Mohammad Zahidul Hasan Bhuiyan

Analysis of Multipath Mitigation Techniques for Satellite-based Positioning Applications

Julkaisu 981 • Publication 981

Tampere 2011
Mohammad Zahidul Hasan Bhuiyan

Analysis of Multipath Mitigation Techniques for Satellite-based Positioning Applications

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB222, at Tampere University of Technology, on the 6th of September 2011, at 12 noon.
Supervisor
Elena Simona Lohan, Dr. Tech., Docent
Department of Communications Engineering
Tampere University of Technology
Tampere, Finland

Pre-examiners
Fabio Dovis, Ph.D., Assistant Professor
Department of Electronics
Faculty of Information and Communication Technology
Politecnico di Torino
Turin, Italy

Heidi Kuusniemi, Dr. Tech.
Chief Research Scientist
Finnish Geodetic Institute
Masala, Finland

Opponent
Gonzalo Seco Granados, Ph.D., Associate Professor
Department of Telecommunications and Systems Engineering
Universitat Autonoma de Barcelona
Barcelona, Spain
Abstract

Multipath remains a dominant source of ranging errors in any Global Navigation Satellite System (GNSS), such as the Global Positioning System (GPS) or the developing European satellite navigation system Galileo. Multipath is undesirable in the context of GNSS, since the reception of multipath can create significant distortion to the shape of the correlation function used in the time delay estimate of a Delay Locked Loop (DLL) of a navigation receiver, leading to an error in the receiver’s position estimate. Therefore, in order to mitigate the impact of multipath on a navigation receiver, the multipath problem has been approached from several directions, including the development of novel signal processing techniques. Many of these techniques rely on modifying the tracking loop discriminator (i.e., the DLL and its enhanced variants) in order to make it resistant to multipath, but their performance in severe multipath scenarios is still rather limited. In this thesis, the Author particularly addresses the challenge of overcoming the difficulties due to multipath propagation by developing several novel correlation-based multipath mitigation techniques, ranging from simple DLL based approach to advanced multi-correlator based solution, whichever is appropriate according to the requirements of positioning applications (i.e., low-cost simpler implementation versus better accuracy). The proposed novel multipath mitigation techniques can be categorized into three major categories considering their implementation complexity: i. the advanced technique, which requires many correlators and have relatively complex implementation; ii. the simple technique, which requires only a few correlators; and iii. the combined technique, which is a combination of two other techniques and has moderate complexity.

The proposed advanced multipath mitigation techniques include Peak Tracking (PT) and its variants based on 2nd order Differentiation (Diff2) and Teager Kaiser (TK) operator, non-coherent Multipath Estimating Delay Lock Loop (MEDLL) and Reduced Search Space Maximum Likelihood (RSSML) delay estimator. The development of these advanced multipath mitigation tech-
Techniques are justified from the fact that they provide better tracking accuracy in harsh multipath environments, for example, in urban canyons, at a cost of increased implementation complexity. Among all these techniques, RSSML offers the best multipath mitigation performance in moderate-to-high C/N\textsubscript{0} scenarios, as verified by the simulations in multipath fading channels. Therefore, RSSML along with other advanced techniques proposed in this thesis can be considered as excellent candidates for implementation in professional GNSS receivers, especially when the tracking accuracy is a concern.

The proposed simple multipath mitigation technique includes a Slope-Based Multipath Estimator (SBME), which requires a-priori information about the slope of the correlation function and an additional correlator (as compared to a traditional DLL) to estimate the multipath error. Simulation results show that SBME has superior multipath mitigation performance to the well-known narrow Early-Minus-Late (nEML) DLL in tested environments.

The proposed combined techniques include a C/N\textsubscript{0}-based two-stage delay tracker and a combined TK operator with nEML DLL. The motivation for having a combined approach is to ensure a better tracking performance with a reasonable implementation complexity than each single combining method that is used to form the combined multipath mitigation technique. The C/N\textsubscript{0}-based two-stage delay tracker offers a better tracking accuracy than its individual counterpart in multipath channel, as validated by the simulations in an open source TUT (Tampere University of Technology) Galileo E1 signal simulator. It also alleviates the problem of false tracking involved in High Resolution Correlator (HRC). Hence, the C/N\textsubscript{0}-based two-stage delay tracker can be considered as a viable solution for legacy GNSS receivers which are currently using nEML or HRC as their delay locked loop discriminator.

The multipath performance of all the novel and the state-of-the-art techniques is analyzed for three different GNSS signals, namely legacy GPS L1 C/A signal, Galileo E1 Open Service (OS) signal and the modernized GPS L1C signal. However, this does not restrict the applicability of these techniques in the context of other GNSS signals (for example, Galileo E5 signal) as long as they are adapted considering the signals’ auto-correlation properties.

This thesis is structured in the form of a compound, including an introductory part in the research field and a collection of eight original publications of the Author, where the main contribution of the thesis lies.
Preface

This Ph.D. dissertation has been carried out at the Department of Communications Engineering (DCE) in Tampere University of Technology (TUT), as part of the Tekes funded research projects “Advanced Techniques for Personal Navigation (ATENA)” and “Future GNSS Applications and Techniques (FUGAT)”, of the Academy of Finland funded research project “Digital Signal Processing Algorithms for Indoor Positioning Systems”, of the EU FP6 research project “Galileo Receiver for Mass Market (GREAT)”, and of the EU FP7 research project “Galileo Ready Advanced Mass Market Receiver (GRAMMAR)”. Being a postgraduate student in Tampere Doctoral Programme in Information Sciences and Engineering (TISE), I have been receiving funding from TISE for three consecutive years from 2009 till 2011 for my doctoral studies along with some time-to-time conference travels. I gratefully acknowledge the support I have been receiving from TISE during these years. I would also like to acknowledge Academy of Finland, Tekes, European Union, Nokia Foundation, Ulla Tuomisen Saatio and Satellite Navigation University Network (SNUN) for their financial support at various stages of my Ph.D. studies.

I am very much fortunate to have Dr. Tech. Elena Simona Lohan as my supervisor. I would like to express my deepest gratitude to her for invaluable guidance, continuous support, patience and encouragement throughout the course of this work. I can still remember her positive encouraging words at times when I was scared of missing a hard deadline. She is so helpful, considerate and innovative that our group executes the work that have assigned to us with full confidence, believing that she is always there to help us in difficult situations. Thank you, Ma’am, for being my academic supervisor.

I would like to thank Assistant Professor Fabio Dovis from the Department of Electronics, Politecnico di Torino, Italy and Dr. Tech. Heidi Kuusniemi from the Finnish Geodetic Institute, Finland, the reviewers of this thesis, for their valuable and constructive feedback and competent judgment. I also
thank Associate Professor Gonzalo Seco Granados for kindly agreeing to act as the opponent in the public defense of the dissertation.

I would like to express my sincere gratitude to Prof. Markku Renfors for offering me a position in DCE and also for guiding me during the initial months of my Ph.D. studies.

I thank Prof. Gerard Lachapelle for his guidance and support during my visit in the Positioning, Location And Navigation (PLAN) group in 2007. Also, I would like to thank all my colleagues and friends in Calgary, Canada for their cordial help and support during my ten months stay over there.

I express my appreciation to all my present and former colleagues at DCE for creating such a pleasant and friendly working atmosphere. In particular, I would like to thank Prof. Mikko Valkama, Dr. Tech. Abdelmonaem Lakhzouri, Dr. Tech. Ridha Hamila, Dr. Tech. Adina Burian, Xuan Hu, Danai Skournetou, Jie Zhang, Elina Laitinen, Heikki Hurskainen, Farzan Samad, Najmul Islam, Alexandru Rusu Casandra, Jukka Talvitie and Bashir Siddiqui.

Warm thanks are due to Ulla Siltaloppi, Tarja Erlaukko, Sari Kinnari and Elina Orava for their kind help with practical matters and friendly support. Specially, I would like to mention Ulla Siltaloppi’s name for her cordial, friendly and generous support in legal matters.

I wish to extend my heartfelt thanks to the whole Bangladeshi community in Tampere for the enjoyable moments we spent together in numerous gatherings and parties. It has been a real pleasure to stay in Tampere for six years which could not be possible without you!

In this occasion, I would like to remember my parents Md. A. Satter Bhuiyan and Mrs. Lily Akter, who were the first mentors in my life to teach me the value of life, education, discipline and knowledge. Perhaps they would be the happiest persons on the Earth seeing me achieving this honor!

Finally, I would like to thank my wife Mehjabin Sultana for all her love, affection, patience and understanding during the last five years, which I believe, have been the main motivating factors to keep me in the right track. I dedicate this thesis to my daughter Lilya Doyeeta Bhuiyan, who redefines our life as beautiful as it could be.

Tampere, August 2011

Mohammad Zahidul Hasan Bhuiyan
## Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Abstract</td>
<td>i</td>
</tr>
<tr>
<td></td>
<td>Preface</td>
<td>iii</td>
</tr>
<tr>
<td></td>
<td>Table of Contents</td>
<td>v</td>
</tr>
<tr>
<td></td>
<td>List of Publications</td>
<td>ix</td>
</tr>
<tr>
<td></td>
<td>List of Abbreviations</td>
<td>xi</td>
</tr>
<tr>
<td></td>
<td>List of Principal Symbols</td>
<td>xv</td>
</tr>
<tr>
<td></td>
<td>List of Figures</td>
<td>xix</td>
</tr>
<tr>
<td>1</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>State-of-the-Art</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Scope of the Thesis</td>
<td>5</td>
</tr>
<tr>
<td>1.3</td>
<td>Thesis Contributions</td>
<td>6</td>
</tr>
<tr>
<td>1.4</td>
<td>Thesis Outline</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>Global Navigation Satellite Systems</td>
<td>9</td>
</tr>
<tr>
<td>2.1</td>
<td>Brief History of Satellite Navigation System</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Fundamentals of Satellite-based Positioning</td>
<td>10</td>
</tr>
<tr>
<td>2.3</td>
<td>Overview of GPS</td>
<td>10</td>
</tr>
<tr>
<td>2.4</td>
<td>Overview of Galileo</td>
<td>12</td>
</tr>
<tr>
<td>2.5</td>
<td>Overview of GLONASS</td>
<td>14</td>
</tr>
<tr>
<td>2.6</td>
<td>Other Satellite Navigation Systems</td>
<td>15</td>
</tr>
<tr>
<td>2.7</td>
<td>GNSS Applications</td>
<td>15</td>
</tr>
</tbody>
</table>
# CONTENTS

6.3 Proposed Simple Slope-based Technique .......................... 48  
6.4 Proposed Combined Techniques ................................. 49  
  6.4.1 C/N₀-based Two-Stage Delay Tracker ....................... 49  
  6.4.2 TK operator combined with a nEML DLL ................. 50  

7 Experimental Analysis ............................................. 51  
  7.1 Semi-analytical Simulation ...................................... 51  
  7.2 Matlab-based Simulation ......................................... 54  
  7.3 Simulink-based TUT Galileo E1 Signal Simulation ........ 56  
    7.3.1 Transmitter block .......................................... 57  
    7.3.2 Channel block .............................................. 58  
    7.3.3 Front-end filter block .................................... 58  
    7.3.4 Tracking block ............................................. 58  
    7.3.5 Simulation Results ......................................... 59  

8 Summary of Publications ............................................ 61  
  8.1 Overview of the Publication Results ......................... 61  
  8.2 Author’s Contribution to the Publications .................. 65  

9 Conclusions ........................................................... 69  

Bibliography ........................................................... 73  

Publications ........................................................... 87
List of Publications

This thesis is based on the following publications. In the text, these publications are referred to as [P1], [P2], ..., and [P8].


List of Abbreviations

This is a list of the most important and recurrently appearing abbreviations in this thesis.

ADC  Analog-to-Digital Conversion  
AGC  Automatic Gain Control  
AME  Average Mean Error  
ATENA  Advanced Techniques for Personal Navigation  
APME  A-Posteriori Multipath Estimation  
AWGN  Additive White Gaussian Noise  
AltBOC  Alternative Binary Offset Carrier  
BW  BandWidth  
BOC  Binary Offset Carrier  
BPSK  Binary Phase Shift Keying  
C/A  Coarse/Acquisition  
CBOC  Composite Binary Offset Carrier  
CDMA  Code Division Multiple Access  
C/N0  Carrier-to-Noise density ratio  
CosBOC  Cosine Binary Offset Carrier  
CS  Commercial Service  
DCE  Department of Communication Engineering  
Diff2  2nd order Differentiation  
DLL  Delay Locked Loop  
DoD  Department of Defense  
DS  Direct Sequence  
DS-CDMA  Direct Sequence - Code Division Multiple Access  
DSSS  Direct Sequence Spread Spectrum  
E1/E2  Early1 / Early2  
ELS  Early-Late-Slope  
EML  Early-Minus-Late
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EU</td>
<td>European Union</td>
</tr>
<tr>
<td>ESA</td>
<td>European Space Agency</td>
</tr>
<tr>
<td>FCC</td>
<td>Federal Communication Commission</td>
</tr>
<tr>
<td>FDE</td>
<td>Fault Detection and Exclusion</td>
</tr>
<tr>
<td>FDMA</td>
<td>Frequency Division Multiple Access</td>
</tr>
<tr>
<td>FIR</td>
<td>Finite Impulse Response</td>
</tr>
<tr>
<td>FLL</td>
<td>Frequency Locked Loop</td>
</tr>
<tr>
<td>FUGAT</td>
<td>Future GNSS Applications and Techniques</td>
</tr>
<tr>
<td>GAGAN</td>
<td>GPS Aided Geo Augmented Navigation</td>
</tr>
<tr>
<td>GLONASS</td>
<td>Global’naya Navigatsionnaya Sputknikkovaya Sistema</td>
</tr>
<tr>
<td>GNSS</td>
<td>Global Navigation Satellite System</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GRAMMAR</td>
<td>Galileo Ready Advanced Mass Market Receiver</td>
</tr>
<tr>
<td>GREAT</td>
<td>Galileo Receiver for Mass Market</td>
</tr>
<tr>
<td>HRC</td>
<td>High Resolution Correlator</td>
</tr>
<tr>
<td>I&amp;D</td>
<td>Integrate and Dump</td>
</tr>
<tr>
<td>IELS</td>
<td>Improved Early-Late-Slope</td>
</tr>
<tr>
<td>IOC</td>
<td>Initial Operational Capability</td>
</tr>
<tr>
<td>IOV</td>
<td>In-Orbit Validation</td>
</tr>
<tr>
<td>kHz</td>
<td>kiloHertz</td>
</tr>
<tr>
<td>km</td>
<td>kilometers</td>
</tr>
<tr>
<td>LOS</td>
<td>Line-Of-Sight</td>
</tr>
<tr>
<td>LEO</td>
<td>Low Earth Orbit</td>
</tr>
<tr>
<td>MBOC</td>
<td>Multiplexed Binary Offset Carrier</td>
</tr>
<tr>
<td>MEDLL</td>
<td>Multipath Estimating Delay Locked Loop</td>
</tr>
<tr>
<td>MEE</td>
<td>Multipath Error Envelope</td>
</tr>
<tr>
<td>MEO</td>
<td>Medium Earth Orbit</td>
</tr>
<tr>
<td>MET</td>
<td>Multipath Elimination Technique</td>
</tr>
<tr>
<td>MF</td>
<td>Matched Filter</td>
</tr>
<tr>
<td>MGD</td>
<td>Multiple Gate Delay</td>
</tr>
<tr>
<td>MHz</td>
<td>MegaHertz</td>
</tr>
<tr>
<td>MMSE</td>
<td>Minimum Mean Square Error</td>
</tr>
<tr>
<td>MMT</td>
<td>Multipath Mitigation Technique</td>
</tr>
<tr>
<td>ms</td>
<td>milliseconds</td>
</tr>
<tr>
<td>MTLL</td>
<td>Mean-Time-to-Lose-Lock</td>
</tr>
<tr>
<td>NC</td>
<td>Narrow Correlator</td>
</tr>
<tr>
<td>NCO</td>
<td>Numerically Controlled Oscillator</td>
</tr>
<tr>
<td>NAVSAT</td>
<td>Navy Navigation Satellite System</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>NAVSTAR</td>
<td>Navigation System by Timing And Ranging</td>
</tr>
<tr>
<td>nEML</td>
<td>narrow Early-Minus-Late</td>
</tr>
<tr>
<td>NLOS</td>
<td>Non-Line-Of-Sight</td>
</tr>
<tr>
<td>NWPR</td>
<td>Narrowband Wideband Power Ratio</td>
</tr>
<tr>
<td>OS</td>
<td>Open Service</td>
</tr>
<tr>
<td>P</td>
<td>Precision</td>
</tr>
<tr>
<td>PAC</td>
<td>Pulse Aperture Correlator</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistant</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Density Function</td>
</tr>
<tr>
<td>PLL</td>
<td>Phase Locked Loop</td>
</tr>
<tr>
<td>PPS</td>
<td>Precise Positioning Service</td>
</tr>
<tr>
<td>PRN</td>
<td>Pseudo-Random Noise</td>
</tr>
<tr>
<td>PRS</td>
<td>Public Regulated Service</td>
</tr>
<tr>
<td>PSD</td>
<td>Power Spectral Density</td>
</tr>
<tr>
<td>PT</td>
<td>Peak Tracking</td>
</tr>
<tr>
<td>PVT</td>
<td>Position, Velocity and Time</td>
</tr>
<tr>
<td>QPSK</td>
<td>Quadrature Phase Shift Keying</td>
</tr>
<tr>
<td>QZSS</td>
<td>Quasi-Zenith Satellite System</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>RAE</td>
<td>Running Average Error</td>
</tr>
<tr>
<td>RAIM</td>
<td>Receiver Autonomous Integrity Monitoring</td>
</tr>
<tr>
<td>RHCP</td>
<td>Right Hand Circular Polarized</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root-Mean-Square Error</td>
</tr>
<tr>
<td>RSSMML</td>
<td>Reduced Search Space Maximum Likelihood</td>
</tr>
<tr>
<td>SAR</td>
<td>Search And Rescue Service</td>
</tr>
<tr>
<td>SARPRS</td>
<td>Search And Rescue Public Regulated Service</td>
</tr>
<tr>
<td>SBME</td>
<td>Slope-Based Multipath Estimator</td>
</tr>
<tr>
<td>SC</td>
<td>Strobe Correlator</td>
</tr>
<tr>
<td>SD</td>
<td>Slope Differential</td>
</tr>
<tr>
<td>SinBOC</td>
<td>Sine Binary Offset Carrier</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>SoL</td>
<td>Safety of Life service</td>
</tr>
<tr>
<td>SPS</td>
<td>Standard Positioning Service</td>
</tr>
<tr>
<td>SV</td>
<td>Satellite Vehicle</td>
</tr>
<tr>
<td>TEC</td>
<td>Total Electron Content</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td>TK</td>
<td>Teager Kaiser</td>
</tr>
<tr>
<td>TMBOC</td>
<td>Time Multiplexed Binary Offset Carrier</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>ToA</td>
<td>Time-of-Arrival</td>
</tr>
<tr>
<td>TrEC</td>
<td>Tracking Error Compensator</td>
</tr>
<tr>
<td>TUT</td>
<td>Tampere University of Technology</td>
</tr>
<tr>
<td>US</td>
<td>United States</td>
</tr>
<tr>
<td>UHF</td>
<td>Ultra High Frequency</td>
</tr>
<tr>
<td>VC</td>
<td>Vision Correlator</td>
</tr>
<tr>
<td>VHF</td>
<td>Very High Frequency</td>
</tr>
</tbody>
</table>
List of Principal Symbols

