)] \\ - \beta(E[N_1(t)N_3(t-x)] + E[N_3(t)N_1(t-x)]) \\ - \frac{P}{2} \beta(E[N_2(t)N_4(t-x)] + E[N_4(t)N_2(t-x)]) \end{array} \right) \quad (\text{D.16})$$

Since the ASPeCT DP discriminator integrated entirely the traditional DP discriminator,

$S_{N_1}(0)$  and  $S_{N_2}(0)$  can directly be taken from Section C.4 and C.5. As a reminder, and

using the new notations, these are given by:

$$S_{N_1}(0) = \frac{N_0^2}{4T_I} \tilde{R}_{F,B}(0) (\tilde{R}_{F,B}(0) - \tilde{R}_{F,B}(\delta)) \quad (\text{D.17})$$

and

$$S_{N_2}(0) = \frac{N_0}{2} \tilde{R}_B^2(0) (\tilde{R}_{F,B}(0) - \tilde{R}_{F,B}(\delta)) \quad (\text{D.18})$$

In a very similar manner as for  $N_1$ , the only difference being the use of a local replica of the PRN code, it can be shown that:

$$S_{N_3}(0) = \frac{N_0^2}{4T_I} \tilde{R}_{F,P}(0) (\tilde{R}_{F,P}(0) - \tilde{R}_{F,P}(\delta)) \quad (\text{D.19})$$

As a conclusion, what is left is the calculation of the discriminator gain and the rest of the noise contribution.

## D.2 Loop Gain

The gain at the origin can be calculated using:

$$K_{DP}^{ASPeCT} = \left. \frac{dD_{DP}^{ASPeCT}}{d\varepsilon_\tau} \right|_{\varepsilon_\tau=0} \quad (\text{D.20})$$

$$K_{DP}^{ASPeCT} = \frac{P}{2} \left( \begin{array}{l} \left( \frac{d}{d\varepsilon_\tau} \tilde{R}_B \left( \varepsilon_\tau - \frac{\delta}{2} \right) - \frac{d}{d\varepsilon_\tau} \tilde{R}_B \left( \varepsilon_\tau + \frac{\delta}{2} \right) \right) R_B(\varepsilon_\tau) \\ + \left( \tilde{R}_B \left( -\frac{\delta}{2} \right) - \tilde{R}_B \left( \frac{\delta}{2} \right) \right) \frac{\delta}{d\varepsilon_\tau} \tilde{R}_B(\varepsilon_\tau) \\ - \left( \frac{d}{d\varepsilon_\tau} \tilde{R}_{BP} \left( \varepsilon_\tau - \frac{\delta}{2} \right) - \frac{d}{d\varepsilon_\tau} \tilde{R}_{BP} \left( \varepsilon_\tau + \frac{\delta}{2} \right) \right) \tilde{R}_{BP}(\varepsilon_\tau) \\ + \left( \tilde{R}_{BP} \left( -\frac{\delta}{2} \right) - \tilde{R}_{BP} \left( \frac{\delta}{2} \right) \right) \frac{d}{d\varepsilon_\tau} \tilde{R}_{BP}(\varepsilon_\tau) \end{array} \right) \Bigg|_{\varepsilon_\tau=0} \quad (\text{D.21})$$

Using Equation (C.16):

$$\frac{P}{2} \left( \begin{array}{c} \left( \frac{d\tilde{R}_B\left(\varepsilon_\tau - \frac{\delta}{2}\right)}{d\varepsilon_\tau} \right) \\ \tilde{R}_B(\varepsilon) \\ \left( -\frac{d\tilde{R}_B\left(\varepsilon_\tau + \frac{\delta}{2}\right)}{d\varepsilon_\tau} \right) \end{array} \right)_{\varepsilon_\tau=0} = 2\pi P \left( \int_{-\infty}^{+\infty} fH(f)G_B(f)\sin(\pi f\delta)df \right) \tilde{R}_B(0) \quad (\text{D.22})$$

$$+ \left( \begin{array}{c} \tilde{R}_B\left(\varepsilon_\tau - \frac{\delta}{2}\right) \\ \frac{d\tilde{R}_B(\varepsilon_\tau)}{d\varepsilon_\tau} \\ -\tilde{R}_B\left(\varepsilon_\tau + \frac{\delta}{2}\right) \end{array} \right)_{\varepsilon_\tau=0}$$