This is a list of the principal symbols and notations used throughout the thesis.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha, \beta$</td>
<td>Weighting factors used to combine E1 channel</td>
</tr>
<tr>
<td>$\alpha_l$</td>
<td>Amplitude of the $l$-th path</td>
</tr>
<tr>
<td>$\vec{\alpha}$</td>
<td>Vector of path amplitudes</td>
</tr>
<tr>
<td>$\delta(\cdot)$</td>
<td>Dirac pulse</td>
</tr>
<tr>
<td>$\otimes$</td>
<td>Convolution operator</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Decision threshold</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Carrier phase</td>
</tr>
<tr>
<td>$\Omega_0$</td>
<td>Loop filter natural radian frequency</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Decaying power delay profile coefficient</td>
</tr>
<tr>
<td>$\eta(\cdot)$</td>
<td>Additive White Gaussian Noise</td>
</tr>
<tr>
<td>$\tau_l$</td>
<td>Delay of the $l$th path</td>
</tr>
<tr>
<td>$\vec{\tau}$</td>
<td>Vector of path delays</td>
</tr>
<tr>
<td>$\hat{\tau}$</td>
<td>Estimated code delay</td>
</tr>
<tr>
<td>$\hat{\tau}_1$</td>
<td>Estimated first path delay</td>
</tr>
<tr>
<td>$\tau_d$</td>
<td>Dwell time</td>
</tr>
<tr>
<td>$\tau_W$</td>
<td>Code delay window in chips</td>
</tr>
<tr>
<td>$\Delta$</td>
<td>Correlator spacing</td>
</tr>
<tr>
<td>$\Delta_{EL}$</td>
<td>Early-late correlator spacing</td>
</tr>
<tr>
<td>$\Delta\Delta$</td>
<td>Double Delta technique</td>
</tr>
<tr>
<td>$\Delta_f$</td>
<td>Frequency bin width</td>
</tr>
<tr>
<td>$\Psi_{TK}(\cdot)$</td>
<td>TK operator</td>
</tr>
<tr>
<td>$\theta_l$</td>
<td>Phase of the $l$th path</td>
</tr>
<tr>
<td>$\vec{\theta}$</td>
<td>Vector of path phases</td>
</tr>
<tr>
<td>$b_n$</td>
<td>Data bit</td>
</tr>
<tr>
<td>$\hat{b}_n$</td>
<td>Estimated data bit</td>
</tr>
<tr>
<td>$c$</td>
<td>Speed of light; $c = 299,792,458$ m/s</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$c_{k,n}$</td>
<td>$k^{th}$ chip corresponding to $n^{th}$ symbol</td>
</tr>
<tr>
<td>$e_{E1B}(\cdot)$</td>
<td>Galileo E1B signal component</td>
</tr>
<tr>
<td>$e_{E1C}(\cdot)$</td>
<td>Galileo E1C signal component</td>
</tr>
<tr>
<td>$f_s$</td>
<td>Sampling frequency</td>
</tr>
<tr>
<td>$f_{IF}$</td>
<td>Intermediate frequency</td>
</tr>
<tr>
<td>$f_D$</td>
<td>Doppler shift</td>
</tr>
<tr>
<td>$\hat{f}_D$</td>
<td>Estimated Doppler frequency</td>
</tr>
<tr>
<td>$f_{\text{chip}}$</td>
<td>Chip rate</td>
</tr>
<tr>
<td>$f_{\text{ref}}$</td>
<td>Reference frequency of 1.023 MHz</td>
</tr>
<tr>
<td>$f_{sc}$</td>
<td>Subcarrier frequency</td>
</tr>
<tr>
<td>$p, q$</td>
<td>BOC modulation parameters</td>
</tr>
<tr>
<td>$p_{\text{MOD}}(\cdot)$</td>
<td>Modulation waveform of type MOD</td>
</tr>
<tr>
<td>$p_{nc}$</td>
<td>Power index used for non-coherent summation</td>
</tr>
<tr>
<td>$m$</td>
<td>Nakagami-$m$ fading parameter</td>
</tr>
<tr>
<td>$c(\cdot)$</td>
<td>Navigation data after spreading</td>
</tr>
<tr>
<td>$c_\delta(\cdot)$</td>
<td>Code signal without pulse shaping</td>
</tr>
<tr>
<td>$r^\cdot$</td>
<td>Received signal</td>
</tr>
<tr>
<td>$r_{E1}^\cdot$</td>
<td>Galileo E1 received signal</td>
</tr>
<tr>
<td>$s_1(\cdot), s_2(\cdot)$</td>
<td>SinBOC components</td>
</tr>
<tr>
<td>$s_{CBOC}(\cdot)$</td>
<td>CBOC modulated waveform</td>
</tr>
<tr>
<td>$s_{E1}(\cdot)$</td>
<td>Galileo E1 signal</td>
</tr>
<tr>
<td>$s_{MBOC}(\cdot)$</td>
<td>MBOC modulated waveform</td>
</tr>
<tr>
<td>$s_{\text{SinBOC}}(\cdot)$</td>
<td>SinBOC modulated waveform</td>
</tr>
<tr>
<td>$s_{\text{SinBOC}(1,1),\text{held}}(\cdot)$</td>
<td>SinBOC(1,1) modulated waveform with hold clock</td>
</tr>
<tr>
<td>$s_{\text{SinBOC}(6,1)}(\cdot)$</td>
<td>SinBOC(6,1) modulated waveform</td>
</tr>
<tr>
<td>$w_1, w_2$</td>
<td>Weighting factors to form CBOC modulation</td>
</tr>
<tr>
<td>$x_{ct}$</td>
<td>Constant successive path spacing</td>
</tr>
<tr>
<td>$x_{\text{data}}(\cdot)$</td>
<td>Data sequence after spreading</td>
</tr>
<tr>
<td>$x_{\text{PRN},n}(\cdot)$</td>
<td>PRN code sequence of $n^{th}$ data symbol</td>
</tr>
<tr>
<td>$x_{\text{SinBOC}}(\cdot)$</td>
<td>Data sequence after spreading and modulation</td>
</tr>
<tr>
<td>$x(\cdot)$</td>
<td>Transmitted signal from one satellite</td>
</tr>
<tr>
<td>$x_{\text{ref}}(\cdot)$</td>
<td>Reference PRN code at the receiver</td>
</tr>
<tr>
<td>$y(\cdot)$</td>
<td>Discrete signal</td>
</tr>
<tr>
<td>$\mathbf{E}(\cdot)$</td>
<td>Expectation operator</td>
</tr>
<tr>
<td>$\text{AME}(\cdot)$</td>
<td>Absolute mean delay error</td>
</tr>
<tr>
<td>$B_W$</td>
<td>Code epoch bandwidth</td>
</tr>
<tr>
<td>$BW$</td>
<td>Front-end bandwidth (double-sided)</td>
</tr>
<tr>
<td>$B_n$</td>
<td>Loop filter bandwidth</td>
</tr>
</tbody>
</table>
\( C/N_0 \)  
Carrier-to-Noise density ratio

\( D(\cdot) \)  
Discriminator function

\( E_b \)  
Bit energy

\( I \)  
In-phase channel

\( Q \)  
Quad-phase channel

\( L \)  
Number of path(s)

\( L_{\text{max}} \)  
Maximum number of path(s)

\( M \)  
Number of correlators

\( \text{MOD} \)  
Type of modulation (i.e., BPSK, CBOC etc.)

\( N_0 \)  
Noise power in 1 kHz bandwidth

\( N_B \)  
BOC modulation order

\( N_{B1} \)  
BOC modulation order for SinBOC(1,1)

\( N_{B2} \)  
BOC modulation order for SinBOC(6,1)

\( N_c \)  
Coherent integration length in milliseconds

\( N_{nc} \)  
Non-coherent integration length in blocks

\( N_s \)  
Oversampling factor

\( N_{\text{random}} \)  
Number of random realizations

\( P_d \)  
Probability of detection

\( P_{fa} \)  
Probability of false alarm

\( P_{T_B}(\cdot) \)  
Rectangular pulse shape

\( P, Q \)  
Blocks of code symbols used in MBOC

\( \mathcal{R}(\cdot) \)  
Running average delay error

\( \mathcal{R}_c(\cdot) \)  
Average coherent correlation function

\( \mathcal{R}_{nc}(\cdot) \)  
Non-coherently averaged correlation function

\( \mathcal{R}_{rx}(\cdot) \)  
Received correlation function

\( \mathcal{R}_{\text{MOD}} \)  
Auto-correlation function with type MOD

\( \text{RMSE}_{\text{chips}} \)  
RMSE in chips

\( \text{RMSE}_m \)  
RMSE in meters

\( R_{s,X} \)  
Sub-carrier rate corresponding to channel \( X \)

\( S_F \)  
Spreading factor

\( T_c \)  
Chip period

\( T_{coh} \)  
Predetection integration time

\( T_{sym} \)  
Code symbol period

\( X \)  
Galileo E1B or E1C channel
## List of Figures

4.1 A conventional GNSS receiver block diagram. .................. 24  
4.2 A simplified signal acquisition block diagram. ................. 27  
4.3 Block diagram for multi-correlator based DLL implementation. 29  

5.1 Received correlation function in two path static channel model, path delays: [0 0.5] chips, path amplitudes: [0 -3] dB, in-phase combination. ................................................. 37  
5.2 Non-coherent correlation functions for different signal modulations. ................................................................. 38  
5.3 Non-coherent correlation functions for different signal modulations in two path static channel. ................................. 39  
5.4 Non-coherent correlation functions for BPSK modulated GPS L1 C/A signal in different front-end bandwidths. ............... 39  

6.1 Generation of competitive peaks for PT(Diff2) technique. .... 46  
6.2 Estimated and received non-coherent correlation functions in two path Rayleigh channel, path delay: [0 0.35] chips, path power: [0 -2] dB, C/N₀: 50 dB-Hz. ................................. 48  
6.3 A non-coherent S-curve for CBOC(-) modulated single path static channel [P6]. ...................................................... 50  

7.1 Running average error curves for CBOC(-) modulated Galileo E1C signal. ......................................................... 54  
7.2 RMSE vs. C/N₀ plot for CBOC(-) modulated Galileo E1C signal in two to five path Rayleigh fading channel. ................. 56  
7.3 TUT Galileo E1 signal simulator (tracking-only model). ....... 57  
7.4 RMSE vs. C/N₀ plots in a two path static channel in a 24.552 MHz double-sided bandwidth [P6]. ............................... 59
9.1 Comparison of multipath mitigation techniques. . . . . . . . . . . . 71
Chapter 1

Introduction

Today, with the remarkable progress in satellite navigation and positioning technology, it is possible to pinpoint the exact location of any user anywhere on the surface of the Earth at any time of day or night. Since its birth in the 1970s, the United States (US) Global Positioning System (GPS), has become the universal satellite navigation system and reached full operational capability in 1990s [60]. This has created a huge monopoly, resulting in technical, political, strategic and economic dependence for millions of users. In recent years, the rapid improvement and competitive price of computing resources have allowed the integration of GPS chips into small autonomous devices such as hand-held GPS receivers, mobile phones and Personal Digital Assistants (PDAs), increasing the speed of its consumption 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, there had been a lot of initiatives during the 1990s, which gave birth to a second generation of Global Navigation Satellite Systems (GNSSs) [63]: the modernization of US GPS, known as GPS II & III; the independent European effort to create its own GNSS, known as Galileo; and the Russian effort to restore the full operational capability of its own navigation system GLONASS (Global’naya Navigatsionnaya Sputnikkovaya Sistema) [47]. In addition, China has also indicated to expand its own regional navigation system, Beidou into a global navigation system, named as Compass.

The GPS III and the European Galileo are currently being finalized and are expected to be commercially available to the public within next couple of years (according to [72] and [27], by 2014). Moreover, the Russian counterpart GLONASS, consisting of 22 operational satellites as of February 2011,
is expected to complete the full constellation of 24 satellites by the end of this year [104]. It has been widely anticipated that once the new European satellite navigation system Galileo is operational, the vast majority of all user receivers sold will be both GPS and Galileo capable. Also, according to [105], GLONASS will introduce new Code Division Multiple Access (CDMA) based civil signals (i.e., GLONASS L1CR and L5R) interoperable with both GPS and Galileo systems. However, it is still unknown to the best of the Author’s knowledge, when these signals will become available.

The benefits of receiving signals from different GNSSs include improved accuracy, integrity, availability and reliability through the use of a single common receiver design, especially in urban environments where the signal reception quality varies a lot [26], [48], [52], [98], [124]. The luxury of having more satellites in conjunction with modernized GNSS signals will provide the potential for a sub-meter level positioning and a shorter initialization time in a standard navigation receiver. Moreover, the user accessing data from multiple satellite systems can continue to operate if one of the systems fails and will benefit from a more reliable signal tracking, also designed for Safety-of-Life (SoL) applications [28], [29], [50]. With such a wide range of new signals and satellite navigation systems, there are still many new design challenges from one end (i.e., the antenna) of the receiver to the other (i.e., the navigation software providing the user with Position, Velocity and Time (PVT) information). A well-documented overview of some of these design challenges can be found in [18], [19]. In this regard, there is always a continuous demand for efficient implementation of digital signal processing techniques in designing a GNSS receiver to fulfill the required quality of service.

1.1 State-of-the-Art

Multipath remains a dominant source of ranging errors in any satellite navigation system. Several approaches have been used in order to reduce the multipath error. Among them, the use of special multipath limiting antennas (i.e., choke ring or multi-beam antennas [102], [103], [126], [127]), the post-processing techniques to reduce carrier multipath [17], [128], the carrier smoothing to reduce code multipath [62], [121], and the code tracking algorithms based on receiver internal correlation technique are the most prominent approaches [21]. In this thesis, the research focus is on the correlation-based multipath mitigation techniques, since they are the most widely used in commercial GNSS receivers. The classical correlation-based code tracking struc-
1.1 State-of-the-Art

ture used in GNSS is based on a feedback delay estimator and is implemented via a feedback loop. The most known feedback delay estimator is the Delay Locked Loop (DLL) or Early-Minus-Late (EML) technique, where two correlators spaced at one chip from each other are used in the receiver in order to form a discriminator function, whose zero crossings determine the path delays of the received signal [2], [10], [16], [36], [37], [68]. The classical EML fails to cope with multipath propagation [21], [112]. Therefore, several enhanced EML-based techniques have been introduced in the literature during the last two decades in order to mitigate the impact of multipath, especially in closely spaced path scenarios. One class of these enhanced EML techniques is based on the idea of narrowing the spacing between the early and late correlators, i.e., narrow EML (nEML) or narrow correlator [21], [55], [80]. The choice of correlator spacing depends on the receiver’s available front-end bandwidth along with the associated sampling frequency [7]. Correlator spacings in the range of 0.05 to 0.2 chips are commercially available for nEML based GPS receivers [14].

Another family of discriminator-based DLL variants proposed for GNSS is the so-called Double-Delta (ΔΔ) technique, which uses more than 3 correlators in the tracking loop (typically, 5 correlators: two early, one in-prompt and two late) [55]. The ΔΔ technique offers better multipath rejection in medium-to-long delay multipath [54], [80] in good Carrier-to-Noise density ratio (C/N₀). Couple of well-known particular cases of ΔΔ technique are the High Resolution Correlator (HRC) [80], the Strobe Correlator (SC) [41], [55], the Pulse Aperture Correlator (PAC) [57] and the modified correlator reference waveform [55], [131]. One other similar tracking structure is the Multiple Gate Delay (MGD) correlator [4], [9], [31], [32], where the number of early and late gates and the weighting factors used to combine them in the discriminator are the parameters of the model, and can be optimized according to the multipath profile as illustrated in [54]. While coping better with the ambiguities of Binary Offset Carrier (BOC) correlation function, the MGD provides slightly better performance than the nEML at the expense of higher complexity and is sensitive to the parameters chosen in the discriminator function (i.e., weights, number of correlators and correlator spacing) [9], [54].

Another tracking structure closely related to ΔΔ technique is the Early1 / Early2 (E1/E2) tracker, initially proposed in [20], and later described in [55]. In E1/E2 tracker, the main purpose is to find a tracking point on the correlation function that is not distorted by multipath. As reported in [55], E1/E2 tracker shows some performance improvement over ΔΔ technique only
for very short delay multipath for GPS L1 Coarse / Acquisition (C/A) signal.

Another feedback tracking structure is the Early-Late-Slope (ELS) [55], which is also known as Multipath Elimination Technique (MET) [122]. The simulation results performed in [55] showed that ELS is outperformed by HRC with respect to Multipath Error Envelopes (MEEs), for both Binary Phase Shift Keying (BPSK) and Sine BOC(1,1) (SinBOC(1,1)) modulated signals.

A new multipath estimation technique, named as A-Posteriori Multipath Estimation (APME), is proposed in [115], which relies on a-posteriori estimation of the multipath error tracking. Multipath error is estimated independently in a multipath estimator module on the basis of the correlation values from the prompt and very late correlators. According to [115], the multipath performance of GPS L1 C/A signal is comparable with that of the Strobe Correlator: slight improvement for very short delays (i.e., delays less than 20 meters), but rather significant deterioration for medium delays.

In [95], a fundamentally different approach is adopted to solve the problem of multipath in the context of GNSS. The proposed technique, named as Tracking Error Compensator (TrEC), utilizes the multipath invariant properties of the received correlation function in order to provide significant performance benefits over nEML for narrow-band GPS receivers [95], [96].

One of the most promising advanced multipath mitigation techniques is the Multipath Estimating Delay Lock Loop (MEDLL) [86], [87], [123] implemented by NovAtel for GPS receivers. MEDLL is considered as a significant evolutionary step in the receiver-based attempt to mitigate multipath. It uses many correlators in order to determine accurately the shape of the multipath corrupted correlation function. According to [123], MEDLL provides superior long delay multipath mitigation performance compared to nEML at the cost of multi-correlator based tracking structure.

A new technique to mitigate multipath by means of correlator reference waveform was proposed in [129]. This technique, referred to as Second Derivative correlator, generates a signal correlation function which has a much narrower width than a standard correlation function, and is therefore capable of mitigating multipath errors over a much wider range of secondary path delays. The narrowing of the correlation function is accomplished by using a specially designed code reference waveform (i.e. the negative of the second order derivative of correlation function) instead of the ideal code waveform used in almost all existing receivers. However, this new technique reduces the multipath errors at the expense of a moderate decrease in the effective Signal-to-Noise Ratio (SNR) due to the effect of narrowing the correlation function.
1.2 Scope of the Thesis

A similar strategy, named as Slope Differential (SD), is based on the second order derivative of the correlation function [69]. It is shown in [69] that this technique has better multipath performance than nEML and Strobe Correlator. However, the performance measure was solely based on the theoretical MEE curves, thus its potential benefit in more realistic multipath environment is still an open issue.

A completely different approach to mitigate multipath error is used in NovAtel’s recently developed Vision Correlator [35]. The Vision Correlator (VC) is based on the concept of Multipath Mitigation Technique (MMT) developed in [130]. It can provide a significant improvement in detecting and removing multipath signals as compared to other standard multipath resistant code tracking algorithms (for example, PAC of NovAtel). However, VC has the shortcoming that it requires a reference function shape to be used to fit the incoming data with the direct path and the secondary path reference signals. The reference function generation has to be accomplished a-priori, and it must incorporate the issues related to Radio Frequency (RF) distortions introduced by the front-end.

Several advanced multipath mitigation techniques were also proposed in [9], [39], [40], [77]. These techniques, in general, offer better tracking performance than the traditional DLL at a cost of increased complexity. However, the performance of these techniques have not yet been evaluated in more realistic multipath channel model with real GNSS signals.

To summarize the discussion, many correlation-based multipath mitigation techniques exist; but even the most promising ones (for example, nEML, HRC, MEDLL, etc.) are not good enough for a closely spaced multipath environment. Hence, this is the key motivation in this thesis to come up with new innovative multipath mitigation techniques, which will improve the positioning accuracy in multipath environments (for example, in urban canyons).

1.2 Scope of the Thesis

The main scope of this thesis is the analysis of multipath mitigation techniques for satellite-based positioning applications. The Author analyzes a wide range of correlation-based multipath mitigation techniques in static and fading multipath channels for a group of GNSS signals, more specifically, the interoperable civilian signals from two different navigation systems, i.e., Galileo E1 Open Service signal and the modernized GPS L1C civil signal, along with the existing GPS L1 C/A signal used in almost all the receivers available to-
day. During the process of analyzing these algorithms, the Author proposes several novel multipath mitigation techniques applicable for a wide range of applications starting from simple low-cost mass-market receivers to relatively complex expensive high-end receivers. The performance of the proposed as well as the state-of-the-art techniques has been tested and evaluated through extensive simulations in terms of different performance criteria, such as Multipath Error Envelope and Root-Mean-Square Error (RMSE) in various multipath scenarios. A general comparison of all these techniques is also presented considering the issues related to multipath performance and required implementation complexity.

1.3 Thesis Contributions

The main contributions of the thesis can be summarized as follows:

- Analyzing state-of-the-art multipath mitigation techniques via theoretical, simulated and Simulink-based models [P1]-[P8],
- Designing and implementing several novel correlation-based multipath mitigation techniques (as mentioned below) and comparing their performance with state-of-the-art techniques,
- Proposing a new approach for multipath mitigation, namely Peak Tracking (PT), a weight-based combination of both feedback and feedforward structures, offering a better tracking performance than the conventional techniques at a cost of increased complexity [P1], [P2],
- Implementing a non-coherent version of Multipath Estimating Delay Lock Loop that incorporates a phase search unit based on the statistical distribution of multipath phases [P2],
- Proposing a simple new multipath mitigation technique, Slope-Based Multipath Estimator (SBME), capable of mitigating the short-delay multipath quite well as compared to other state-of-the-art techniques, such as nEML and HRC [P5],
- Proposing a novel maximum likelihood based advanced multipath mitigation technique, Reduced Search Space Maximum Likelihood (RSSML) delay estimator, capable of mitigating the multipath effects in harsh multipath environment (i.e., more than two path scenarios) at a cost of a higher number of correlators [P4], [P7],
1.4 Thesis Outline

- Presenting a C/N\textsubscript{0}-based two-stage delay tracking structure, capable of reducing the instability involved in Multiple Gate Delay (MGD) correlators while preserving the benefits of multipath mitigation [P6],

- Designing and implementing several Simulink blocks for the Galileo E1 open source Simulink receiver [P6],

- Developing semi-analytical models for MBOC modulation and its variants selected for Galileo E1 Open Service (OS) and modernized GPS L1C signals [P8].

1.4 Thesis Outline

The core of this thesis is in the area of multipath mitigation for satellite-based positioning applications. It is composed of an introductory part with nine chapters and a compendium of eight publications referred in text as [P1], [P2], ..., [P8]. These include five articles published in international conferences, two articles in international journals and one article in a renowned GNSS-related magazine. The new techniques and the main results of the thesis have been originally presented in [P1]-[P8], and they are briefly referred in the text. In this thesis, the presented multipath mitigation techniques are analyzed mainly for Galileo E1 signal and the modernized GPS L1C signal, while keeping the legacy GPS L1 C/A signal as benchmark. However, saying so, this does not limit the applicability of these multipath mitigation techniques to other GNSS signals, for example, to Galileo E5 or GPS L5 signal; but proper measures have to be taken to adapt these techniques in the context of any new GNSS signals not reported in this thesis. The remaining of this thesis is organized as follows.

Chapter 2 starts with a description on the basic operating principles of satellite-based positioning technology, provides a brief overview of the most promising Global Navigation Satellite Systems, and finally, discusses about the potential GNSS application areas.

The signal and channel model were described in Chapter 3. First, the BOC modulation family, recently selected for future GNSS signals, is discussed. After that, an overview of a simplified baseband signal and channel model for GNSS signals is presented in the context of this thesis.

The main functionalities of a GNSS receiver are discussed in Chapter 4 with particular attention to signal acquisition and tracking. In this chapter, the Author also introduces a multi-correlator based delay tracking structure
required by the proposed advanced multipath mitigation techniques for estimating the channel properties to take decision on the correct Line-Of-Sight (LOS) delay.

In Chapter 5, the various error sources for satellite-based positioning technology are briefly described with a major focus on multipath error, as being the most challenging error due to its uncorrelated behavior. The effects of various signal and receiver parameters on signal tracking performance, and the relation between these parameters and the multipath error are also analyzed here to better understand the research problem addressed in this thesis.

After providing a brief overview of some of the most promising state-of-the-art multipath mitigation techniques in Chapter 6, the Author presents the novel multipath mitigation techniques, which are originally proposed by the Author in various publications from [P1]-[P7]. The performance of these multipath mitigation techniques are evaluated in terms of different well-known performance criteria, such as running average error and root-mean-square error in different simulation models as described in Chapter 7.

A short summary of the thesis publications [P1]-[P8] is presented in Chapter 8, where the Author’s contribution to each of the publications is also clarified. The general conclusions of the thesis and the remaining open issues are addressed in Chapter 9. Also, a comparison of the proposed as well as some promising state-of-the-art multipath mitigation techniques is presented in this chapter, considering the issues related to multipath performance and implementation complexity.

Finally, the original results of the thesis, which are summarized in the introductory part, are reported in the publications, attached as appendices to the thesis.
Chapter 2

Global Navigation Satellite Systems

This chapter first provides a brief history of the satellite navigation system, and then, discusses the basic operating principle of satellite-based positioning technology. After that, an overview of current and future GNSSs is presented with a major focus on signal structures and services offered by these navigation systems. Finally, a discussion on the major GNSS application areas is addressed at the end of this chapter.

2.1 Brief History of Satellite Navigation System

The Navy Navigation Satellite System (NAVSAT, better known as TRANSIT) was the first operational GNSS in history, prior to the development of the NAVSTAR (Navigation System by Timing and Ranging) GPS. It was developed by the US Navy in late 1950s, became operational in 1964 and finally accessible to civil users in 1967. TRANSIT was used primarily for the navigation of surface ships and submarines, as well as for hydro-graphic surveying and geodetic position determination. A TRANSIT receiver used the known characteristics of a satellite’s orbit and measurements of the Doppler shift of the satellite’s radio signal to establish an accurate position on Earth. TRANSIT was being operational until the mid 1990s, when the new GPS came into operation. All present and future navigation systems can be considered as successors of TRANSIT.
2.2 Fundamentals of Satellite-based Positioning

Since ancient times, human beings have pondered about how to locate themselves with respect to known references. This human effort of positioning oneself is continuing even today with the advent of satellite navigation that offers an accuracy far beyond the early days when positioning was accomplished by simply observing the sun and the stars. Satellite navigation, being a constantly developing technology, offers anyone with a GNSS receiver capable of picking up signals emitted by a constellation of satellites to instantly determine his or her position in time and space very accurately.

The operating principle of a satellite-based positioning technology is based on the Time-of-Arrival (ToA) measurements of the transmitted signals. The position of the receiver is determined by estimating the propagation delay that the signal takes to arrive at the receiver from the satellite. Since the transmission time can be measured from the received navigation data, the propagation delay can then be calculated as the difference between the transmission time and the ToA. The ToA can be usually obtained via a code synchronization process in a Direct Sequence - Code Division Multiple Access (DS-CDMA) receiver. The measured propagation delay is then multiplied with the speed of light in order to get the distance between the transmitter and the receiver (the electro-magnetic satellite signal traverses at a speed of light, i.e., 299,792,458 m/s). In satellite navigation, this distance is popularly known as pseudorange. The propagation time of the signal has to be measured very accurately, since a small fractional error can lead to a large error in pseudorange. For example, in case of GPS L1 C/A signal, 1 micro-second (i.e., \(10^{-6}\) seconds) timing error can lead to an error of about 293 meters in the pseudorange. Therefore, the timing accuracy has to be in the order of nano-seconds (i.e., \(10^{-9}\) seconds) or even lesser for an acceptable position accuracy (i.e., in the order of few meters or less). The receiver is usually able to estimate its position after successful determination of four or more pseudoranges by using multilateration method. More details on the fundamentals of satellite-based positioning technology can be found, for example, in [60], [92], and [98].

2.3 Overview of GPS

The NAVSTAR GPS is a worldwide continuously available all-weather space-based radio navigation system, which provides three dimensional position, velocity, and time to end-users with appropriate receivers. The system is
2.3 Overview of GPS

implemented and operated by the US Department of Defense (DoD), and consists of three major segments: a space segment (the actual satellites), a control segment (management of satellite operations), and a user segment.

The full operational constellation of GPS was declared in April 1995 with the baseline GPS being specified for 24 satellites. However, the system currently employs more satellites than specified in the nominal constellation, and at the time of writing, the GPS constellation consists of 31 Block II/IIA/IIR/IIR-M/IIF satellites \[125\]. The constellation operates in six Earth-centered orbital planes, 60 degrees apart, nominally inclined at 55 degrees to the equatorial plane. Each orbital plane thus contains four to five satellites orbiting at an altitude of 20200 kilometers (km) from the mean surface of the Earth, with a period of one-half of a sidereal day (i.e., 11 hours and 58 minutes). This ensures that a stationary user on the ground would see the same spatial distribution of the satellites after one sidereal day (i.e., 23 hours and 56 minutes).

GPS signals use a Direct Sequence Spread Spectrum (DSSS) technique, and are based on CDMA principle to distinguish signals coming from different satellites \[92\], \[94\]. The legacy GPS signals are transmitted into two frequency bands: L1 centered at 1575.42 MHz, and L2 centered at 1227.60 MHz. The carrier signals are modulated by the Pseudorandom Noise (PRN) codes using BPSK modulation. Each satellite uses the same carrier frequencies, but the signals are separated with specific PRN codes in order to avoid interference and to be able to detect the desired signal. In legacy GPS, there are two basic code types: a short C/A code with 1 ms period that offers Standard Positioning Service (SPS) at L1 frequency, and a long Precision (P) code (further encrypted with Y code) that offers Precise Positioning Service (PPS) both at L1 and L2 frequencies. The navigation message is superimposed on both the C/A and P codes, which contains satellite ephemeris data, atmospheric propagation correction data, and satellite clock bias.

The modernization of GPS became obvious when the design of a new signal with better performance and flexibility was started. The modernization of GPS satellites has already been initiated with the launch of the first operational Block IIR-M satellite in December 2005, followed by the launch of the first operational Block IIF satellite in May 2010 \[43\]. The GPS modernization plans are related to the new generations of navigation satellites, which are briefly discussed in the following.

Block IIR-M satellites offer a second civil signal on L2 (denoted as L2C). The L2C signal uses the same BPSK modulation as the L1 C/A signal. For
military purposes, a new BOC(10,5) modulated M-Code will be added on L1 and L2 frequency bands, where the M-code will be spectrally separated from the civil signals by being centered 6 to 9 MHz above and below the L1 and L2 centers. In addition, future GPS satellites will be designed to be capable of broadcasting regionally the M-codes at a 20 dB higher power level.

Block IIF satellites provide a third civil signal on L5, where QPSK (Quadrature Phase Shift Keying) modulated L5 carrier is centered at 1176.45 MHz. The new L5 signal has a code rate 10 times higher than the L1 C/A signal. This will eventually improve code measurement accuracy, reduce code noise, reduce cross-correlation concerns, and provide improved multipath mitigation. To take full advantage of L5, one of the two quadrature signals to be transmitted without data modulation. The data free signal provides advantages for accurate phase tracking and more precise carrier phase measurements, of special interest to the survey and scientific communities. As like L5, the new L2C signal is also time multiplexed to provide a data free signal along with a signal with data. The addition of new civil signals offers capabilities like ionospheric correction, improved signal robustness, increased interference rejection and improved dynamic precision through the use of techniques for resolving the ambiguities associated with precision carrier phase measurements [79].

Block III satellites will offer increased signal power at the Earth’s surface, improved accuracy, greater availability and controlled integrity. These satellites will transmit a fourth civil signal on L1 (denoted as L1C), which is Multiplexed BOC (MBOC) modulated to ensure compatibility and interoperability with the Galileo E1 Open Service (OS) signal. The Multiplexed BOC (MBOC) modulation concept is explained in Section 3.1. The fourth civil signal will be fully available by approximately 2020.

2.4 Overview of Galileo

Galileo, the permanent European footprint in time and space, will provide a highly accurate, guaranteed global positioning service under civilian control. It will be interoperable with two other global satellite navigation systems, the US GPS and the Russian GLONASS. Therefore, a user will be able to calculate his or her position using the same receiver for any satellite in any combination.

After the completion of the definition phase of Galileo, the development and In-Orbit Validation (IOV) phase was initiated in late 2003. During this phase, two experimental Galileo satellites were launched to secure the Galileo
frequencies filing, characterize the Medium Earth orbit (MEO) environment and test in orbit the most critical satellite technologies. The first experimental satellite, GIOVE-A, was launched on 28th December 2005, and placed in the first orbital plane from where it is being used to test the equipment on board and the functioning of ground station equipment. The second experimental satellite, GIOVE-B, was launched on 27th April 2008, and it continued the testing begun by its older sister craft with the addition of a passive hydrogen maser and with a mechanical design more representative of the operational satellites. A reduced constellation of four satellites, the basic minimum for satellite navigation in principle, will be launched in 2011 to validate the navigation concept with both space and ground segments [28]. After the completion of IOV phase, additional satellites will be launched to reach the Initial Operational Capability (IOC). At this IOC stage, the OS and Search And Rescue and Public Regulated Service (SARPRS) will be available with initial performances. The full Galileo constellation is scheduled to be available approximately by 2014 [27].

The fully deployed Galileo system consists of 30 satellites (27 operational + 3 active spares) divided into three circular orbits inclined at 56 degrees at an altitude of 23222 km to cover the Earth’s entire surface. Ten satellites will be spread evenly around each plane, with each taking about 14 hours to orbit the Earth. Each plane will also have one active spare satellite, which is able to cover for any failed satellite in that plane. Galileo is designed to satisfy requirements that can be divided into five distinct service groups [24], [29], [50], [89], as mentioned below.

Galileo Open Service is designed for mass-market applications, to deliver signals for timing and positioning free of charge. The OS will be available to any user equipped with a receiver capable of navigating with Galileo signals. It is anticipated that most of the applications in future will use a combination of Galileo and GPS signals, which will improve performance in severe environments, such as in urban areas.

The Safety-of-Life service will be used mainly for safety critical applications like aviation and other transport means on land and water. The SoL service will provide the same level of accuracy in position and timing as the OS with the main difference being the high integrity level obtained by means of an integrity data message. The SoL service will automatically inform users within a 6 second time-to-alarm of any signal failure possibly affecting its specified performance. The SoL service will be certified and it will be accessed through a dual-frequency receiver (e.g., frequency bands L1 and E5a).
The Commercial Service (CS) is aimed at market applications which require higher performance than offered by the OS. It will provide value-added services on payment of a fee with the addition of two signals to the OS signal. This pair of additional signals is protected at receiver level through commercial encryption using access-protection keys, which will be managed by the service providers and a future Galileo operating company. These value-added services include, e.g., high data-rate broadcasting, precise timing services, service guarantees, the provision of ionosphere delay models, and local differential correction signals for extreme-precision position determination.

The fourth service that Galileo will offer is the Public Regulated Service (PRS) that is expected to be used by groups such as the police, coastguard and customs. Civilian institutions will control access to the encrypted public regulated service, which is mandated to be operational due to the robustness of its signal at all times and in all circumstances, especially during periods of crisis, when some other services may be intentionally jammed.

The fifth service, Search And Rescue (SAR), will allow important improvements to the existing humanitarian search and rescue services. These will include near real-time reception of distress messages from anywhere on the Earth and transferring it to a rescue coordination center. A return signal will then be sent back to the users advising that help is on the way.

2.5 Overview of GLONASS

GLONASS satellite navigation system, the Russian counterpart to GPS became fully operational in 1995. The operational space segment of GLONASS consists of 21 satellites with 3 on-orbit spares. Nominally, these satellites are in three orbital planes separated by 120 degrees, and equally spaced within each plane at a nominal inclination of 64.8 degrees. The orbits are roughly circular and the satellites orbit the Earth at an altitude of 19100 km, which yields an orbital period of approximately 11 hours and 15 minutes. Due to lack of funding and relatively short life span of only 3 to 4.5 years, the GLONASS constellation could not be maintained and the number of operational satellites decreased to only 7 in 2001. Forced by a GLONASS modernization program, the number of operational satellites has now been increased to 20 at the end of 2010. With the modernization effort, the GLONASS performance is expected to be comparable to that of GPS and Galileo. In order to achieve this goal, the GLONASS modernization plan includes modernization of satellites transmitting new navigation signals, extension of the existing ground support
2.6 Other Satellite Navigation Systems

In recent years, there have been many activities all around the globe that intend to provide either full navigation capabilities or local or regional augmentations to the global positioning systems. This is the case for Beidou (i.e., the Chinese satellite-based regional navigation system), which has now been expanded to a global navigation system in the form of Compass by the end of this decade. There are also GAGAN (GPS Aided Geo Augmented Navigation) from India and QZSS (Quasi-Zenith Satellite System) from Japan, which also provide regional augmentations to India and Japan, respectively.