Following the exact same procedure, it can be shown that:

$$\left( \begin{array}{c} \left( \frac{d\tilde{R}_{BP}\left(\varepsilon_\tau - \frac{\delta}{2}\right)}{d\varepsilon_\tau} \right) \\ \tilde{R}_{BP}(\varepsilon_\tau) \\ \left( -\frac{d\tilde{R}_{BP}\left(\varepsilon_\tau + \frac{\delta}{2}\right)}{d\varepsilon_\tau} \right) \end{array} \right)_{\varepsilon_\tau=0} = 2\pi P \left( \begin{array}{c} \left( \tilde{R}_{BP}\left(-\frac{\delta}{2}\right) - \tilde{R}_{BP}\left(\frac{\delta}{2}\right) \right) \\ \times \left( \int_{-\infty}^{+\infty} jfH(f)F_B(f)F_P^*(f)df \right) \end{array} \right) \quad (\text{D.23})$$

$$+ \left( \begin{array}{c} \tilde{R}_{BP}\left(\varepsilon_\tau - \frac{\delta}{2}\right) \\ \frac{d\tilde{R}_{BP}(\varepsilon_\tau)}{d\varepsilon_\tau} \\ -\tilde{R}_{BP}\left(\varepsilon_\tau + \frac{\delta}{2}\right) \end{array} \right)_{\varepsilon_\tau=0}$$

and

$$K = 2\pi P \left( \begin{array}{c} \left( \int_{-\infty}^{+\infty} fH(f)G_B(f)\sin(\pi f\delta)df \right) \tilde{R}_B(0) \\ -\beta \left( \tilde{R}_{BP}\left(-\frac{\delta}{2}\right) - \tilde{R}_{BP}\left(\frac{\delta}{2}\right) \right) \left( \int_{-\infty}^{+\infty} jfH(f)F_B(f)F_P^*(f)df \right) \end{array} \right) \quad (\text{D.24})$$

### D.3 $N_4$ PSD in $f=0$

From Equation (D.9), one can write:

$$R_{N_4}(x) = E[N_4(t)N_4(t-x)] = \left( \tilde{R}_{BP}\left(\frac{\delta}{2}\right) - \tilde{R}_{BP}\left(-\frac{\delta}{2}\right) \right)^2 E[n_{I_{P,BP}}(t)n_{I_{P,BP}}(t-x)] \quad (D.25)$$

In an similar way as in Section C.4, this gives:

$$R_{N_4}(x) = \frac{N_0}{4} \left( \tilde{R}_{BP}\left(\frac{\delta}{2}\right) - \tilde{R}_{BP}\left(-\frac{\delta}{2}\right) \right)^2 \tilde{R}_{F,P}(0) R_{h_{ID}}(x) \quad (D.26)$$

Finally,

$$S_{N_4}(0) = \frac{N_0}{4} \left( \tilde{R}_{BP}\left(\frac{\delta}{2}\right) - \tilde{R}_{BP}\left(-\frac{\delta}{2}\right) \right)^2 \tilde{R}_{F,P}(0) \quad (D.27)$$

### C.4 Contribution of the Correlation between $N_1$ and $N_3$

From Equations (D.4) and (D.6), it can be shown that:

$$R_{N_1 N_3}(x) = E[N_1(t)N_3(t-x)] = E \left[ \begin{array}{l} \left( n_{I_{P,B}}(t) \left( n_{I_{E,B}}(t) - n_{I_{L,B}}(t) \right) \right. \\ \left. + n_{Q_{P,B}}(t) \left( n_{Q_{E,B}}(t) - n_{Q_{L,B}}(t) \right) \right) \\ \left( n_{I_{P,BP}}(t-x) \left( n_{I_{E,BP}}(t-x) \right. \right. \\ \left. \left. - n_{I_{L,BP}}(t-x) \right) \right) \\ \left. + n_{Q_{P,BP}}(t-x) \left( n_{Q_{E,BP}}(t-x) \right. \right. \\ \left. \left. - n_{Q_{L,BP}}(t-x) \right) \right) \end{array} \right] \quad (D.28)$$

Following the work already done in Section C.3, it can be shown that:

$$R_{N_1 N_3}(x) = 2 \left( \frac{N_0}{4} \right)^2 \begin{pmatrix} 2\tilde{R}_{F,BP}\left(-\frac{\delta}{2}\right)\tilde{R}_{F,BP}\left(\frac{\delta}{2}\right) \\ -\tilde{R}_{F,BP}^2\left(-\frac{\delta}{2}\right) - \tilde{R}_{F,BP}^2\left(\frac{\delta}{2}\right) \end{pmatrix} R_{h_{ID}}^2(x) \quad (\text{D.29})$$

Since  $F_P^*(f)$  is even, and  $F_B(f)$  is odd, it can be shown, assuming a symmetric filter,

that:

$$\tilde{R}_{F,BP}\left(\frac{\delta}{2}\right) = -\tilde{R}_{F,BP}\left(-\frac{\delta}{2}\right) \quad (\text{D.30})$$

and

$$R_{N_1 N_3}(x) = -\frac{N_0^2}{2} \tilde{R}_{F,BP}^2\left(\frac{\delta}{2}\right) R_{h_{ID}}^2(x) \quad (\text{D.31})$$

Finally,

$$S_{N_1 N_3}(0) = -\frac{N_0^2}{2T_I} \tilde{R}_{F,BP}^2\left(\frac{\delta}{2}\right) \quad (\text{D.32})$$