2.7 GNSS Applications

As of now, GPS is the only prevailing GNSS primarily used by both military and civilian users. However, the future users will have several additional GNSS systems at their disposal as new systems (for example, Galileo) come online. Many of these GNSS signals, being free and globally available, will be used in advanced applications that were initially pioneered by GPS. Some of the major GNSS applications are briefly summarized below:

- **Personal navigation:** This consists of applications to aid people to navigate. Perhaps the most known form of personal navigation are the car navigators navigating the user to a specified location. GNSS is currently making its way into cell phones (for example, Nokia E97, N97, N8 [88]; iPhone 4 [1], etc.) and PDAs, expanding satellite navigation to a large new group of users.

- **Aviation applications:** The aviation users require a very high level of
performance in terms of accuracy and robustness for en route navigation as well as for precision approach and landing.

- **Marine applications:** GNSS emerged as a blessing to marine users due to the clear views of the sky and modest accuracy requirements of most maritime applications. Today, GNSS receivers have become standard equipment for all types of boats, and they perform a very precious service to the global maritime community.

- **Space applications:** GNSS receivers, GPS in particular, have proven to be a very valuable tool for Low Earth Orbit (LEO) satellites. Their use is currently expanding into space vehicles operating at higher altitudes. GPS has the full potential to be implemented as an attitude determination sensor.

- **Geodesy and surveying:** The users of these applications are perhaps utilizing the best benefits that have resulted from the public availability of GPS signals. Geodesy applications require precision positioning information at the centimeter or millimeter level and include applications such as the monitoring of the movements of the Earth’s crystal plates or ice shelves, often involving extensive post-processing [42]. Similarly, surveying with GNSS receivers has also become widespread with relatively relaxed accuracy requirements.

- **Forestry, agriculture and natural resource exploration:** These diverse applications include forest management, geological monitoring, mining, and oil exploration. These applications often combine GNSS field measurements with geographic information system tools to produce accurate regional maps for resource monitoring and management.

In addition to all the above listed applications, there are also many other application areas which are becoming more and more attractive for GNSS users. One such application area is object and person tracking. Many personal tracking applications are developed for sports, both to enhance training [25], [120], and spectator experience in sports like car racing, cricket, triathlon, cycling, etc. Asset tracking is another example of this, where trains, trucks and valuable containers can be tracked for better management and increased security [99].
Chapter 3

Signal and Channel Model

In this chapter, the Binary Offset Carrier (BOC) modulation used in modernized GPS and Galileo systems is explained. Next, an overview of a simplified baseband signal and channel model for GNSS signals is presented in the context of this thesis. It is worth to mention here that the work carried out in this thesis mainly focuses on three different signal modulations, namely Binary Phase Shift Keying (BPSK) modulation used in legacy GPS L1 C/A signal, Multiplexed Binary Offset Carrier (MBOC) modulation used in Galileo E1 signal and modernized GPS L1C signal, and Sine BOC (SinBOC) modulation that can be used as an alternative to demodulated MBOC signal at the receiver side.

3.1 Binary Offset Carrier Modulation

The BOC modulation was first introduced by Betz for the modernized GPS system [5], [8]. Since then, several variants have been developed including Sine BOC and Cosine BOC (CosBOC) [6], [8], Alternative BOC (AltBOC) [50], [51], Complex Double BOC (CDBOC) [76] and Multiplexed BOC (MBOC) modulations [49], [106]. According to the July 2007 agreement between the EU and the US [113], there will be a common GPS-Galileo signal, known as MBOC, for civilian use in order to ensure the compatibility and interoperability at the user level. The second Galileo satellite, GIOVE-B already started transmitting the Galileo E1 signal with the MBOC modulation, that will be interoperable with the L1C signal to be used in future Block III GPS satellites. In this thesis, the MBOC modulation and its variants are considered as they are specified to be used for civilian users in both GPS and Galileo systems. In
addition, BPSK modulation used for GPS L1 C/A signal and SinBOC(1,1) modulation used to form the MBOC signal, are also considered in this thesis as benchmark modulations.

A BOC modulated signal can be obtained through the product of a Non-Return to Zero (NRZ) spreading code with a synchronized square wave sub-carrier. The square wave subcarrier can be either sine or cosine phased, and they are referred to as SinBOC and CosBOC, respectively [51]. The typical notation used for a BOC modulated signal is BOC($f_{sc}$, $f_{chip}$), where $f_{sc}$ is the subcarrier frequency in MHz and $f_{chip}$ is the code chip rate in MHz [8]. Alternatively, BOC($p$, $q$) notation is also used, where $p$ and $q$ are two indices computed from $f_{sc}$ and $f_{chip}$, respectively, with respect to the reference frequency $f_{ref} = 1.023$ MHz, $p = \frac{f_{sc}}{f_{ref}}$ and $q = \frac{f_{chip}}{f_{ref}}$. The ratio $N_B = 2 \frac{p}{q} = 2 \frac{f_{sc}}{f_{chip}}$ denotes the BOC modulation order and is a positive integer [75]. For example, $N_B = 2$ represents, e.g., BOC(1,1) and BOC(2,2) modulations, whereas $N_B = 12$ represents, e.g., BOC(15,2.5) or BOC(6,1) modulations. A special case of BOC modulation is the BPSK modulation with $N_B = 1$ [75].

According to the definition in [8], the SinBOC subcarrier can be defined as:

$$s_{\text{SinBOC}}(t) = \text{sign} \left( \sin \left( \frac{N_B \pi t}{T_c} \right) \right), \quad 0 \leq t < T_c$$  \hfill (3.1)

where $\text{sign}(\cdot)$ is the signum operator and $T_c$ is the chip period ($T_c = 1/f_{chip}$). The PRN code sequence can be defined as:

$$x_{PRN,n}(t) = \sum_{k=1}^{S_F} c_{k,n} P_{TB}(t - kT_c - nT_c S_F),$$  \hfill (3.2)

where $k$ is the index, $n$ is the data symbol index, $c_{k,n}$ is the $k$-th chip corresponding to the $n$-th symbol, $S_F$ is the spreading factor, and $P_{TB}(\cdot)$ is the rectangular pulse shape with unit amplitude. After spreading, the data sequence can be expressed as:

$$x_{\text{data}}(t) = \sum_{n=-\infty}^{\infty} \sqrt{E_b} b_n x_{PRN,n}(t),$$  \hfill (3.3)

where $b_n$ is the data bit, and $E_b$ is the bit energy. Now, the data sequence after spreading and BOC modulation can be written as:

$$x_{\text{SinBOC}}(t) = x_{\text{data}}(t) \otimes s_{\text{SinBOC}}(t),$$  \hfill (3.4)

where $\otimes$ represents the convolution operation.
The Composite BOC (CBOC) modulation, a variant of MBOC used in Galileo E1 signal, can be written as [75]:

\[ s_{CBOC}(t) = w_1 s_{\text{SinBOC}(1,1),\text{held}}(t) \pm w_2 s_{\text{SinBOC}(6,1)}(t) \]

\[ = w_1 \sum_{i=0}^{N_{B_1}-1} \sum_{k=0}^{N_{B_2}-1} (-1)^i c(t - i \frac{T_c}{N_{B_1}} - k \frac{T_c}{N_{B_2}}) \]

\[ \pm w_2 \sum_{i=0}^{N_{B_2}-1} (-1)^i c(t - i \frac{T_c}{N_{B_2}}) \]  

(3.5)

In the above, when the two right-hand terms are added, additive CBOC or CBOC(’+’) is formed, and when the two terms are subtracted, we have the inverse CBOC or CBOC(’-’) implementation. Alternatively, CBOC(’+/-’) implementation can be used, when odd chips are CBOC(’+’) modulated and even chips are CBOC(’-’) modulated [49]. In Eqn. 3.5, \( N_{B_1} = 2 \) is the BOC modulation order for SinBOC(1,1) signal, \( N_{B_2} = 12 \) is the BOC modulation order for SinBOC(6,1) signal, the term \( s_{\text{SinBOC}(1,1),\text{held}} \) represents that SinBOC(1,1) signal is passed through a hold clock in order to match the higher rate of SinBOC(6,1); and \( w_1 \) and \( w_2 \) are amplitude weighting factors such that \( w_1 = \sqrt{\frac{10}{11}} = 0.9535 \) and \( w_2 = \sqrt{\frac{1}{11}} = 0.3015 \), and \( c(t) \) is the pseudorandom code. In Eqn. 3.5, the first term comes from the SinBOC(1,1) modulated code (held at rate \( 12/T_c \) in order to match the rate of the second term), and the second term comes from a SinBOC(6,1) modulated code.

Now, in case of Time Multiplexed BOC (TMBOC) modulation, a variant of MBOC that will be used in modernized GPS L1C signal, the whole signal is divided into blocks of \( Q \) code symbols and \( P < Q \) of \( Q \) code symbols are SinBOC(1,1) modulated, while \( Q - P \) code symbols are SinBOC(6,1) modulated. Using similar derivations as in [75], we can obtain the formula for TMBOC waveform. An equivalent unified model of CBOC and TMBOC modulations was derived in [78] using the facts that \( P, Q << \infty \) and that, since \( w_1, w_2 \) are amplitude coefficients and \( P, Q - P \) define the power division between SinBOC(1,1) and SinBOC(6,1), we may set the following relationship between \( w_1, w_2 \) and \( P, Q \): \( w_1 = \sqrt{\frac{P}{Q}} \) and \( w_2 = \sqrt{\frac{Q-P}{P}} \). Therefore, in accordance with [78], the unified model can be written as:

\[ s_{MBOC}(t) = w_1 c_3(t) \otimes s_1(t) \otimes p_{TB}(t) + w_2 c_3(t) \otimes s_2(t) \otimes p_{TB}(t) \]  

(3.6)
where \( \delta(\cdot) \) is the Dirac pulse, \( \otimes \) is the convolution operation, \( p_{T_2}(\cdot) \) is a rectangular pulse of support \( T_c/N_{B_2} \) with unit amplitude and \( c_\delta(t) \) is the code signal without pulse shaping:

\[
c_\delta(t) = \sqrt{E_b} \sum_{n=-\infty}^{\infty} b_n \sum_{m=1}^{S_F} c_{m,n}(t - nT_cS_F - mT_c),
\]

(3.7)

and \( s_1(t) \), \( s_2(t) \) are SinBOC-modulated parts (with associated hold block when needed), given by:

\[
s_1(t) = \sum_{i=0}^{N_{B_1}-1} \sum_{k=0}^{N_{B_2}-1} (-1)^i \delta(t - i T_c N_{B_1} - k T_c N_{B_2}),
\]

(3.8)

and, respectively:

\[
s_2(t) = \sum_{i=0}^{N_{B_2}-1} (-1)^i \delta(t - i T_c N_{B_2}).
\]

(3.9)

### 3.2 Channel Model

Typical GNSS signals, such as those used in GPS or Galileo employ DS-CDMA technique, where a Pseudo-Random Noise (PRN) code from a specific satellite is spreading the navigation data over \( S_F \) chips (or over a code epoch length) \([30], [60]\). In what follows, a baseband model is adopted for clarity reason. The estimation of code delay in today’s receivers is typically done in digital domain using the baseband correlation samples. In the following, the time notation \( t \) denotes the discrete time instant. The signal \( x(t) \) transmitted from one satellite with a specific PRN code can be written as:

\[
x(t) = \sqrt{E_b} p_{\text{MOD}}(t) \otimes c(t),
\]

(3.10)

where \( E_b \) is the bit energy, \( p_{\text{MOD}}(t) \) is the modulation waveform of type MOD (i.e., BPSK for GPS L1 C/A code or CBOC(-) for Galileo E1C signals), and \( c(t) \) is the navigation data after spreading as written below (spreading is done with a PRN code of chip interval \( T_c \) and spreading factor \( S_F \)):

\[
c(t) = \sum_{n=-\infty}^{\infty} b_n \sum_{k=1}^{S_F} c_{k,n}(t - nT_cS_F - kT_c).
\]

(3.11)
Above $\delta(\cdot)$ is the Dirac unit pulse, $b_n$ is the $n$-th data bit (for pilot channels, $b_n = 1, \forall n$) and $c_{k,n}$ is the $k$-th chip ($\pm 1$ valued) corresponding to the $n$-th spread bit.

The modulation waveform for BPSK or BOC can be written as [75]:

$$p_{MOD}(t) = p_T(t) \otimes \sum_{i=0}^{N_B-1} \delta(t - iT_B), \quad (3.12)$$

where $N_B$ is BOC modulation order, $T_B = \frac{T_c}{N_B}$ is the BOC interval, and $p_T(t)$ is the pulse shaping filter (for example, for unlimited bandwidth case, $p_T(t)$ is a rectangular pulse of width $T_B$ and unit amplitude).

The signal $x(t)$ is typically transmitted over a multipath static or fading channel, where all interference sources except the multipath are lumped into a single additive Gaussian noise term $\eta(t)$:

$$r(t) = \sum_{l=1}^{L} \alpha_l x(t - \tau_l)e^{j(2\pi f_D t + \theta_l)} + \eta(t), \quad (3.13)$$

where $r(t)$ is the received signal, $L$ is the number of channel paths, $\alpha_l$ is the amplitude of the $l$-th path, $\theta_l$ is the phase of the $l$-th path, $\tau_l$ is the channel delay introduced by the $l$-th path (typically assumed to be slowly varying or constant within the observation interval), $f_D$ is the Doppler shift introduced by the channel, and $\eta(t)$ is a wideband additive noise, incorporating all sources of interferences over the channel. Assuming that the signal is sampled at $N_s$ samples per chip (for BPSK) or per BOC interval (for BOC modulation), then the power spectral density of $\eta(\cdot)$ can be written as $N_0/(N_sN_B S_F)$, where $N_0$ is the noise power in 1 kiloHertz (kHz) bandwidth (i.e., bandwidth corresponding to one code epoch). Generally, the SNR for any GNSS signal is expressed with respect to the code epoch bandwidth $B_W$, under the name of Carrier-to-Noise density ratio ($C/N_0$). The relationship between $C/N_0$ and bit-energy-to-noise ratio (in dB) can be written as [15]:

$$C/N_0[\text{dB-Hz}] = \frac{E_b}{N_0} + 10 \log_{10}(B_W). \quad (3.14)$$

The delay tracking is typically based on the code epoch-by-epoch correlation $\mathcal{R}(\cdot)$ between the incoming signal and the reference PRN code $x_{\text{ref}}(\cdot)$ with a certain candidate Doppler frequency $\hat{f}_D$ and delay $\hat{\tau}$:

$$\mathcal{R}(\hat{\tau}, \hat{f}_D, m) = \mathbf{E} \left( \frac{1}{T_{\text{sym}}} \int_{mT_{\text{sym}}}^{(m+1)T_{\text{sym}}} r(t)x_{\text{ref}}(\hat{\tau}, \hat{f}_D)dt \right), \quad (3.15)$$
where \( m \) is the code epoch index, \( T_{\text{sym}} \) is the symbol period (i.e., \( T_{\text{sym}} = S_F T_c \)), and \( \mathbb{E}(\cdot) \) is the expectation operator with respect to the random variables (e.g., PRN code, channel effects, etc.), and

\[
x_{\text{ref}}(\hat{\tau}, \hat{f}_D) = \text{pMod}(t) \otimes \sum_{n=-\infty}^{\infty} \sum_{k=1}^{S_F} \hat{b}_n c_{k,n} \delta(t - nT_{\text{sym}} - kT_c) \otimes p_{TB}(t)e^{-j2\pi\hat{f}_Dt}, \quad (3.16)
\]

where \( \hat{b}_n \) is the estimated data bit. For Galileo signals, a separate pilot channel is transmitted \([30]\). In what follows, it is assumed that data bits are perfectly estimated \((\hat{b}_n = b_n)\), and removed before the correlation process. In a practical receiver, in order to cope with noise, coherent and non-coherent integration can be used. The average coherent correlation function \( \bar{R}_c(\hat{\tau}, \hat{f}_D) \) can be written as:

\[
\bar{R}_c(\hat{\tau}, \hat{f}_D) = \frac{1}{N_c} \sum_{m=1}^{N_c} R(\hat{\tau}, \hat{f}_D, m), \quad (3.17)
\]

where \( N_c \) is the coherent integration time expressed in code epochs or milliseconds for GPS or Galileo signal, and the non-coherently averaged correlation function \( \bar{R}_{nc}(\hat{\tau}, \hat{f}_D) \) can be written as:

\[
\bar{R}_{nc}(\hat{\tau}, \hat{f}_D) = \frac{1}{N_{nc}} \sum_{N_{nc}} \frac{1}{N_c} \sum_{m=1}^{N_c} R(\hat{\tau}, \hat{f}_D, m) \bigg|_{p_{nc}}, \quad (3.18)
\]

where \( N_{nc} \) is the non-coherent integration time expressed in blocks of length \( N_c \) milliseconds (for clarity reason, we avoid using the block indexes for the non-coherent summations), and \( p_{nc} \) is the power index used for non-coherent summation. The most encountered variants for \( p_{nc} \) are: \( p_{nc} = 1 \) (with \( p_{nc} = 1 \), Eqn. 3.18 provides the sum of absolute correlation values), and \( p_{nc} = 2 \) (i.e., with \( p_{nc} = 2 \), Eqn. 3.18 provides the sum of squared-absolute correlation values). In all the simulations reported in \([P1]-[P8]\), the later option (i.e., \( p_{nc} = 2 \)) is used, if not mentioned otherwise.
Chapter 4

Functional Description of a GNSS Receiver

A typical GNSS receiver is responsible for several different functions, starting from the successful reception of GNSS signals to the computation of the user’s position. In this chapter, the main functionalities of a GNSS receiver are discussed with a major focus on signal acquisition and tracking.

4.1 GNSS Signal Reception

The major challenge in GNSS signal reception is that the satellite signals travel more than 20000 km through space, and when arrive at the surface of the Earth, they are totally buried under noise. Therefore, the most essential functionality of a conventional GNSS receiver is to be able to estimate the code delay and the carrier frequency of a signal buried in noise via a despreading operation. A conventional GNSS receiver, after some necessary RF front-end processing, does the despreading operation in two stages: i. a coarse acquisition stage, followed by, ii. a fine tracking stage. The block diagram of a conventional GNSS receiver is shown in Fig. 4.1. As shown in Fig. 4.1, a GNSS antenna, the first interfacing component in a receiver setup, is usually tuned to receive a few MegaHertz (MHz) bandwidth around the center frequency of the band [60]. In general, all GPS and Galileo antennas are optimized for Right Hand Circular Polarized (RHCP) radio waves and their radiation (i.e., reception) pattern is hemispherical. Upon receiving the satellite signal, antenna passes it to a RF chain, where the main function is to perform signal conditioning in such a way that it can be later processed in
the receiver. Signal conditioning in the RF front-end includes several signal processing operations like amplification, filtering, down-conversion, mixing, Automatic Gain Control (AGC), Analog-to-Digital Conversion (ADC) and so on. The functional description of each of this signal processing operation is well-documented in literature, for example, in [15], [60], and [92]; and hence, it is not elaborated herein for the sake of clarity and compactness.