In the same way, it can be shown that:

$$S_{N_3 N_1}(0) = -\frac{N_0^2}{2T_I} \tilde{R}_{F,BP}^2\left(\frac{\delta}{2}\right) \quad (\text{D.33})$$

## D.5 Contribution of the Correlation between $N_2$ and $N_4$

From Equations (D.8) and (D.9), the correlation of  $N_2$  and  $N_4$  equals:

$$R_{N_2N_4}(x) = E[N_2(t)N_4(t-x)] = \left( \begin{array}{c} \tilde{R}_B(0) \left( \tilde{R}_{BP}\left(\frac{\delta}{2}\right) - \tilde{R}_{BP}\left(-\frac{\delta}{2}\right) \right) \\ \times E \left[ \left( n_{I_{E,B}}(t) - n_{I_{L,B}}(t) \right) n_{I_{P,BP}}(t-x) \right] \end{array} \right) \quad (\text{D.34})$$

Once again, with a derivation similar to what was done in Section C.5, it can be shown that:

$$R_{N_2N_4}(x) = \frac{N_0}{4} \tilde{R}_B(0) \left( \tilde{R}_{BP}\left(\frac{\delta}{2}\right) - \tilde{R}_{BP}\left(-\frac{\delta}{2}\right) \right) \left( \begin{array}{c} \tilde{R}_{F,BP}\left(\frac{\delta}{2}\right) \\ -\tilde{R}_{F,BP}\left(-\frac{\delta}{2}\right) \end{array} \right) R_{h_{ID}}(x) \quad (\text{D.35})$$

In a similar way:

$$R_{N_4N_2}(x) = \frac{N_0}{4} \tilde{R}_B(0) \left( \tilde{R}_{BP}\left(\frac{\delta}{2}\right) - \tilde{R}_{BP}\left(-\frac{\delta}{2}\right) \right) \left( \begin{array}{c} \tilde{R}_{F,BP}\left(-\frac{\delta}{2}\right) \\ -\tilde{R}_{F,BP}\left(\frac{\delta}{2}\right) \end{array} \right) R_{h_{ID}}(x) \quad (\text{D.36})$$

Since  $\tilde{R}_{BP}$  is asymmetric (see Equation (D.30)), then:

$$R_{N_2N_4}(x) + R_{N_4N_2}(x) = 0 \quad (\text{D.37})$$

Finally,

$$S_{N_2N_4}(0) + S_{N_4N_2}(0) = 0 \quad (\text{D.38})$$

## D.6 Conclusion

Taking Equations (D.16), (D.17), (D.18), (D.19), (D.27), (D.32), (D.33), (D.38) into account, the ASPeCT DP discriminator noise PSD in  $f = 0$  equals:

$$S_N(0) = \left( \begin{aligned} & \frac{N_0^2}{4T_I} \tilde{R}_{F,B}(0) (\tilde{R}_{F,B}(0) - \tilde{R}_{F,B}(\delta)) + \frac{PN_0}{4} \tilde{R}_B^2(0) (\tilde{R}_{F,B}(0) - \tilde{R}_{F,B}(\delta)) \\ & + \frac{N_0^2}{4T_I} \beta^2 \tilde{R}_{F,P}(0) (\tilde{R}_{F,P}(0) - \tilde{R}_{F,P}(\delta)) \\ & \frac{PN_0}{8} \beta^2 \left( \tilde{R}_{BP}\left(\frac{\delta}{2}\right) - \tilde{R}_{BP}\left(-\frac{\delta}{2}\right) \right)^2 \tilde{R}_{F,P}(0) \\ & + \frac{N_0^2}{T_I} \beta \tilde{R}_{F,BP}^2\left(\frac{\delta}{2}\right) \end{aligned} \right) \quad (\text{D.39})$$

This can be re-arranged as:

$$S_N(0) = \left( \begin{aligned} & \frac{PN_0}{8} \left( 2\tilde{R}_B^2(0) (\tilde{R}_{F,B}(0) - \tilde{R}_{F,B}(\delta)) \right. \\ & \quad \left. + \beta^2 \left( \tilde{R}_{BP}\left(\frac{\delta}{2}\right) - \tilde{R}_{BP}\left(-\frac{\delta}{2}\right) \right)^2 \tilde{R}_{F,P}(0) \right) \\ & + \frac{N_0^2}{4T_I} \left( \begin{aligned} & \tilde{R}_{F,B}(0) (\tilde{R}_{F,B}(0) - \tilde{R}_{F,B}(\delta)) \\ & + \beta^2 \tilde{R}_{F,P}(0) (\tilde{R}_{F,P}(0) - \tilde{R}_{F,P}(\delta)) \\ & + 4\beta \tilde{R}_{F,BP}^2\left(\frac{\delta}{2}\right) \end{aligned} \right) \end{aligned} \right) \quad (\text{D.40})$$

Using Equation (5.28) and Equation (D.24):

$$\sigma_{ASPeCT,DP,\varepsilon_\tau}^2 = \left( \frac{B_L(1-0.5B_L T_I) \left( 2\tilde{R}_B^2(0)(\tilde{R}_{F,B}(0) - \tilde{R}_{F,B}(\delta)) + \beta^2 \left( \tilde{R}_{BP}\left(\frac{\delta}{2}\right) - \tilde{R}_{BP}\left(-\frac{\delta}{2}\right) \right)^2 \tilde{R}_{F,P}(0) \right)}{4 \frac{P}{N_0} \left( 2\pi \left( \left( \int_{-\infty}^{+\infty} fH(f)G_B(f)\sin(\pi f\delta)df \right) \tilde{R}_B(0) - \beta \left( \tilde{R}_{BP}\left(-\frac{\delta}{2}\right) - \tilde{R}_{BP}\left(\frac{\delta}{2}\right) \right) \left( \int_{-\infty}^{+\infty} jfH(f)F_B(f)F_P^*(f)df \right) \right) \right)^2} \right)^2 \times \left( 1 + \frac{\left( \tilde{R}_{F,B}(0)(\tilde{R}_{F,B}(0) - \tilde{R}_{F,B}(\delta)) + \beta^2 \tilde{R}_{F,P}(0)(\tilde{R}_{F,P}(0) - \tilde{R}_{F,P}(\delta)) + 4\beta \tilde{R}_{F,BP}^2\left(\frac{\delta}{2}\right) \right)}{\frac{P}{N_0} \left( 2\tilde{R}_B^2(0)(\tilde{R}_{F,B}(0) - \tilde{R}_{F,B}(\delta)) + \beta^2 \left( \tilde{R}_{BP}\left(\frac{\delta}{2}\right) - \tilde{R}_{BP}\left(-\frac{\delta}{2}\right) \right)^2 \tilde{R}_{F,P}(0) \right)} \right) \right) \quad (D.41)$$

Assuming a front-end filter with unity gain within  $\pm B$  Hz, and zero elsewhere, the expression becomes:

$$\sigma_{ASPeCT, DP, \varepsilon_\tau}^2 = \left( \frac{B_L(1-0.5B_L T_I) \left( 2\tilde{R}_B^2(0)(\tilde{R}_B(0) - \tilde{R}_B(\delta)) + \beta^2 \left( \tilde{R}_{BP}\left(\frac{\delta}{2}\right) - \tilde{R}_{BP}\left(-\frac{\delta}{2}\right) \right)^2 \tilde{R}_P(0) \right)}{4 \frac{P}{N_0} \left( 2\pi \left( \int_{-B}^{+B} f G_B(f) \sin(\pi f \delta) df \right) \tilde{R}_B(0) - \beta \left( \tilde{R}_{BP}\left(-\frac{\delta}{2}\right) - \tilde{R}_{BP}\left(\frac{\delta}{2}\right) \right) \left( \int_{-B}^{+B} j f F_B(f) F_P^*(f) df \right) \right)^2} \right)^2 \times \left( 1 + \frac{\left( \begin{array}{c} \tilde{R}_B(0)(\tilde{R}_B(0) - \tilde{R}_B(\delta)) \\ 2 + \beta^2 \tilde{R}_P(0)(\tilde{R}_P(0) - \tilde{R}_P(\delta)) \\ + 4\beta \tilde{R}_{BP}^2\left(\frac{\delta}{2}\right) \end{array} \right)}{\frac{P}{N_0} \left( 2\tilde{R}_B^2(0)(\tilde{R}_B(0) - \tilde{R}_B(\delta)) + \beta^2 \left( \tilde{R}_{BP}\left(\frac{\delta}{2}\right) - \tilde{R}_{BP}\left(-\frac{\delta}{2}\right) \right)^2 \tilde{R}_P(0) \right)} \right) \quad (D.42)$$

Finally, after a few trigonometric manipulations, the analytical expression can be written as:

$$\sigma_{ASPeCT, DP, \varepsilon_\tau}^2 = \left( \frac{B_L(1-0.5B_L T_I) \left( \left( \int_{-B}^{+B} |F_B|^2 df \right)^2 \left( \int_{-B}^{+B} |F_B|^2 \sin^2(\pi f d) df \right) + \beta^2 \left( \int_{-B}^{+B} jF_B(f)F_P^*(f)\sin(\pi f d) df \right)^2 \left( \int_{-B}^{+B} |F_P(f)|^2 df \right) \right)}{\frac{P}{N_0} \left( 2\pi \left( \left( \int_{-B}^{+B} f|F_B(f)|^2 \sin(\pi f d) df \right) \left( \int_{-B}^{+B} |F_B(f)|^2 df \right) - \beta \left( \int_{-B}^{+B} F_B(f)F_P^*(f)\sin(\pi f d) df \right) \left( \int_{-B}^{+B} fF_B(f)F_P^*(f) df \right) \right) \right)^2} \right) \times \left( 1 + \frac{\left( \int_{-B}^{+B} |F_B|^2 df \right) \left( \int_{-B}^{+B} |F_B|^2 \sin^2(\pi f d) df \right) + \beta^2 \left( \int_{-\infty}^{+\infty} |F_P(f)|^2 df \right) \left( \int_{-\infty}^{+\infty} |F_P(f)|^2 \sin^2(\pi f d) df \right) + 2\beta \left( \int_{-B}^{+B} jF_B(f)F_P^*(f)\sin(\pi f d) df \right)^2}{\frac{P}{N_0} T_I \left( \left( \int_{-B}^{+B} |F_B|^2 df \right)^2 \left( \int_{-B}^{+B} |F_B|^2 \sin^2(\pi f d) df \right) + \beta^2 \left( \int_{-B}^{+B} jF_B(f)F_P^*(f)\sin(\pi f d) df \right)^2 \left( \int_{-B}^{+B} |F_P(f)|^2 df \right) \right)} \right)$$