![Figure 4.1: A conventional GNSS receiver block diagram.](image)

### 4.2 Signal Acquisition

The purpose of signal acquisition is to determine the visible satellites and to achieve the coarse estimates of the carrier frequency and of the code phase of the satellite signals. Like any other CDMA-based receivers, a GNSS receiver also achieves this acquisition operation in two stages, namely, the search stage and the detection stage, which are briefly summarized below.

#### 4.2.1 Signal Search Stage

GNSS signal acquisition is a three-dimensional search process, which determines the identity of the received signal (i.e., Satellite Vehicle (SV) number), code delay and Doppler shift. The search process requires that, for a certain SV, both the replica code and the carrier are aligned with the received signal. The correct alignment is identified by measurement of the output power of the correlators. In other words, when both the code and carrier Doppler match the incident signal, the signal is despread and a carrier signal is recovered. The result of the code and Doppler search is an estimate of the code offset typically...
within 1/2 chip and the Doppler to within half the Doppler search bin size. In case of a Galileo E1 receiver, acquisition can be performed using a carrier signal and a E1B or a E1C or a coherently or a non-coherently combined E1B and E1C code replica [11], [110].

Search Space

The search space must cover the full range of uncertainty in the code delay and Doppler frequency domain. The code search space is typically equal to the length of the spreading code, which is 1023 chips for GPS L1 C/A signal, and 4092 chips for Galileo E1 OS signal. The resolution of the code search is typically set to a fraction of a chip (i.e., ≤ 1/2 chip for GPS L1 C/A signal), and it can often be specified by the sampling frequency of the received signal. The Doppler search space is governed by the receiver and GNSS satellite dynamics and the stability of the receiver oscillator. For a terrestrial user system, this is typically in the range of 5 to 10 kHz [64]. The frequency resolution is determined by the coherent integration time (or dwell time). According to [60], the rule of thumb is: $\Delta f = \frac{2}{3 T_{coh}}$, where $\Delta f$ is the frequency bin width in hertz and $T_{coh}$ is the predetection integration time in seconds.

Search Window

Each tentative code delay is called a code bin (or a time bin), and respectively, each tentative frequency shift is denoted as a Doppler bin (or a frequency bin). A single code bin along with a frequency bin compose a search bin or a test cell [60]. The whole code-frequency search space can be divided into several search windows depending on the type of search techniques chosen for implementation.

Search Techniques

The proposed PRN codes for Galileo have higher lengths (e.g., 4092 chips for E1 OS signal and 10230 chips for E5 signals [30]) than those used by legacy GPS C/A signal. The use of longer codes leads to an increased search space, which eventually makes the search process more time consuming. Several search techniques have been proposed in literature (for example, in [60], [83], and [92]) in order to obtain a faster and a more efficient signal acquisition. These search techniques can be classified in three major categories as serial search, parallel search and combined serial/parallel search (or, hybrid search).
In a serial search technique, the search window consists of only one bin and the delay shift is changed by steps of one time-bin length. Hence, all time bins are examined one by one in a serial manner and only one search detector is needed for the acquisition. Since the time-frequency bins are tested one at a time in the serial search technique, the mean acquisition time is too slow to meet today’s user requirements, specially in the presence of longer GNSS codes. The availability of computational resources in today’s receivers and the adoption of longer codes for new GNSS signals have made the serial search technique almost obsolete in most cases [53].

A parallel search technique employs a bank of matched filters, where each of them matched to a different waveform pattern of PRN code subsequences corresponding to all possible PRN code delays and all possible Doppler bins, and then makes a decision based on all the filter outputs [116], [133]. In a fully parallel search technique, the acquisition scheme simultaneously tests all possible code delays and all possible Doppler bins, resulting in a significant reduction in mean acquisition time. Undoubtedly, the smaller mean acquisition time comes at a cost of higher implementation complexity, since a large number of correlators is required.

A hybrid search technique is a nice trade-off between the serial and parallel search techniques, as it attempts to achieve a proper balance between the acquisition speed and the hardware complexity. Many hybrid search acquisition schemes have been studied for decades ([3], [59], [66], [74], [97]), considering the serial- and parallel-search schemes as two extreme cases. With the growth of computing resources and a higher demand set by longer codes, the hybrid search acquisition technique has become a popular choice for present day GNSS receivers.

4.2.2 Signal Detection Stage

A simplified signal acquisition block diagram is shown in Fig. 4.2. Acquisition starts with the correlation between the received signal and the locally generated replica code for a coherent integration period of $N_c$ ms (for example, 1 ms for GPS L1 C/A, or 4 ms for Galileo E1 OS signal), followed by an Integrate and Dump (I&D) process to form the correlation output. After the coherent processing, a non-coherent integration is usually preferred in order to decrease the noise floor and also because of the fact that the coherent integration time $N_c$ might be limited by the channel fading, Doppler and by the instability of the oscillator clocks [101]. The next step in the acquisition process is to
determine whether the satellite signal is present or absent (i.e., if there is a synchronization between the code and the received signal or not). This is the main objective of the detection stage.

Figure 4.2: A simplified signal acquisition block diagram.

At the detection stage, a test statistic is usually computed per each search window based on the current correlation output, and then this test statistic is compared to a certain predetermined threshold \( \gamma \) in order to decide whether the signal is present or absent. The test statistic can be formed, for example, as the value of the global maximum of the correlation output in one search window [22], [111] or as the ratio between the global maximum and the next significant local maximum [58], [90], [91]. If the value of the test statistic is higher than the threshold, the signal is decided to be present and an estimate for the code phase and frequency is achieved. The time to form the test statistic is usually denoted as the dwell time \( \tau_d \). Every time when the test statistic is higher than the threshold, the signal is decided to be present. The probability of a signal being detected correctly is denoted as a detection probability \( P_d \). A false alarm triggers when a delay and/or frequency estimate is wrong but the test statistic is still higher than the threshold, i.e., the signal is declared present in an incorrect window. The probability of a false alarm is denoted as the false alarm probability \( P_{fa} \). It may also happen that the signal is present, but not detected: a miss detection occurs. This may happen if the threshold is set too high or the environment is so noisy that the signal is lost into the background noise. Therefore, the choice of a suitable detection threshold \( \gamma \) has a significant role in the acquisition process. If the threshold is set too low, the probability of detection (i.e., \( P_d \)) naturally increases, but at the same time, the probability of false alarm (i.e., \( P_{fa} \)) increases as well.
Respectively, if the threshold is set too high, the $P_{fa}$ decreases, but also the $P_d$ is low. The selection of a suitable detection threshold can be found, for example, in [61].

### 4.3 Signal Tracking

#### 4.3.1 Background on Tracking Loops

The signal tracking stage refines the estimates of the signal parameters (i.e., code delay and Doppler frequency) obtained at the acquisition stage. In fact, these parameters are not accurate enough to be used for positioning and navigation. Apart from that, the carrier phase information is also ignored at the acquisition stage and more interestingly, all these parameters change over time. The tracking stage gives a fine estimate of the code delay, the Doppler frequency and the carrier phase, and it continuously follows their variations.

The signal tracking stage consists of two inter-connected tracking loops. They are used for tracking the quantities to be estimated: the code delay $\tau$, the Doppler frequency $f_D$, and the carrier phase $\phi$ in the form of Delay Locked Loop (DLL), Frequency Locked Loop (FLL) or Phase Locked Loop (PLL), respectively. In certain applications, FLL and PLL can be used in sequence or in a FLL-assisted-PLL configuration, where the receiver uses the FLL first in order to reduce the frequency uncertainty before switching to PLL [114]. A similar FLL-assisted-PLL configuration was also implemented in the TUT Galileo E1 Simulink receiver, as mentioned in [110].

Both the DLL and FLL-assisted-PLL loops are closely inter-related and they work in an inter-connected way. More specifically, the DLL is used to reproduce a local version of the spreading code that is removed from the incoming signal before entering the PLL (code wipe-off). In this way, a pure carrier is found at the input of the PLL. Similarly, the PLL is used to generate a local version of the signal carrier that is down-converted to baseband. In this way, a noisy baseband binary sequence enters the DLL (carrier wipe-off).

#### 4.3.2 Multi-Correlator based Delay Tracking Structure

In a multi-correlator based delay tracking structure, a bank of correlators is generated, unlike the conventional DLL-based tracking structure, where only few correlators (i.e., in the range of three to seven complex correlators depending on the type of techniques) are used. This large number of correlators are required by the advanced multipath mitigation techniques, as we proposed in
[P2] and [P8], in order to estimate the channel properties and to take a decision on the correct LOS code delay. As shown in Fig. 4.3, after the necessary

Figure 4.3: Block diagram for multi-correlator based DLL implementation.

front-end processing, and after the carrier has been wiped-off, the received post-processed signal is passed through a bank of correlators. The NCO (Numerically Controlled Oscillator) and PRN generator block produces a bank of early and late versions of replica codes based on the delay of the LOS signal \( \tau \), the correlator spacing \( \Delta \), and the number of correlators \( M \). In case of an EML tracking loop, the corresponding early-late spacing is equal to \( 2\Delta \). The received signal is correlated with each replica in the correlator bank, and the output of the correlator bank is a vector of samples in the correlation envelope. Therefore, we obtain the correlation values for the range of \( \pm M\Delta \) chips from the prompt correlator, where \( M \) is the number of correlators and \( \Delta \) is the correlator spacing between successive correlators. The various code tracking techniques (named as Discriminator in Fig. 4.3) utilize the correlation values as input, and generate the estimated LOS delay as output, which is then smoothed by a loop filter. A loop filter is generally used to improve the code delay estimate, reducing the noise present at the output of the discriminator, and to follow the signal dynamics. The order of a loop filter determines the ability of the filter to respond to different types of dynamics, whereas the filter bandwidth ensures that a low bandwidth leads to a good filtering with a high
amount of noise filtered, but also requires that the dynamics of the signals are not too high. The loop bandwidth is usually determined by the coefficients of the filter, and can thus be considered as a design parameter for the filter. In accordance with [60], the code loop filter implemented in [P7], is a 1st order filter, which can be modeled as:

\[ \hat{r}(k + 1) = \hat{r}(k) + \omega_0 d(k), \] (4.1)

where \( \omega_0 \) is calculated based on the loop filter bandwidth, \( B_n \). As shown in [P7], the multi-correlator based tracking structure being used by the advanced multipath mitigation techniques, offers a superior tracking performance to the traditional nEML DLL at the cost of higher number of correlators. Therefore, this structure is only suitable for professional high-end receivers.

4.3.3 \( C/N_0 \) Estimation

\( C/N_0 \) is an important parameter to measure the quality of a received GNSS signal, that can also be used to enhance the performance of delay tracking loops. The ratio between the received power of the signal carrier and the noise power in a 1 Hz bandwidth is represented as \( C/N_0 \) [67]. The total power received by the antenna, as stated by the Friis transmission formula, depends on the satellite transmitted power, the satellite antenna gain, the receiver antenna gain, the free-space losses and the attenuations of the channel. Standard GNSS receivers can handle signals with \( C/N_0 \) in the range of 35–53 dB-Hz, whereas, high-sensitivity receivers can process even weaker signals. The most known \( C/N_0 \) estimation method is based on the ratio of the signal’s wideband power to its narrowband power as mentioned in [92]. In this method, the power of the signal is computed over a wide bandwidth with a relatively short coherent integration time and over a narrow bandwidth with a longer coherent integration time. In [P6], the author computed the wideband power after 4 ms of coherent integration (after each code epoch length), and the narrowband power after 16 ms of coherent integration in order to estimate the \( C/N_0 \) for each particular channel.

4.4 Navigation Solution

The navigation solution evaluates the user’s position and velocity from the pseudoranges and the navigation data. An estimate of the user position and velocity can be attempted in a GNSS receiver, only if the following conditions are fulfilled:
1. A minimum of four satellites are being tracked.

2. The transmission time for each satellite at the measurement data sample is known.

3. Ephemerides for the tracked satellites are available.

After fulfilling the above conditions, the ephemerides are processed in order to obtain satellite locations and a navigation solution can be calculated based on the satellite-to-user distances. The pseudoranges, determined by converting the code delays into distances, provide the position of the user, whereas the velocity is obtained by converting the information on the Doppler frequency. The details on how to implement the navigation algorithms are out of scope of this thesis, and the theory behind it is very well presented, for example, in [12], [42], [60], [83], and [92].
Chapter 5

Multipath and Other GNSS Error Sources

Errors in satellite navigation signals can be classified into three main categories: satellite-based errors, signal propagation errors, and receiver-based errors. Satellite-based errors include satellite clock error and ephemeris error. Signal propagation errors include errors associated with the atmospheric propagation delay due to ionosphere, troposphere, multipath propagation delay and interference. Receiver-based errors include receiver noise and other smaller errors, such as, e.g., inter-channel biases and antenna error. A knowledge on the possible error sources is essential in obtaining a user position solution with the desired performance level. In the following, the various error sources of satellite navigation signals are discussed with a major focus on multipath error, as being the most challenging error due to its uncorrelated behavior.

5.1 Satellite-based Errors

Satellite-based errors consist of the errors in the orbital plane and of the satellite clock parameters broadcast in the navigation message for which the GNSS control segment is responsible. The prediction error of the satellite ephemeris and clock parameters grows with the age of the data, i.e., the time since the last parameters have been uploaded. Therefore, the more frequent the control segment uploads data to the satellites and the more accurate the models used to estimate and predict the ephemeris and clock parameters, the less significant are the satellite-based errors.
5.2 Signal Propagation Errors

The navigation signals are affected by the medium through which they travel from the satellites to the receiver antenna. The signals encounter the ionosphere at a height of about 1000 km, and the electronically neutral gaseous troposphere at a height of around 40 km from the surface of the Earth. Apart from these two propagation error sources, there can be multipath propagation and interference and jamming, which further degrade the navigation performance of the receiver. In the subsequent subsections, a brief discussion on these error sources is provided, except the multipath propagation, which is covered in more detail in Section 5.4.

5.2.1 Ionosphere

The ionosphere is a dispersive medium which extends from about 50 to 1000 km above the Earth and is characterized by an abundance of free electrons and ions. The signal delay due to the ionosphere is directly proportional to the integrated electron density along the signal path, i.e., the Total Electron Content (TEC), and inversely proportional to the squared frequency of the signal \[119\]. TEC is defined as the number of electrons in a tube of 1 m\(^2\) in cross-section extending from the receiver to the satellite \[83\]. Since the ionosphere is a dispersive medium, i.e., the refractive index of the ionosphere is dependent on the frequency of the navigation signal, two-frequency GNSS users can take advantage of this property of the ionosphere to measure and correct for the first-order ionospheric range and range rate effects directly \[93\].

In general, the major effects that the ionosphere can have on the navigation signals include: delay of the code phase, i.e., absolute range error, carrier phase advance, Doppler shift (i.e., range rate error), refraction or bending of the radio wave, distortion of pulse waveforms, and signal amplitude and phase scintillation \[93\]. 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.

Ionospheric correction models should be employed for single-frequency GNSS users. For example, an ionospheric delay compensation model by Klobuchar removes, on an average, about 50 percent of the ionospheric delay at mid latitudes by assuming that the vertical ionospheric delay can be approximated by utilizing the broadcast ionospheric delay coefficients in a model including half a cosine function of the local time during daytime and a constant level during night-time \[60\], \[83\]. The ionospheric delay can be
5.2 Signal Propagation Errors

determined from the broadcast parameters and the user’s latitude, longitude, satellite elevation, azimuth angles, and local time.

5.2.2 Troposphere

The lower part of the Earth’s atmosphere, the troposphere, consists of dry gases (i.e., the dry component) and water vapor (i.e., the wet component) causing the GNSS signals to be refracted. Water vapor generally exists only below altitudes of 12 km above sea level and most of the water vapor is below 4 km. The dry component of the troposphere (mainly N₂ and O₂ gases) extends to a height of about 40 km. These dry gases, however, can be found in gradually thinning layers at altitudes of hundreds of meters [83]. About 90 percent of the tropospheric delay is due to the dry component and it is easily predictable based on the user latitude, altitude, and season. The wet atmosphere consisting of water vapor is much harder to estimate since it varies with local weather and can change rapidly. Models of the troposphere attempt to estimate the dry and wet refractivities along the signal path in order to predict the total tropospheric delay [83]. As distinct from the ionosphere, the troposphere is non-dispersive, and hence, it induces the same delay on both signal code and carrier.

5.2.3 Interference and Jamming

The satellite navigation frequency bands are well-protected by international and Federal Communication Commission (FCC) frequency assignments. However, as GNSS signals are rather weak, they are vulnerable to unintentional interference, and possibly even intentional interference [60], [92]. Any radio-navigation system can be disrupted by interference of sufficiently high power. Terrestrial interference caused by out-of-band emissions of other signal sources, such as broadcast television, mobile and fixed Very High Frequency (VHF) and Ultra High Frequency (UHF) transmitters, and ultra-wideband radar and communications, may produce harmonics in the desired GNSS frequency bands. This kind of interference can result either in degraded navigation performance or even in a complete loss of receiver tracking.

The intentional emission of RF energy of sufficient power and characteristics to prevent receivers in a target area from tracking GNSS signals is called jamming [38]. Jamming can be accomplished by continuous waves, wideband, narrowband, or GNSS-type signals typically exceeding the GNSS signal power by around 40 dB to jam an already locked GNSS receiver. Techniques to
improve jam-resistant GNSS receivers may be classified into precorrelation methods that are waveform-specific and include adaptive spatial, temporal, and spectral processing [38]. In addition, Receiver Autonomous Integrity Monitoring (RAIM) and FDE (Fault Detection and Exclusion) methods can also be used to mitigate the effects of unintentional and intentional interference by detecting the inconsistency [65]. A more comprehensive illustration on interference and jamming can be found in [13], [60], [85] and [92].

5.3 Receiver-based Errors

A receiver cannot perfectly follow changes in the signal waveform due to receiver-based errors which include antenna error, inter-channel biases, thermal noise, interference, signal quantization noise and tracking error. Inherent receiver noise has random effects on the precision of the code and carrier measurements. The measurement error due to receiver noise varies with the signal strength (i.e., SNR), which, in turn, varies with the satellite navigation angle [83]. Any source of interference that spectrally overlaps with the GNSS signals will have some impact on the tracking performance since it will enter the receiver front-end and will not be filtered out. In the delay locked loop of a GNSS receiver, the dominant error sources are the thermal noise jitter and the effects of dynamic stress, whereas the secondary error sources include code hardware and software resolution, and oscillator stability [60].

5.4 The Major Challenge: Multipath

Multipath errors are mostly due to reflected GNSS signals from surfaces (such as buildings, metal surfaces etc.) near the receiver, resulting in one or more secondary propagation paths. These secondary path signals, which are superimposed on the desired direct path signal, always have a longer propagation time and can significantly distort the amplitude and phase of the direct path signal. This eventually leads to a deformation in the correlation function as shown in Fig. 5.1, where a direct LOS signal is added constructively with an in-phase (i.e., $0^\circ$ phase difference), delayed (0.5 chips delayed) and attenuated (-3 dB attenuated) version of it to form a compound signal. The deformed correlation shape introduces an error bias in the pseudorange measurement that resulted in a degraded positioning performance.