(D.43)

## APPENDIX E

### ASPECT EMLP DISCRIMINATOR VARIANCE IN THE PRESENCE OF GAUSSIAN NOISE

The following calculations are strongly based on what was done in Appendix C.

#### E.1 Background

Following the notations introduced in Chapter 2 and Chapter 5, ASPeCT EMLP discriminator is given by:

$$D_{EMLP}^{ASPeCT}(\varepsilon_\tau) = \left[ (I_E^2 - I_L^2) + (Q_E^2 - Q_L^2) \right]_B - \beta \left[ (I_E^2 - I_L^2) + (Q_E^2 - Q_L^2) \right]_{BP} \quad (\text{E.1})$$

Assuming that the carrier-phase tracking is perfect, we can write that:

$$D_{EMLP}^{ASPeCT}(\varepsilon_\tau) = \begin{pmatrix} \left( \frac{P}{2} \left( \tilde{R}_B^2 \left( \varepsilon_\tau + \frac{\delta}{2} \right) - \tilde{R}_B^2 \left( \varepsilon_\tau - \frac{\delta}{2} \right) \right) \right. \\ \left. + \left( n_{I_{E,B}}^2 - n_{I_{L,B}}^2 \right) + \left( n_{Q_{E,B}}^2 - n_{Q_{L,B}}^2 \right) \right. \\ \left. + \sqrt{2P} \left( \tilde{R}_B \left( \varepsilon_\tau - \frac{\delta}{2} \right) n_{I_{E,B}} - \tilde{R}_B \left( \varepsilon_\tau + \frac{\delta}{2} \right) n_{I_{L,B}} \right) \right) \\ -\beta \left( \frac{P}{2} \left( \tilde{R}_{BP}^2 \left( \varepsilon_\tau + \frac{\delta}{2} \right) - \tilde{R}_{BP}^2 \left( \varepsilon_\tau - \frac{\delta}{2} \right) \right) \right. \\ \left. + \left( n_{I_{E,BP}}^2 - n_{I_{L,BP}}^2 \right) + \left( n_{Q_{E,BP}}^2 - n_{Q_{L,BP}}^2 \right) \right. \\ \left. + \sqrt{2P} \left( \tilde{R}_{BP} \left( \varepsilon_\tau - \frac{\delta}{2} \right) n_{I_{E,BP}} - \tilde{R}_{BP} \left( \varepsilon_\tau + \frac{\delta}{2} \right) n_{I_{L,BP}} \right) \right) \end{pmatrix} \quad (\text{E.2})$$

The noise  $N$  can be divided into four parts:

$$N_1(t) = \left( n_{I_{E,B}}^2 - n_{I_{L,B}}^2 \right) + \left( n_{Q_{E,B}}^2 - n_{Q_{L,B}}^2 \right) \quad (\text{E.3})$$

$$N_2(t) = \left( \tilde{R}_B \left( \varepsilon_\tau - \frac{\delta}{2} \right) n_{I_{E,B}} - \tilde{R}_B \left( \varepsilon_\tau + \frac{\delta}{2} \right) n_{I_{L,B}} \right) \quad (\text{E.4})$$

$$N_3(t) = \left( \left( n_{I_{E,BP}}^2 - n_{I_{L,BP}}^2 \right) + \left( n_{Q_{E,BP}}^2 - n_{Q_{L,BP}}^2 \right) \right) \quad (\text{E.5})$$

$$N_4(t) = \left( \tilde{R}_{BP} \left( \varepsilon_\tau - \frac{\delta}{2} \right) n_{I_{E,BP}} - \tilde{R}_{BP} \left( \varepsilon_\tau + \frac{\delta}{2} \right) n_{I_{L,BP}} \right) \quad (\text{E.6})$$

Assuming a  $\varepsilon_\tau \approx 0$ , and knowing that  $\tilde{R}_B$  is even and  $\tilde{R}_{BP}$  is odd,  $N_2$  and  $N_4$  can be simplified into:

$$N_2(t) = \tilde{R}_B \left( \frac{\delta}{2} \right) \left( n_{I_{E,B}} - n_{I_{L,B}} \right) \quad (\text{E.7})$$

$$N_4(t) = \tilde{R}_{BP} \left( -\frac{\delta}{2} \right) \left( n_{I_{E,BP}} + n_{I_{L,BP}} \right) \quad (\text{E.8})$$

The discriminator output noise autocorrelation is given by:

$$R_N(x) = E \left[ \begin{aligned} & \left( N_1(t) + \sqrt{2P}N_2(t) - \beta N_3(t) - \sqrt{2P}\beta N_4(t) \right) \\ & \times \left( N_1(t-x) + \sqrt{2P}N_2(t-x) - \beta N_3(t-x) - \sqrt{2P}\beta N_4(t-x) \right) \end{aligned} \right] \quad (\text{E.9})$$

$$R_N(x) = \begin{pmatrix} E[N_1(t)N_1(t-x)] + 2PE[N_2(t)N_2(t-x)] \\ + \beta^2 E[N_3(t)N_3(t-x)] + 2P\beta^2 E[N_4(t)N_4(t-x)] \\ + \sqrt{2P}E[N_1(t)N_2(t-x)] - \beta E[N_1(t)N_3(t-x)] \\ - \sqrt{2P}\beta E[N_1(t)N_4(t-x)] + \sqrt{2P}E[N_2(t)N_1(t-x)] \\ - \sqrt{2P}\beta E[N_2(t)N_3(t-x)] - 2P\beta E[N_2(t)N_4(t-x)] \\ - \beta E[N_3(t)N_1(t-x)] - \sqrt{2P}\beta E[N_3(t)N_2(t-x)] \\ + \sqrt{2P}\beta^2 E[N_3(t)N_4(t-x)] - \sqrt{2P}\beta E[N_4(t)N_1(t-x)] \\ - 2P\beta E[N_4(t)N_2(t-x)] + \sqrt{2P}\beta^2 E[N_4(t)N_3(t-x)] \end{pmatrix} \quad (\text{E.10})$$

It can be shown (see Section C.3), that

$$E[N_1(t)N_2(t-x)] = E[N_2(t)N_1(t-x)] = 0 \quad (\text{E.11})$$

$$E[N_3(t)N_4(t-x)] = E[N_4(t)N_3(t-x)] = 0 \quad (\text{E.12})$$

$$E[N_1(t)N_4(t-x)] = E[N_4(t)N_1(t-x)] = 0 \quad (\text{E.13})$$

$$E[N_2(t)N_3(t-x)] = E[N_3(t)N_2(t-x)] = 0 \quad (\text{E.14})$$

So, one can then write

$$R_N(x) = \begin{pmatrix} E[N_1(t)N_1(t-x)] + 2PE[N_2(t)N_2(t-x)] \\ + \beta^2 E[N_3(t)N_3(t-x)] + 2P\beta^2 E[N_4(t)N_4(t-x)] \\ - \beta(E[N_1(t)N_3(t-x)] + E[N_3(t)N_1(t-x)]) \\ - 2P\beta E[N_2(t)N_4(t-x)] + E[N_4(t)N_2(t-x)] \end{pmatrix} \quad (\text{E.15})$$

## E.2 Loop Gain

The discriminator gain can be calculated using:

$$K_{EMLP}^{ASPeCT} = \left. \frac{d_{EMLP}^{ASPeCT}}{d\varepsilon_\tau} \right|_{\varepsilon_\tau=0} \quad (\text{E.16})$$

$$K_{EMLP}^{ASPeCT} = P \left( \begin{array}{c} \left( \tilde{R}_B \left( -\frac{\delta}{2} \right) \frac{d\tilde{R}_B \left( \varepsilon_\tau - \frac{\delta}{2} \right)}{d\varepsilon_\tau} - \tilde{R}_B \left( \frac{\delta}{2} \right) \frac{d\tilde{R}_B \left( \varepsilon_\tau + \frac{\delta}{2} \right)}{d\varepsilon_\tau} \right) \\ -\beta \left( \tilde{R}_{BP} \left( -\frac{\delta}{2} \right) \frac{d\tilde{R}_{BP} \left( \varepsilon_\tau - \frac{\delta}{2} \right)}{d\varepsilon_\tau} - \tilde{R}_{BP} \left( \frac{\delta}{2} \right) \frac{d\tilde{R}_{BP} \left( \varepsilon_\tau + \frac{\delta}{2} \right)}{d\varepsilon_\tau} \right) \end{array} \right) \Bigg|_{\varepsilon_\tau=0} \quad (\text{E.17})$$