In severe multipath environments like those in dense urban areas, it may be possible that the LOS signal is obstructed completely and only the reflected
signals are present. These multipath effects on the code phase measurements are most crucial, and the multipath error can reach up to a few tens of meters, or a couple of hundred at most [42]. Moreover, unlike other error sources, multipath cannot be reduced through differential processing, since it decorrelates spatially very rapidly. All these issues are the main driving factors for the research conducted in the context of this thesis striving for an optimum correlation-based multipath mitigation technique in terms of mitigation performance as well as implementation complexity.

5.4.1 Influence of Signal and Receiver Parameters on Multipath Error

The way multipath affects the tracking and navigation performance of a receiver depends on a number of signal and receiver parameters. Among them, the most influential parameters are:

- Type of signal modulation,
- Front-end filter bandwidth (i.e., precorrelation bandwidth),
- Correlator spacing used in the code tracking,
- Type of discriminator used to run the DLL (i.e., nEML, HRC, etc.),
- Code chipping rate,
- Number of multipath signals,
- Amplitudes, delays and phases of multipath signals with respect to the LOS signal, etc.

The type of signal modulation basically determines the shape of the correlation function. For example, BPSK is used to modulate GPS L1 C/A signal, which has a single significant tracking peak within ±1 chip delay from the correct code delay, whereas CBOC modulations (i.e. CBOC(+) for data channel and CBOC(-) for pilot channel) are used to modulate Galileo E1 signal, each of which has more than one significant tracking peak within ±1 chip delay from the correct code delay. Non-coherent (i.e., absolute value of the correlation function) correlation functions for the above modulations are shown in Fig. 5.2, where the extra peaks can be clearly observed in case of CBOC modulations.

![Figure 5.2: Non-coherent correlation functions for different signal modulations.](image)

modulations. This is the situation in the most ideal single path scenario. The situation would get far worse in the presence of multipath signals, for example, in a typical fading channel model with a two to four path assumption. Fig. 5.3 shows the distorted correlation shapes of different signal modulations in a two path static channel with path delays [0 0.1] chips and with path powers [0 -6] dB. As seen in Fig 5.3, the presence of an additional peak (in case of CBOC(+) and CBOC(-)) due to multipath imposes a challenge for the signal acquisition and tracking techniques to lock to the correct peak. If the receiver fails to lock to the correct peak, a multipath error in the order of few tens of meters is of no surprise.
Figure 5.3: Non-coherent correlation functions for different signal modulations in two path static channel.

The front-end filter bandwidth used for band-limiting the received signal also has some impact on the correlation shape. The bandwidth, if not chosen sufficiently high, may round off the correlation peak as well as flatten the width of the correlation function, as shown in Fig. 5.4. For this particular

Figure 5.4: Non-coherent correlation functions for BPSK modulated GPS L1 C/A signal in different front-end bandwidths.

reason, the choice of correlator spacing depends on the receiver’s available front-end bandwidth (and of course, on the sampling frequency), that follows
the relation: the more the bandwidth, the smaller the correlator spacing. As mentioned in [7], the early-late spacing $\Delta_{EL}$ (i.e., twice the correlator spacing) is related to the front-end bandwidth (double-sided) $BW$ and the code chip rate $f_{chip}$ according to the following equation:

$$\Delta_{EL} \geq \frac{f_{chip}}{BW}$$

(5.1)

The type of the discriminator and the correlator spacing used to form the discriminator function (i.e., nEML, HRC, SC, etc.) determines the behavior of the code discriminator that strongly influences the resulting multipath performance. Generally speaking, a narrower correlator spacing leads to a reduced multipath error and a tracking jitter error, as long as sufficient front-end bandwidth is ensured [21].

The code chipping rate determines the chip length ($T_c$), which ultimately decides the resulting ranging error caused by the multipath. This means that a signal with a larger chip length results in a smaller multipath error contribution. That is why, the modernized GPS L5 signal can offer ten times smaller multipath error contribution than the legacy GPS L1 C/A signal, as it has ten times higher chipping rate than that of L1.

The remaining multipath related parameters (i.e., amplitudes, delays, phases and number of multipath signals) depend on the multipath environment, and have direct influence on the tracking performance of the receiver. These parameters are used to define various simulation models (for example, multipath fading channel model) in order to analyze the performance of different multipath mitigation techniques developed in the context of this thesis along with some of the most promising state-of-the-art techniques.

The echo-only signal reception is one example of a severe case of multipath signal degradation, which occurs when there is no direct LOS signal between the satellite and the receiver. Due to the absence of the LOS signal, it can cause multipath error in the order of few tens of meters. Most of the multipath mitigation techniques measure the pseudorange with respect to the first arriving path, and unfortunately, there is no such straightforward way to detect the reception of the echo-only signal.
Chapter 6

Multipath Mitigation Techniques

Multipath remains a dominant source of ranging errors in GNSS, such as the GPS or the developing European satellite navigation system, Galileo. Several approaches have been implemented for the last few decades to reduce the effect of multipath in a GNSS receiver. Among them, the use of special multipath limiting antennas (i.e., choke ring or multi-beam antennas), the post-processing techniques to reduce carrier multipath, the carrier smoothing to reduce code multipath, and the code tracking algorithms based on the receiver internal correlation technique are the most prominent approaches [21].

In this thesis, the research focus is on the correlation-based multipath mitigation techniques, since they are the most widely used in commercial GNSS receivers. This chapter provides a detail overview of the multipath mitigation techniques, generalized here in four major categories: i. the state-of-the-art techniques, which are already available in commercial receivers; ii. the proposed advanced techniques, which have relatively complex implementation; iii. the proposed simple slope-based technique, which does not require many correlators; and iv. the proposed combined techniques, which are a combination of two different techniques. Among the proposed advanced techniques, the non-coherent MEDLL is proposed in [P2], the basic Peak Tracking and its improved variants are proposed in [P1] and [P2], respectively, and the basic Reduced Search Space Maximum Likelihood (RSSML) delay estimator is proposed first in [P4], and it is later enhanced in [P7]. The proposed simple slope-based multipath mitigation technique, named as Slope-Based Multipath Estimator (SBME), is proposed in [P5], which requires a-priori information
about the slope of the correlation function and an additional correlator to estimate the multipath error. Among the proposed combined techniques, a C/N$_0$-based two-stage delay estimator is proposed in [P6], and a Teager Kaiser (TK) operator combined with nEML DLL is proposed in [P7].

6.1 State-of-the-art Techniques

The state-of-the-art multipath mitigation techniques are already commercially available in GNSS receivers. The GNSS community has started the correlation-based multipath mitigation studies in early 1990s with the advent of the Narrow Correlator (NC) or the narrow Early-Minus-Late (nEML) DLL [21]. This section highlights some of the most prominent state-of-the-art techniques, which have gained a lot of interest in the research community by now.

6.1.1 Early-Minus-Late Delay Locked Loop

The classical correlation-based code tracking structure used in a GNSS receiver is based on a feedback delay estimator and is implemented via a feedback loop. The most known feedback delay estimator is the Early-Minus-Late (EML) DLL, where two correlators spaced at one chip from each other, are used in the receiver in order to form a discriminator function, whose zero crossings determine the path delays of the received signal [2], [10], [16], [36], [37], [68], [73]. The classical EML usually fails to cope with multipath propagation [21]. Therefore, several enhanced EML-based techniques have been introduced in the literature for the last two decades in order to mitigate the impact of multipath, especially in closely spaced path scenarios. A first approach to reduce the influences of code multipath is based on the idea of narrowing the spacing between the early and late correlators, i.e., nEML or narrow correlator [21], [33], [34]. The choice of correlator spacing depends on the receiver’s available front-end bandwidth along with the associated sampling frequency [7]. Correlator spacings in the range of 0.05 to 0.2 chips are commercially available for nEML based GPS receivers [14]. nEML has also been used as a benchmark algorithm for our studied algorithms in publications [P1]-[P8].

6.1.2 Double Delta (ΔΔ) Technique

Another family of discriminator-based DLL variants proposed for GNSS receivers is the so-called Double Delta (ΔΔ) technique, which uses more than three correlators in the tracking loop (typically, five correlators: two early,
one in-prompt and two late) [55]. $\Delta \Delta$ technique offers better multipath rejection in medium-to-long delay multipath in good $C/N_0$ [54, 80]. Couple of well-known particular cases of $\Delta \Delta$ technique are the High Resolution Correlator (HRC) [80], the Strobe Correlator (SC) [41], [55], the Pulse Aperture Correlator (PAC) [57] and the modified correlator reference waveform [55], [131]. One other similar tracking structure is the Multiple Gate Delay (MGD) correlator [4], [9], [31], [32], [56], where the number of early and late gates and the weighting factors used to combine them in the discriminator are the parameters of the model, and can be optimized according to the multipath profile as illustrated in [54], [56]. While coping better with the ambiguities of BOC correlation function, the MGD provides slightly better performance than the nEML at the expense of higher complexity and is sensitive to the parameters chosen in the discriminator function (i.e., weights, number of correlators and correlator spacing) [9], [54], [56]. In [54], it is also shown that the $\Delta \Delta$ technique is a particular case of the MGD implementation.

### 6.1.3 Early-Late-Slope

Another feedback tracking structure is the Early-Late-Slope (ELS) [55], which is also known as the Multipath Elimination Technique (MET) [122]. The ELS is based on two correlator pairs at both sides of the correlation function’s central peak with parameterized spacing. Once both slopes are known, they can be used to compute a pseudorange correction that can be applied to the pseudorange measurement. However, simulation results performed in [55] showed that ELS is outperformed by HRC with respect to Multipath Error Envelopes (MEEs), for both BPSK and SinBOC(1,1) modulated signals. An Improved ELS (IELS) technique was proposed by the Author in [P2], which introduced two enhancements to the basic ELS approach. The first enhancement was the adaptation of random spacing between the early and the late correlator pairs, while the later one was the utilization of feedforward information in order to determine the most appropriate peak on which the IELS technique should be applied. It was shown in [P2] that IELS performed better than nEML only in good $C/N_0$ for BPSK and SinBOC(1,1) modulated signals in case of short-delay multipath, but still had poorer performance than HRC.

### 6.1.4 A-Posteriori Multipath Estimation

A new multipath estimation technique, named as A-Posteriori Multipath Estimation (APME), is proposed in [115], which relies on a-posteriori estimation
of multipath error. Multipath error is estimated independently in a multipath estimator module on the basis of the correlation values from the prompt and very late correlators. The performance of APME in multipath environment is comparable with that of the Strobe Correlator: a slight improvement for very short delays (i.e., delays less than 20 meters), but rather significant deterioration for medium delays [115].

6.1.5 Multipath Estimating Delay Lock Loop

One of the most promising state-of-the-art multipath mitigation techniques is the Multipath Estimating Delay Lock Loop (MEDLL) [86], [87], [123] implemented by NovAtel for GPS receivers. MEDLL uses several correlators per channel in order to determine accurately the shape of the multipath-corrupted correlation function. Then, a reference correlation function is used in a software module in order to determine the best combination of LOS and NLOS components (i.e., amplitudes, delays, phases and number of multipath). An important aspect of the MEDLL is an accurate reference correlation function that can be constructed by averaging the measured correlation functions over a significant amount of total averaging time [87]. However, the MEDLL provides superior multipath mitigation performance to the nEML at a cost of an expensive multi-correlator based delay tracking structure.

6.2 Proposed Advanced Techniques

The advanced multipath mitigation techniques, proposed by the Author in [P1]-[P4] and in [P7], require a significant number of correlators (in the range of 80 to 200 correlators) in order to estimate the channel characteristics, which are then used to mitigate the multipath effect. These multipath mitigation techniques are presented in the following sub-sections, while references are made to the corresponding publications to help readers avoid too many technical details at once.

6.2.1 Non-coherent Multipath Estimating Delay Lock Loop

MEDLL is considered as a significant evolutionary step in the correlation-based multipath mitigation approach. Moreover, MEDLL has stimulated the design of different maximum likelihood based implementations for multipath mitigation. One such variant is the non-coherent MEDLL, developed by the authors, as described in [P2]. The classical MEDLL is based on a maximum
likelihood search, which is computationally extensive. The authors implemented a non-coherent version of the MEDLL that reduces the search space by incorporating a phase search unit, based on the statistical distribution of multipath phases. It was shown in [P2] that the performance of this suggested approach depends on the number of random phases considered; meaning that the larger the number is, the better the performance will be. But this will also increase the processing burden significantly. The results reported in [P2], show that the non-coherent MEDLL provides very good performance in terms of RMSE, but has a rather poor Mean-Time-to-Lose-Lock (MTLL) as compared to the conventional DLL techniques.

6.2.2 Peak Tracking

Peak Tracking (PT) based techniques, namely PT based on 2nd order Differentiation (PT(Diff2)) and PT based on Teager Kaiser (PT(TK)), were proposed in [P2]. Both the techniques utilize the adaptive thresholds computed from the estimated noise variance of the channel in order to decide on the correct code delay. The adaptive thresholds are computed according to the equations given in [P2]. After that, the advanced techniques generate the competitive peaks which are above the computed adaptive thresholds. The generation of competitive peaks for PT(Diff2) technique is shown in Fig. 6.1 in a two path Nakagami-m fading channel. For each of the competitive peak, a decision variable is formed based on the peak power, the peak position and the delay difference of the peak from the previous delay estimate. Finally, the PT techniques select the peak which has the maximum weight as being the best LOS candidate. It was shown in [P2] that PT(Diff2) has superior multipath mitigation performance over PT(TK) in a two to five path Nakagami-m fading channel.

6.2.3 Teager Kaiser Operator

The Teager Kaiser based delay estimation technique is based on the principle of extracting the signal energy of various channel paths via a nonlinear TK operator [45], [46]. The output \( \Psi_{TK}(x(n)) \) of the TK operator applied to a discrete signal \( y(n) \), can be defined as [46]:

\[
\Psi_{TK}(y(n)) = y(n - 1)y^*(n - 1) - \frac{1}{2}[y(n - 2)y^*(n) + y(n)y^*(n - 2)]
\]
If a non-coherent correlation function is used as an input to the nonlinear TK operator, it can then signal the presence of a multipath component more clearly than looking directly at the correlation function. At least three correlation values (in-prompt, early and very early) are required to compute the TK operation. But usually, TK based delay estimation utilizes a higher number of correlators (for example, 193 correlators were used in the simulations reported in [P7]) and is sensitive to the noise dependent threshold choice. Firstly, it computes the noise variance, which is then used to compute an adaptive threshold. The peaks which are above the adaptive threshold are considered as competitive peaks. Among all the competitive peaks, TK selects the delay associated to that competitive peak which has the closest delay difference from the previous delay estimate. The TK-based estimator has also been used as a benchmark technique in the publications [P2]-[P4] and [P7].

6.2.4 Reduced Search Space Maximum Likelihood Delay Estimator

The RSSML delay estimator is another good example of a maximum likelihood based approach, which is capable of mitigating the multipath effects reasonably well at the expense of increased complexity. The RSSML, proposed by
the Author in [P4] and then further enhanced in [P7], attempts to compensate the multipath error contribution by performing a nonlinear curve fit on the input correlation function which finds a perfect match from a set of ideal reference correlation functions with certain amplitude(s), phase(s) and delay(s) of the multipath signal. Conceptually, a conventional spread spectrum receiver does the same thing, but for only one signal (i.e., the LOS signal). With the presence of a multipath signal, RSSML tries to separate the LOS component from the combined signal by estimating all the signal parameters in a maximum likelihood sense, which consequently achieves the best curve fit on the received input correlation function. As mentioned in [P7], it also incorporates a threshold-based peak detection method, which eventually reduces the code delay search space significantly. However, the downfall of RSSML is the memory requirement which it uses to store the reference correlation functions.

In a multi-correlator based structure, the estimated LOS delay, theoretically, can be anywhere within the code delay window range of $\pm \tau_W$ chips, though in practice, it is quite likely to have a delay error around the previous delay estimate. The code delay window range essentially depends on the number of correlators (i.e., $M$) and the spacing between the correlators (i.e., $\Delta$) according to the following equation:

$$\tau_W = \pm \frac{(M - 1)}{2} \Delta$$ (6.2)

For example, in [P7], 193 correlators were used with a correlator spacing of 0.0208 chips, resulting in a code delay window range of $\pm 2$ chips with respect to the prompt correlator. Therefore, the LOS delay estimate can be anywhere within this $\pm 2$ chips window range. The ideal non-coherent reference correlation functions are generated for up to $L$ paths only for the middle delay index (i.e., $(\frac{M}{2} + 1)$-th delay index; for $M = 193$, the middle delay index is 97). These ideal correlation functions for the middle delay index are generated off-line and saved in a look-up table in memory. In real-time, RSSML reads the correlation values from the look-up table, translates the ideal reference correlation functions at the middle delay index to the corresponding candidate delay index within the code delay window, and then computes the Minimum Mean Square Error (MMSE) for that specific delay candidate. Instead of considering all possible LOS delays within a predefined code delay window as delay candidates, the search space is first reduced to some competitive peaks which are generated based on the computed noise thresholds as explained in [P7]. This will eventually reduce the processing time required to compute the MMSE (i.e., MMSE needs to be computed only for the reduced
An example is shown in Fig. 6.2, where RSSML estimates a best-fitted non-coherent correlation function at a cost of $3.6 \times 10^{-4}$ MMSE in a two path Rayleigh channel with path delays $[0 \ 0.35]$ chips, path powers $[0 \ -2]$ dB at a $C/N_0$ of 50 dB-Hz.

Figure 6.2: Estimated and received non-coherent correlation functions in two path Rayleigh channel, path delay: $[0 \ 0.35]$ chips, path power: $[0 \ -2]$ dB, $C/N_0$: 50 dB-Hz.

### 6.3 Proposed Simple Slope-based Technique

The Author proposed a simple slope-based multipath mitigation technique, named as Slope-Based Multipath Estimator, in [P5]. Unlike the multipath mitigation techniques discussed above, SBME does not require a huge number of correlators, rather it only requires an additional correlator (as compared to a conventional DLL) at the late side of the correlation function. In fact, SBME is used in conjunction with a nEML tracking loop [P5]. It first derives a multipath estimation equation by utilizing the correlation shape of the ideal normalized correlation function, which is then used to compensate for the multipath bias of a nEML tracking loop. The derivation of the multipath estimation equation for BPSK modulated GPS L1 C/A signal can be found...
in [P5]. It is reported in [P5] that SBME has superior multipath mitigation performance to the nEML in a closely spaced two path channel model.