We can deduce that:

$$\left( \begin{array}{c} \tilde{R}_B \left( -\frac{\delta}{2} \right) \frac{d\tilde{R}_B \left( \varepsilon_\tau - \frac{\delta}{2} \right)}{d\varepsilon_\tau} \\ -\tilde{R}_B \left( \frac{\delta}{2} \right) \frac{d\tilde{R}_B \left( \varepsilon_\tau + \frac{\delta}{2} \right)}{d\varepsilon_\tau} \end{array} \right) \Bigg|_{\varepsilon_\tau=0} = 4\pi \left( \tilde{R}_B \left( \frac{\delta}{2} \right) \left( \int_{-\infty}^{+\infty} fH(f) |F_B(f)|^2 \sin(\pi fd) df \right) \right) \quad (\text{E.18})$$

Following the exact same procedure, we obtain:

$$\left( \begin{array}{c} \tilde{R}_{BP}\left(-\frac{\delta}{2}\right) \frac{d\tilde{R}_{BP}\left(\varepsilon_\tau - \frac{\delta}{2}\right)}{d\varepsilon_\tau} \\ -\tilde{R}_{BP}\left(\frac{\delta}{2}\right) \frac{d\tilde{R}_{BP}\left(\varepsilon_\tau + \frac{\delta}{2}\right)}{d\varepsilon_\tau} \end{array} \right)_{\varepsilon_\tau=0} = 4\pi \left( \tilde{R}_{BP}\left(\frac{\delta}{2}\right) \left( \int_{-\infty}^{+\infty} jfH(f)F_B(f)F_P^*(f)\cos(\pi fd)df \right) \right) \quad (\text{E.19})$$

So,

$$K_{EMLP}^{ASPeCT} = 4\pi P \left( \begin{array}{c} \tilde{R}_B\left(\frac{\delta}{2}\right) \left( \int_{-B}^{+B} f|F_B(f)|^2 \sin(\pi fd)df \right) \\ -\beta \tilde{R}_{BP}\left(\frac{\delta}{2}\right) \left( \int_{-B}^{+B} jfF_B(f)F_P^*(f)\cos(\pi fd)df \right) \end{array} \right) \quad (\text{E.20})$$

Without going into detailed calculations and using the same model as in Appendix D, one can obtain:

$$S_{N_1}(0) = \frac{N_0^2}{2T_I} \left( \tilde{R}_{F,B}(0) - \tilde{R}_{F,B}^2(\delta) \right) \quad (\text{E.21})$$

$$S_{N_2}(0) = \frac{N_0}{2} \left( \tilde{R}_B^2\left(\frac{\delta}{2}\right) \left( \tilde{R}_{F,B}(0) - \tilde{R}_{F,B}(\delta) \right) \right) \quad (\text{E.22})$$

$$S_{N_3}(0) = \frac{N_0^2}{2T_I} \left( \tilde{R}_{F,P}(0) - \tilde{R}_{F,P}^2(\delta) \right) \quad (\text{E.23})$$

$$S_{N_4}(0) = \frac{N_0}{2} \left( \tilde{R}_{BP}^2\left(\frac{\delta}{2}\right) \left( \tilde{R}_{F,P}(0) + \tilde{R}_{F,P}(\delta) \right) \right) \quad (\text{E.24})$$

$$S_{N_1N_3}(0) = S_{N_3N_1}(0) = -\frac{N_0^2}{2T_I} R_{F,BP}^2(\delta) \quad (\text{E.25})$$

$$S_{N_2N_4}(0) = -S_{N_2N_4}(0) \quad (\text{E.26})$$

### E.3 Conclusion

One can write

$$S_N(0) = \left( \begin{aligned} & \frac{N_0^2}{2T_I} \left( \tilde{R}_{F,B}(0) - \tilde{R}_{F,B}^2(\delta) \right) + 2P \frac{N_0}{2} \left( \tilde{R}_B^2 \left( \frac{\delta}{2} \right) \left( \tilde{R}_{F,B}(0) - \tilde{R}_{F,B}(\delta) \right) \right) \\ & + \beta^2 \frac{N_0^2}{2T_I} \left( \tilde{R}_{F,P}(0) - \tilde{R}_{F,P}^2(\delta) \right) \\ & + 2P\beta^2 \frac{N_0}{2} \left( \tilde{R}_{BP}^2 \left( \frac{\delta}{2} \right) \left( \tilde{R}_{F,P}(0) + \tilde{R}_{F,P}(\delta) \right) \right) \\ & + 2\beta \frac{N_0^2}{2T_I} R_{F,BP}^2(\delta) \end{aligned} \right) \quad (\text{E.27})$$

Re-arranging, it gives:

$$S_N(0) = \left( \begin{aligned} & PN_0 \left( \begin{aligned} & \tilde{R}_B^2 \left( \frac{\delta}{2} \right) \left( \tilde{R}_{F,B}(0) - \tilde{R}_{F,B}(\delta) \right) \\ & + \beta^2 \left( \tilde{R}_{BP}^2 \left( \frac{\delta}{2} \right) \left( \tilde{R}_{F,P}(0) + \tilde{R}_{F,P}(\delta) \right) \right) \end{aligned} \right) \\ & + \frac{N_0^2}{2T_I} \left( \begin{aligned} & \left( \tilde{R}_{F,B}(0) - \tilde{R}_{F,B}^2(\delta) \right) + \beta^2 \left( \tilde{R}_{F,P}(0) - \tilde{R}_{F,P}^2(\delta) \right) \\ & + 2\beta R_{F,BP}^2(\delta) \end{aligned} \right) \end{aligned} \right) \quad (\text{E.28})$$

So,

$$\sigma_{\varepsilon_\tau}^2 = \left( \frac{B_L(1-0.5B_L T_I) \left( \tilde{R}_B^2 \left( \frac{\delta}{2} \right) \left( \tilde{R}_{F,B}(0) - \tilde{R}_{F,B}(\delta) \right) + \beta^2 \left( \tilde{R}_{BP}^2 \left( \frac{\delta}{2} \right) \left( \tilde{R}_{F,P}(0) + \tilde{R}_{F,P}^2(\delta) \right) \right) \right)}{2 \frac{P}{N_0} \left( 2\pi \left( \tilde{R}_B \left( \frac{\delta}{2} \right) \int_{-B}^{+B} |F_B(f)|^2 \sin(\pi f d) df \right) - \beta \tilde{R}_{BP} \left( \frac{\delta}{2} \right) \int_{-B}^{+B} j f F_B(f) F_P^*(f) \cos(\pi f d) df \right)} \right)^2 \times \left( 1 + \frac{\left( \tilde{R}_{F,B}(0) - \tilde{R}_{F,B}^2(\delta) \right) + \beta^2 \left( \tilde{R}_{F,P}(0) - \tilde{R}_{F,P}^2(\delta) \right) + 2\beta \tilde{R}_{F,BP}^2(\delta)}{\frac{P}{N_0} T_I \left( \tilde{R}_B^2 \left( \frac{\delta}{2} \right) \left( \tilde{R}_{F,B}(0) - \tilde{R}_{F,B}(\delta) \right) + \beta^2 \left( \tilde{R}_{BP}^2 \left( \frac{\delta}{2} \right) \left( \tilde{R}_{F,P}(0) + \tilde{R}_{F,P}^2(\delta) \right) \right) \right)} \right) \right) \quad (\text{E.29})$$

Assuming a front-end filter with unity amplitude within  $\pm B$  Hz, and null elsewhere, the analytical expression can be calculated equal to:

$$\sigma_{\varepsilon_\tau}^2 = \left( \begin{array}{l} B_L(1-0.5B_L T_I) \left( \begin{array}{l} \left( \int_{-\infty}^{+\infty} |F_B(f)|^2 \cos(\pi f d) df \right)^2 \left( \int_{-\infty}^{+\infty} |F_B(f)|^2 \sin^2(\pi f d) df \right) \\ + \beta^2 \left( \int_{-B}^{+B} jF_B(f)F_P^*(f)\sin(\pi f d) df \right)^2 \left( \int_{-B}^{+B} |F_P(f)|^2 \cos^2(\pi f d) df \right) \end{array} \right) \\ \frac{P}{N_0} \left( \begin{array}{l} 2\pi \left( \begin{array}{l} \left( \int_{-B}^{+B} |F_B(f)|^2 \cos(\pi f d) df \right) \times \left( \int_{-B}^{+B} f |F_B(f)|^2 \sin(\pi f d) df \right) - \\ \beta \left( \int_{-B}^{+B} F_B(f)F_P^*(f)\sin(\pi f d) df \right) \times \left( \int_{-B}^{+B} f F_B(f)F_P^*(f)\cos(\pi f d) df \right) \end{array} \right) \end{array} \right)^2 \\ \left( \begin{array}{l} \left( \int_{-B}^{+B} |F_B(f)|^2 \sin^2(\pi f d) df \right) \left( \int_{-B}^{+B} |F_B(f)|^2 \cos^2(\pi f d) df \right) \\ + \beta^2 \left( \int_{-B}^{+B} |F_P(f)|^2 \sin^2(\pi f d) df \right) \left( \int_{-B}^{+B} |F_P(f)|^2 \cos^2(\pi f d) df \right) \\ + \frac{\beta}{2} \left( \int_{-B}^{+B} jF_B(f)F_P^*(f)\sin(2\pi f d) df \right)^2 \end{array} \right) \\ \times 1 + \frac{P}{N_0} T_I \left( \begin{array}{l} \left( \int_{-\infty}^{+\infty} |F_B(f)|^2 \cos(\pi f d) df \right)^2 \left( \int_{-\infty}^{+\infty} |F_B(f)|^2 \sin^2(\pi f d) df \right) \\ + \beta^2 \left( \int_{-B}^{+B} jF_B(f)F_P^*(f)\sin(\pi f d) df \right)^2 \left( \int_{-B}^{+B} |F_P(f)|^2 \cos^2(\pi f d) df \right) \end{array} \right) \end{array} \right)$$

(E.30)