6.4 Proposed Combined Techniques

The motivation for having a combined approach is to ensure a better multipath mitigation performance with a reasonable implementation complexity than each single combining technique that is used to form the combined multipath mitigation technique. In what follows, two such combining techniques are presented, as proposed by the Author in [P6] and [P7], respectively.

6.4.1 C/N$_0$-based Two-Stage Delay Tracker

A C/N$_0$-based two-stage delay tracker is a combination of two individual tracking techniques, namely nEML and HRC (or MGD). The tracking has been divided into two stages based on the tracking duration and the received signal strength (i.e., C/N$_0$). At the first stage of tracking (for about 0.1 seconds or so), the two-stage delay tracker always starts with a nEML tracking loop, since it begins to track the signal with a coarsely estimated code delay as obtained from the acquisition stage. And, at the second or final stage of tracking (i.e., when the DLL tracking error is around zero), the two-stage delay tracker switches its DLL discriminator from nEML to HRC (or MGD), since HRC (or MGD) has better multipath mitigation capability as compared to nEML. While doing so, it has to be ensured that the estimated C/N$_0$ level meets a certain threshold set by the two-stage tracker. This is because of the fact that HRC (or MGD) involves one (or two in case of MGD) more discrimination than NEML, which makes its discriminator output much noisier than nEML. It has been empirically found that a C/N$_0$ threshold of 35 dB-Hz can be a good choice, as mentioned in [P6]. Therefore, at this fine tracking stage, the two-stage delay tracker switches from nEML to HRC (or MGD) only when the estimated C/N$_0$ meets the above criteria (i.e., C/N$_0$ threshold is greater than 35 dB-Hz).

An example non-coherent S-curve is shown in Fig. 6.3 for a CBOC(-) modulated signal in a single path static channel [P6]. The nearest ambiguous zero crossings for HRC (around ±0.16 chips) is much closer as compared to that of nEML (around ±0.54 chips) in this particular case. This indicates the fact that the probability of locking to any of the side peaks is much higher for HRC than that of nEML, especially in the initial stage of tracking when
the code delay may not necessarily be near the main peak of the correlation function. This is the main motivation to choose a nEML tracking at the initial stage for a specific time duration (for example, 0.1 seconds or so). This will eventually pull the delay tracking error around zero after the initial stage.

### 6.4.2 TK operator combined with a nEML DLL

A combined simplified approach with the TK operator and a nEML DLL was implemented in [P7], in order to justify the feasibility of having a nEML discrimination after the TK operation on the non-coherent correlation function. In this combined approach, the TK operator is first applied to the non-coherent correlation function, and then nEML discrimination is applied to the TK output. The motivation for this combined approach comes from the fact that - when we apply the TK operation to the non-coherent correlation function, it usually makes the main lobe of the non-coherent correlation function (after TK operation) much steeper than before. This eventually reduces the effect of multipath in case of TK based nEML (TK+nEML) as compared to nEML.
Chapter 7

Experimental Analysis

In this chapter, the performance of some of the multipath mitigation techniques are evaluated in different simulation models, i.e., the semi-analytical simulation for the Running Average Error (RAE) analysis, the Matlab-based simulation in a multipath fading channel and the simulation carried out in the TUT Galileo E1 Simulink simulator. Each simulation model is briefly described first before presenting the results obtained with it. It is worth to mention here that more simulation results have been presented in the publications [P1]-[P8] and this chapter only provides a summary of the main findings regarding various multipath mitigation techniques and of the main performance criteria used within the scope of this thesis.

7.1 Semi-analytical Simulation

The most typical way to evaluate the performance of a multipath mitigation technique is via Multipath Error Envelopes (MEE). Typically, two paths, either in-phase or out-of-phase, are assumed to be present, and the multipath errors are computed for multipath delays up to 1.2 chips at maximum, since the multipath errors become less significant after that. The upper multipath error envelope can be obtained when the paths are in-phase and the lower multipath error envelope when the paths are out-of-phase (i.e., 180° phase difference). In MEE analysis, several simplifying assumptions are usually made in order to distinguish the performance degradation caused by the multipath errors only. Such assumptions include zero Additive-White-Gaussian-Noise (AWGN), ideal infinite-length PRN codes, and zero residual Doppler. Under these assumptions, the correlation $R_{rx}(\tau)$ between the reference code of the
modulation type MOD (for example, BPSK or CBOC(-)) and the received MOD-modulated signal via an \( L \)-path channel can be written as:

\[
R_{rx}(\tau) = \sum_{l=1}^{L} \alpha_l e^{j\theta_l} R_{MOD}(\tau - \tau_l) \tag{7.1}
\]

where \( \alpha_l, \theta_l, \tau_l \) are the amplitude, phase, and delay, respectively, of the \( l \)-th path; and \( R_{MOD}(\tau) \) is the auto-correlation function of a signal with the modulation type MOD. The analytical expressions for MEEs become complicated in the presence of more than two paths due to the complexity of channel interactions. Therefore, an alternative Monte-Carlo simulation-based approach is presented here, in accordance with [P7], for multipath error analysis in more than one path scenarios (i.e, for \( L \geq 2 \)). First, a sufficient number of random realizations, \( N_{\text{random}} \) are generated (i.e., in the simulation, we choose \( N_{\text{random}} \) equals to 1000), and then we look at the absolute mean error for each path delay over \( N_{\text{random}} \) points. The objective is to analyze the multipath performance of some of the proposed advanced techniques along with some conventional DLLs in the presence of more than two channel paths, which may occur in urban or indoor scenarios.

The following assumptions are made while running the simulation for generating the RAE curves [49]. According to [49], RAE is computed from the area enclosed within the multipath error and averaged over the range of the multipath delays from zero to the plotted delay values. In the simulation, the channel follows a decaying Power Delay Profile (PDP), which can be expressed by the equation:

\[
\alpha_l = \alpha_l \exp^{-\mu(\tau_l - \tau_1)}, \tag{7.2}
\]

where \((\tau_l - \tau_1) \neq 0 \) for \( l > 1 \), \( \mu \) is the PDP coefficient (assumed to be uniformly distributed in the interval \([0.05; 0.2]\), when the path delays are expressed in samples). The channel path phases \( \theta_l \) are uniformly distributed in the interval \([0; 2\pi]\) and the number of channel paths \( L \) is uniformly distributed between 2 and \( L_{\text{max}} \), where \( L_{\text{max}} \) is set to 5 in the simulation. A constant successive path spacing \( x_{ct} \) is chosen in the range \([0; 1.167]\) chips with a step of 0.0417 chips (which defines the multipath delay axis in the RAE curves). It is worth to mention here that the number of paths is reduced to only one LOS path when \( x_{ct} = 0 \). The successive path delays can be found using the formula \( \tau_l = l x_{ct} \) in chips. Therefore, for each channel realization (which is a combination of amplitudes \( \vec{\alpha} = \alpha_1, \ldots, \alpha_L \), phases \( \vec{\theta} = \theta_1, \ldots, \theta_L \), fixed path spacings, and the number of channel paths \( L \)), a certain LOS delay is estimated \( \tau_1(\vec{\alpha}, \vec{\theta}, L) \).
from the zero crossing of the discriminator function (i.e., \(D(\tau) = 0\)), when searched in the linear range of \(D(\tau)\) in case of conventional DLLs, or directly from the auto-correlation function in case of advanced multi-correlator based techniques. The estimation error due to multipath is \(\hat{\tau}_1(\vec{\alpha}, \vec{\theta}, L) - \tau_1\), where \(\tau_1\) is the true LOS path delay. The RAE curves are generated in accordance with [49]. RAE is actually computed from the area enclosed within the multipath error and averaged over the range of the multipath delays from zero to the plotted delay values. Therefore, in order to generate the RAE curves, the Absolute Mean Error (AME) is computed for all \(N_{\text{random}}\) random points via eqn. 7.3:

\[
\text{AME}(x_{ct}) = \text{mean}\left(\left|\hat{\tau}_1(\vec{\alpha}, \vec{\theta}, L) - \tau_1\right|\right),
\]  

where AME\((x_{ct})\) is the mean of the absolute multipath error for the successive path delay \(x_{ct}\). Now, the running average error for each particular delay in the range \([0;1.167]\) chips can be computed as follows:

\[
\text{RAE}(x_{ct}) = \frac{\sum_{i=1}^{i} \text{AME}(x_{ct})}{i},
\]

where \(i\) is the successive path delay index, and RAE\((x_{ct})\) is the RAE for the successive path delay \(x_{ct}\).

The RAE curves for CBOC(-) modulated Galileo E1C signal (i.e., pilot channel) is shown in Fig. 7.1. It is obvious from Fig. 7.1 that the proposed RSSML and PT(Diff2) show the best performance in terms of RAE as compared to other techniques in this noise-free two to five paths static channel model. Among other techniques, TK+nEML showed very good performance followed by SBME, HRC and nEML. The SBME coefficient and the late slope at very late spacing of 0.0833 chips were determined according to [P5] for a 24.552 MHz front-end bandwidth (double-sided). For the above configuration, the SBME coefficient is 0.007 and the late slope is \(-4.5\).

It is worth to mention here that the RAE analysis is quite theoretical from two perspectives: firstly, the delay estimation is a one-shot estimate, and does not really include any tracking loop in the process; and secondly, the analysis is usually carried out with an ideal noise free assumption. These facts probably explain the reason why an algorithm which performs very well with respect to RAE may not necessarily provide the same performance in a more realistic closed loop fading channel model, especially in the presence of more than two channel paths. However, MEE or RAE analysis has been widely used
Figure 7.1: Running average error curves for CBOC(−) modulated Galileo E1C signal.

by the research community as an important tool for analyzing the multipath performance due to simpler implementation, and also due to the fact that it is hard to isolate multipath from other GNSS error sources in real life.

### 7.2 Matlab-based Simulation

Simulation has been carried out in closely spaced multipath environments for CBOC(-) modulated Galileo E1C (i.e., pilot channel) signal for a 24.552 MHz front-end bandwidth. The simulation profile is summarized in Table 7.1. Rayleigh fading channel model is used in the simulation, where the number of channel paths follows a uniform distribution between two and five. The successive path separation is random between 0.02 and 0.35 chips. The channel paths are assumed to obey a decaying PDP according to Eqn. 7.2, where \((\tau_l - \tau_1) \neq 0\) for \(l > 1\), and the PDP coefficient \(\mu = 0.1\). The received signal is sampled such that there are 48 samples per chip. The received signal duration is 800 milliseconds (ms) or 0.8 seconds for each particular \(C/N_0\) level. The tracking errors are computed after each \(N_c \times N_{nc}\) ms (in this case, \(N_c \times N_{nc} = 20\) ms) interval. In the final statistics, the first 600 ms are ignored in order to remove the initial error bias that may come from the delay difference between the received signal and the locally generated reference code. Therefore, for the above configuration (i.e., code loop filter parameters and the first path
### Table 7.1: Simulation profile description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel model</td>
<td>Rayleigh fading channel</td>
</tr>
<tr>
<td>Number of paths</td>
<td>between 2 to 5</td>
</tr>
<tr>
<td>Path Power</td>
<td>Decaying PDP with $\mu = 0.1$</td>
</tr>
<tr>
<td>Path Spacing</td>
<td>Random between 0.02 and 0.35 chips</td>
</tr>
<tr>
<td>Path Phase</td>
<td>Random between 0 and $2\pi$</td>
</tr>
<tr>
<td>Samples per Chip, $N_s$</td>
<td>48</td>
</tr>
<tr>
<td>E-L Spacing, $\Delta_{EL}$</td>
<td>0.0833 chips</td>
</tr>
<tr>
<td>Number of Correlators, $M$</td>
<td>193(^1)</td>
</tr>
<tr>
<td>Double-sided Bandwidth, BW</td>
<td>24.552 MHz</td>
</tr>
<tr>
<td>Filter Type</td>
<td>Finite Impulse Response (FIR)</td>
</tr>
<tr>
<td>Filter Order</td>
<td>6</td>
</tr>
<tr>
<td>Coherent Integration, $N_c$</td>
<td>20 ms</td>
</tr>
<tr>
<td>Non-coherent Integration, $N_{nc}$</td>
<td>1 block</td>
</tr>
<tr>
<td>Initial Delay Error</td>
<td>$\pm$ 0.1 chips</td>
</tr>
<tr>
<td>First Path Delay</td>
<td>0.2 chips</td>
</tr>
<tr>
<td>Code Tracking Loop Bandwidth</td>
<td>2 Hz</td>
</tr>
<tr>
<td>Code Tracking Loop Order</td>
<td>1(^{st}) order</td>
</tr>
</tbody>
</table>

The simulation has been carried out for 100 random realizations, which give a total of 10\*100=1000 statistical points, for each C/N\(0\) level. The RMSE of the delay estimates are plotted in meters, by using the relationship:

$$\text{RMSE}_m = \text{RMSE}_{\text{chips}} c T_c$$

where $c$ is the speed of light, $T_c$ is the chip duration, and RMSE\(_{\text{chips}}\) is the RMSE in chips.

RMSE vs. C/N\(0\) plot for the given multipath channel profile is shown in Fig. 7.2. It can be seen from Fig. 7.2 that the proposed RSSML clearly achieves the best multipath mitigation performance in this two to five paths closely spaced multipath channel. Among other techniques, PT(Diff2) and HRC have better performance only in good C/N\(0\) (around 40 dB-Hz and onwards). It can also be observed that the proposed SBME and TK+nEML do

\(^1\)Not all the correlators are used in all the tracking algorithms. For example, nEML only requires 3 correlators.
not bring any advantage in the tracking performance as compared to nEML in this multipath fading channel model. Here also, the SBME coefficient and the late slope were set to 0.007 and $-4.5$, respectively.

![Figure 7.2: RMSE vs. C/N₀ plot for CBOC(−) modulated Galileo E1C signal in two to five path Rayleigh fading channel.](image)

**7.3 Simulink-based TUT Galileo E1 Signal Simulation**

All the simulations reported in [P6] were carried out in the TUT Galileo E1 signal simulator. The TUT Galileo E1 signal simulator was developed in a Simulink-based platform at Tampere University of Technology, Finland. The basic version of the Simulink model was created by a former DCE researcher, Hu Xuan in early 2009. Since then, the Simulink model has been extended by the Author by adding several signal processing blocks according to the nature and objective of the research. An overview of the TUT Galileo E1 signal simulator with a tracking-only feature is presented herein in accordance with [P6]. In [P6], a tracking-only model was utilized since the objective was to analyze the performance of the proposed C/N₀-based two-stage delay tracker in multipath scenarios. The tracking-only Simulink model consists of four blocks, as shown in Fig. 7.3. A brief overview of these blocks is presented in the following.
7.3 Simulink-based TUT Galileo E1 Signal Simulation

7.3.1 Transmitter block

The Galileo E1 transmitter block is implemented according to the latest Galileo Signal-In-Space Interface Control Document (SIS-ICD) [30]. The E1B and E1C channels are modeled according to the following equation [30]:

$$\begin{align*}
s_{E1}(t) &= \frac{1}{\sqrt{2}} (e_{E1B}(t)(\alpha sc_{E1B,a}(t) + \beta sc_{E1B,b}(t)) \\
&\quad - (e_{E1C}(t)(\alpha sc_{E1C,a}(t) - \beta sc_{E1C,b}(t)))) \quad (7.6)
\end{align*}$$

where $sc_X(t) = \text{sgn}(\sin(2\pi R_{s,X} t))$, $e_{E1B}(t)$ and $e_{E1C}(t)$ are binary signal components, and $\alpha$ and $\beta$ are weighting factors. Above, $R_{s,X}$ is the sub-carrier rate corresponding to channel $X$ (i.e., either E1B or E1C). As explained in [30], $\alpha = \sqrt{10/11}$ and $\beta = \sqrt{1/11}$. The code length for the Galileo OS signal is 4092 chips (or 4 ms), which is four times higher than the GPS C/A code length of 1023 chips. The block utilizes the frame data with a frame duration of 1 ms. In each frame, $f_s * 10^{-3}$ samples are included, where $f_s$ is the sampling rate. The sampling frequency $f_s$ is a variable of the model, which is usually varying depending on the available front-end bandwidth and the corresponding intermediate frequency ($f_{IF}$). For example, simulation results reported in [P6] were obtained with $f_s = 13$ MHz in case of infinite bandwidth at $f_{IF} = 3.42$ MHz, and $f_s = 52$ MHz in case of 24.552 MHz double-sided bandwidth at $f_{IF} = 13$ MHz.
7.3.2 Channel block

The multipath signals and the complex AWGN are modeled in the channel block according to the following equation:

$$r_{E1}(t) = \sum_{l=1}^{L} \alpha_l(t)s_{E1}(t - \tau_l) + \eta(t)$$  \hspace{1cm} (7.7)

Here, $r_{E1}(t)$ is the received E1 signal at the output of the channel block; $\alpha_l$ and $\tau_l$ are the path gain and path delay for the $l^{th}$ path, respectively; and $\eta(t)$ is the complex AWGN.

7.3.3 Front-end filter block

Simulink’s ‘Digital Filter Design’ toolbox is used to design the front-end filters in all the simulations (where finite front-end bandwidth is assumed) reported in [P6]. A 6th order Chebyshev type I filter with a 24.552 MHz double-sided bandwidth is used in the simulation.

7.3.4 Tracking block

The tracking block consists of three major blocks: ‘Carrier wipe-off block’, ‘Code NCO block’, and ‘Dual channel correlation and discriminators block’, as described in detail in [P6] and [110]. The incoming signal is down converted to the baseband in the ‘Carrier Wipe-Off’ block. After the carrier wipe-off, the real part and the imaginary part of the complex signal are separated as the in-phase channel (or, I channel) and the quad-phase channel (or, Q channel) in baseband. The ‘Code NCO’ block considers the estimated code phase error from the DLL in order to shift the code phase accordingly. This block generates four signals as output: the adjusted E1B and E1C replicas, the trigger enabling signal and the shifted NCO phase. The trigger enabling signal is used in conjunction with ‘tracking_en’ which eventually enables both FLL/PLL and DLL blocks of the E1B and E1C channels (when both the variables are set to 1). Both the code and carrier NCOs were implemented using a C-language based S-function, the details of which are not addressed here for the sake of compactness. The variable ‘tracking_en’ is intentionally set to 1 in order to run the tracking block continuously. In addition, the initially estimated frequency and the initially estimated code delay, which are to be used by the tracking block, are also set such that the initial frequency error is less than 88 Hz, and the initial code delay error is less than 0.1 chips. The
reason for such a scheme is to run the tracking block independently in order to be able to calculate the tracking error in terms of RMSE.

**C/N₀ Estimation**

At the tracking stage, the C/N₀ estimation is performed based on the ratio of the signal’s wideband power to its narrowband power as discussed in [92]. In this method, the power of the signal is computed over a wide bandwidth with a relatively short coherent integration time and over a narrow bandwidth with a longer coherent integration time. In the simulation, the wideband power is computed after 4 ms of coherent integration (after each code epoch length), and the narrowband power is computed after 16 ms of coherent integration in order to estimate the carrier-to-noise density ratio for each particular channel.

### 7.3.5 Simulation Results

Fig. 7.4 shows the RMSE versus C/N₀ plots in a two path static channel model with path delays [0.1 0.2] chips and with path powers [0 -3] dB in a 24.552 MHz double-sided bandwidth.

![Graph showing RMSE vs C/N₀ plots](image)

**Figure 7.4:** RMSE vs. C/N₀ plots in a two path static channel in a 24.552 MHz double-sided bandwidth [P6].
MHz double-sided bandwidth. Both HRC and MGD lock to a side peak at 35 dB-Hz, whereas the two-stage delay trackers can avoid the false lock problem and have similar performance like nEML (or NEML) at 35 dB-Hz C/N₀ and lower. When the estimated C/N₀ is above 35 dB-Hz (from 40 dB-Hz and onwards in the plots), the two-stage delay tracker switches to HRC (or MGD) at the fine tracking stage (after 0.1 seconds of tracking). This eventually leads to a better overall tracking performance as compared to tracking with a single technique (either nEML or HRC).
Chapter 8

Summary of Publications

The second part of this compound thesis consists of eight publications [P1]-[P8]: five articles published in international conferences, two articles in international journals and one article in a renowned GNSS-related magazine. None of these publications has been used (or planned to be used) as a part of any other dissertation. In this chapter, the results of the publications and the contributions of the Author of this thesis are summarized.

8.1 Overview of the Publication Results

A novel Peak Tracking algorithm was proposed in [P1] as a delay estimation technique, which utilizes the advantages of both feedforward and feedback techniques and thereby improves the delay estimation accuracy. The proposed technique combines the feedback (as used by a conventional DLL) and the feedforward techniques in such a way that it increases the delay estimation accuracy while preserving a good mean-time-to-lose-lock. Simulation results in Nakagami-\(m\) multipath fading channel model were presented in order to compare the performance of the proposed technique with some of the conventional feedback DLLs along with few other promising feedforward techniques, previously studied in [73] and [77]. It was shown in [P1] that the proposed PT algorithm distinctively shows the best performance in the presented Nakagami-\(m\) fading channel model at moderate to high \(C/N_0\) in terms of RMSE, while preserving an MTLL close to that of feedback DLLs.

In [P2], two variants of the basic Peak Tracking based delay estimation technique were proposed and implemented for GPS and Galileo signals. Peak Tracking with 2\textsuperscript{nd} order Differentiation PT(Diff2) and Peak Tracking with
Teager Kaiser PT(TK) operator are the two variants of PT, which utilize the inherent advantages of both the feedback and feedforward structures. We remark here that the basic PT introduced in [P1] was using only the second order derivative estimates and it is valid only for the SinBOC(1,1) modulated signal, while the enhanced PT from [P2] has been extended to BPSK modulation and includes also a TK-based approach. In [P2], the authors also proposed an Improved Early-Late-Slope multipath elimination technique and a unique non-coherent implementation of the Multipath Estimating Delay Locked Loop, where the phase information was searched via statistical assumptions. Extensive simulation results in both limited and unlimited receiver bandwidths were presented for nine different multipath mitigation techniques. It was shown that among all the considered techniques, the best trade off between RMSE and MTLL was achieved by PT(Diff2), which reduced the delay estimation error considerably at moderate-to-high C/N₀, while preserving a better MTLL compared with other feedforward techniques. It was also shown that our implementation of the non-coherent MEDLL offered the best multipath mitigation performance in RMSE sense, but suffered from poor MTLL.

The advanced multi-correlator based feedforward techniques, previously proposed by the authors in [P1] and [P2], were implemented in closed loop (or feedback) model (i.e., in the presence of an NCO and a loop filter) in publication [P3]. The multipath performance of some promising feedforward techniques (i.e., PT(Diff2) and TK) along with the conventional DLLs were presented in terms of RAE and RMSE for three different signal modulations, including the newly proposed MBOC modulation. It was shown in [P3] that the advanced feedforward techniques showed much better multipath mitigation performance than the traditional DLLs at moderate-to-high C/N₀ for all three modulated signals (namely, BPSK, SinBOC(1,1) and MBOC) in both closed and open loop models. It was also shown that - under the given circumstances, the closed loop and open loop models provide very similar tracking performance. The equivalence between closed loop (feedback) and open loop (feedforward) delay tracking models has not been proved before to the best of the Author’s knowledge. The multipath improvement of the MBOC signal over BPSK and SinBOC(1,1) signals was also evident from the simulation results.

A maximum likelihood based multipath mitigation technique, namely Reduced Search Space Maximum Likelihood delay estimator was proposed in [P4]. The proposed RSSML is capable of mitigating the multipath effects reasonably well at the expense of increased complexity. The RSSML attempts
to compensate the multipath error contribution by performing a nonlinear curve fit on the input correlation function which finds a perfect match from a set of ideal reference correlation functions with certain amplitude(s), phase(s) and delay(s) of the multipath signal. It also incorporates a threshold-based peak detection method, which eventually reduces the code delay search space significantly. However, the downfall of RSSML is the memory requirement which it uses to store the reference correlation functions. The multipath performance of the newly proposed technique along with the conventional DLLs and the other feedforward techniques was presented in a multipath Rayleigh fading channel model. Here also, BPSK and SinBOC(1,1) modulations were considered along with the CBOC modulation selected for the Galileo E1 signal. It was shown that the RSSML, in general, achieved the best multipath mitigation performance in RMSE sense in a two path Rayleigh fading model.

In [P5], a novel multipath mitigation technique, Slope-Based Multipath Estimator was proposed and tested for Galileo E1 signal and for GPS L1 C/A signal. This new multipath mitigation technique is capable of mitigating the short-delay multipath (i.e., multipath delays less than 0.35 chips) quite well compared to other state-of-the-art mitigation techniques, such as nEML and HRC. The proposed SBME first derives a multipath estimation equation by utilizing the correlation shape of the ideal normalized correlation function of a BPSK- or CBOC-modulated signal, which is then used to compensate for the multipath bias of a nEML tracking loop. SBME requires an additional correlator at the late side of the correlation function compared to the basic nEML structure and it is used in-conjunction with a nEML tracking loop. The multipath performance of the newly proposed technique along with the conventional DLLs was studied in terms of theoretical RAE and the simulated RMSE for short-delay multipath scenarios. It was shown that SBME provided the best overall performance as compared to nEML and HRC in short-delay multipath scenarios.

In publication [P6], a C/N0-based two-stage delay tracking technique was proposed and implemented in the Simulink-based TUT Galileo E1 signal simulator. The false lock problem of the classical HRC and MGD was addressed in case of the Galileo E1 OS signal. The multipath performance of the proposed two-stage delay trackers was presented along with their respective counterparts in two different bandwidth assumptions. It was shown that the two-stage delay trackers solve the false lock problem while preserving the multipath mitigation performance of HRC or MGD at good C/N0 (i.e., from 40 dB-Hz and onwards).
The novel RSSML delay estimator, as initially proposed in [P4], was enhanced and optimized for more than two path scenarios in [P7]. The RSSML delay estimator, as presented in [P7], requires a large set of correlation functions only for the prompt correlator, not for all possible delays in a predefined code delay window range, and thereby, reduces the memory requirement significantly. In addition, the RSSML was also adapted for a finite bandwidth assumption for any number of paths up to four. Moreover, in [P7], a combined simplified approach with the Teager Kaiser and the narrow EML was proposed and implemented in order to justify the feasibility of having a nEML discrimination after the TK operation on the non-coherent correlation function. The multipath performance of the proposed techniques along with the state-of-the-art DLLs and other advanced techniques was presented in terms of RAE and RMSE. The performance of these techniques were analyzed for the newly defined CBOC modulation along with the existing BPSK and SinBOC(1,1) modulations. It was shown that RSSML, in general, achieved the best multipath mitigation performance for all three different modulations in a two-to-four path closely spaced multipath scenario. However, the proposed RSSML increases the receiver complexity, since it is based on a multi-correlator based delay tracking structure, and at the same time, it requires a good amount of memory to keep the reference non-coherent correlation functions available for the MMSE computation. Therefore, the RSSML and other advanced multipath mitigation techniques are more suitable for professional high-end receivers (i.e., receivers with large front-end bandwidths and high sampling frequencies, offering the possibility of having many closely-spaced correlators at the code tracking stage); whereas for mass-market receivers, the nEML and the HRC still provide the best trade off between performance and complexity.

In publication [P8], a novel spectral analysis of different types of MBOC modulation was presented in an unified manner. The spectral differences between CBOC and TMBOC variants were quantized in terms of C/N0 deterioration in order to analyze the impact of these differences on the system performance. It was shown that the CBOC(-) variant is the best variant in terms of multipath mitigation and tracking error variance, while TMBOC behaves better than CBOC in terms of detection error probability of the demodulated data. It was also shown in [P8] that the spectral differences and the differences between CBOC and TMBOC variants are rather small in terms of the considered performance criteria, especially when the receiver bandwidth is not very high.
8.2 Author’s Contribution to the Publications

The research work for this thesis was carried out at the Department of Communications Engineering (DCE) in Tampere University of Technology (TUT), as part of the Tekes funded research projects “Advanced Techniques for Personal Navigation (ATENA)” during the years 2006 - 2007 and “Future GNSS Applications and Techniques (FUGAT)” during the years 2007 - 2009, of the Academy of Finland funded research project “Digital Signal Processing Algorithms for Indoor Positioning Systems” during the years 2008 - 2011, of the EU FP6 research project “Galileo Receiver for Mass Market (GREAT)” during the years 2006 - 2008, and of the EU FP7 research project “Galileo Ready Advanced Mass Market Receiver (GRAMMAR)” during the years 2009 - 2011. During the work, the Author has been a member of an active research group involved in analyzing and developing signal processing techniques for GNSS receivers. Many of these ideas have originated in formal group meetings as well as in casual discussions within the group, and some of the simulation models (built in Matlab and Simulink) have been designed in cooperation with the co-authors. Therefore, the Author’s contribution cannot be separated completely from the contributions of the co-authors. However, the Author’s contribution to all of the publications included in this thesis has been essential in the sense that he developed the main theoretical framework, developed new techniques, performed the simulations, analyzed the performance, and prepared the manuscripts where he is the main author, and a good part of the manuscript where he is the second author. The main contributions of the Author to the publications are summarized as follows.

In [P1], the Author developed the novel Peak Tracking technique in an open loop model (i.e., without the presence of the NCO and the loop filter), elaborated the operating principles of the developed technique, performed the simulations and analyzed the multipath performance with some well-known feedback and feedforward techniques. The manuscript was also fully written by the Author. The novel idea of having a combined feedback and feedforward strategy was the result of numerous discussions with Dr. Elena Simona Lohan. The co-authors helped to build the simulation model for the Nakagami-\(m\) fading channel model and also implemented the Matched Filter technique used in the simulations for performance comparison.

In [P2], the Author proposed and implemented a bunch of multipath mitigation techniques in close cooperation with the co-authors. Among them, PT(Diff2), PT(TK), and IELS were developed entirely by the Author, whereas
the non-coherent MEDLL was implemented by the co-authors, which was further optimized by the Author for a better multipath performance in a fading channel model. The idea of applying a 2nd order differentiation on the correlation function was first introduced to the Author by Dr. Ridha Hamila. The Author carried out all the simulations and wrote most part of the manuscript, whereas the co-authors cooperated to develop the simulation model.

A closed loop model for evaluating the performance of various multipath mitigation techniques was implemented in publication [P3] together with the co-authors of the paper. The co-author Xuan Hu built the NCO model in Matlab, which was then used to run the simulations. The Author adapted the feedforward techniques to the closed loop model, performed all the required simulations, analyzed the multipath performance of various tracking techniques, and wrote the manuscript.

In [P4], the Author developed the proposed maximum likelihood based multipath mitigation technique RSSML, elaborated the implementation related issues of the new technique, performed the simulations, analyzed the multipath performance of the proposed technique along with few other techniques and wrote the manuscript. The discussions with the co-authors helped in choosing the performance criteria and the benchmark algorithms.

In [P5], the Author proposed the novel slope-based multipath mitigation technique for two GNSS signals (i.e., Galileo E1 signal and GPS L1 C/A signal), derived the multipath equation in an infinite front-end bandwidth case, performed the simulations, and wrote the manuscript. The development of such a novel technique was the outcome of several productive discussions with the co-authors.

In [P6], the Author proposed a C/N0-based two-stage delay tracking technique for the Galileo E1 signal, implemented a C/N0 estimator and the proposed tracking structure in the TUT Galileo E1 signal simulator and wrote most parts of the manuscript. The simulations were carried out in cooperation with the co-authors.

In [P7], the Author enhanced and optimized the RSSML delay estimator and proposed a combined delay tracking method, namely the combined TK plus nEML technique. In addition, the Author developed the simulation model, performed the simulations, analyzed the performance of various multipath mitigation techniques and wrote the manuscript. The discussions with the co-author helped in choosing the performance criteria and the benchmark techniques.

In [P8], the Author analyzed the impact of spectral differences between
8.2 Author’s Contribution to the Publications

various CBOC and TMBOC variants on receiver performance in terms of multipath mitigation via MEE curves as well as via Monte Carlo simulations. The Author also carried out the simulations required for multipath performance assessment. The co-authors derived the exact frequency-domain form of the PSD for CBOC and TMBOC waveforms. The manuscript was written jointly with the co-authors.
Chapter 9

Conclusions

In this thesis, the Author particularly addressed the challenges encountered by a GNSS signal due to multipath propagation. In this regard, the Author analyzed a wide range of correlation-based multipath mitigation techniques in static and fading multipath channels for a group of GNSS signals, more specifically, the interoperable civilian signals from two different navigation systems, i.e., Galileo E1 signal and the modernized GPS L1C signal, along with the existing GPS L1 C/A signal used in almost all the receivers available today. During the process of analyzing these techniques, the Author also proposed several novel multipath mitigation techniques applicable for a wide range of applications (from simple low-cost mass-market receivers to a relatively complex expensive high-end receivers).

In Chapter 1, the challenges, the motivation, the prior art, the scope, and the main contributions of the thesis were discussed. Chapter 2 presented briefly the principles of satellite-based positioning, provided an overview of current and future GNSSs, and discussed the major application areas for GNSS users. The signal and channel model were described in Chapter 3 with respect to BOC modulation and its variants.

After providing a brief introduction on GNSS receiver structure in Chapter 4 with particular attention to signal acquisition and tracking, the Author introduced a multi-correlator based delay tracking structure for the advanced multipath mitigation techniques in order to take decision on the correct code delay. As shown in [P7], the multi-correlator based tracking structure offers a superior tracking performance compared to the traditional nEML DLL at the cost of higher number of correlators. Therefore, this structure is only suitable for professional high-end receivers (i.e., receivers with large front-end
bandwidths and high sampling frequencies, offering the possibility of having many closely-spaced correlators at the code tracking stage).

The various error sources for satellite-based positioning technology were briefly described in Chapter 5 with a major focus on multipath error, as being the most challenging error due to its uncorrelated behavior. The effects of various signal and receiver parameters on signal tracking performance, and the relation between these parameters and the multipath error were also discussed here to better understand the research problem addressed in this thesis.

After providing a detailed overview of some of the most promising state-of-the-art multipath mitigation techniques in Chapter 6, the Author presented the novel multipath mitigation techniques, which were originally proposed by the Author in various publications from [P1]-[P7]. The performance of these multipath mitigation techniques were evaluated in terms of different performance criteria, such as running average error and root-mean-square error in different simulation models as described in Chapter 7. A general comparison of all these techniques is summarized briefly in Fig. 9.1, considering the issues related to multipath performance and required implementation complexity.

The implementation complexity of any multipath mitigation technique mainly depends on the correlation structure and the implementation issues concerning channel estimation, correlator requirement, required number of mathematical operations, memory requirement and so on. The advanced mitigation techniques are usually complex, since they generally utilize a large number of correlators (i.e., in the range of 80 to 200 correlators) for channel estimation, which are then used to estimate the first arriving path delay. Among the advanced techniques, the proposed RSSML is the most complex one, since it requires a large set of reference correlation functions as a-priori information. Among the other proposed techniques, SBME and the C/N$_0$-based two-stage delay tracker have the least implementation complexity, since they only require few correlators for the LOS delay estimation. Also, the TK with nEML has moderate implementation complexity, as it applies the TK operation before the nEML discrimination.

Among the proposed advanced techniques, RSSML, in general, achieved the best multipath mitigation performance in moderate-to-high C/N$_0$ scenarios (for example, 30 dB-Hz and onwards). The other advanced techniques, such as PT(Diff2) and non-coherent MEDLL showed good multipath mitigation performance only in high C/N$_0$ scenarios (for example, 40 dB-Hz and onwards). The proposed simple technique SBME offered superior multipath mitigation performance to the well-known nEML DLL at the cost of an ad-
<table>
<thead>
<tr>
<th>Multipath mitigation techniques</th>
<th>Number of complex correlators</th>
<th>Complexity</th>
<th>Performance in multipath</th>
<th>Limitations and observations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>State-of-the-art</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nEML</td>
<td>3</td>
<td>Low</td>
<td>Moderate at low $C/N_0$, low at high $C/N_0$</td>
<td>The most encountered structure in nowadays receivers</td>
</tr>
<tr>
<td>HRC</td>
<td>5</td>
<td>Low</td>
<td>Typically slightly better than nEML</td>
<td>Significant loss of lock problems for BOC/CBOC</td>
</tr>
<tr>
<td>MGD</td>
<td>3-9</td>
<td>Low</td>
<td>Typically slightly better than nEML</td>
<td>Relies on parameter optimization according to environment</td>
</tr>
<tr>
<td>MEDLL</td>
<td>&gt;10</td>
<td>High</td>
<td>Good</td>
<td>A maximum likelihood approach that estimates also parameters not relevant to GNSS positioning</td>
</tr>
<tr>
<td>APME</td>
<td>&gt;=4</td>
<td>Low</td>
<td>Moderate</td>
<td>Better performance than nEML</td>
</tr>
<tr>
<td>ELS</td>
<td>5</td>
<td>Low</td>
<td>Low/Moderate</td>
<td>Very little advantage (if any) over nEML</td>
</tr>
<tr>
<td>TK</td>
<td>&gt;=80</td>
<td>High</td>
<td>Good at high $C/N_0$</td>
<td>Sensitive to noise</td>
</tr>
<tr>
<td><strong>Proposed by Author</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-coherent MEDLL</td>
<td>Tens</td>
<td>Moderate to High</td>
<td>Good at high $C/N_0$</td>
<td>Sensitive to the choice of parameters; trading-off complexity for performance</td>
</tr>
<tr>
<td>Improved ELS</td>
<td>5</td>
<td>Low</td>
<td>Low/moderate</td>
<td>Outperforms nEML in good $C/N_0$; worse than HRC</td>
</tr>
<tr>
<td>SBME</td>
<td>4</td>
<td>Low</td>
<td>Better performance than nEML</td>
<td>Best trade-off between complexity and performance</td>
</tr>
<tr>
<td>RSSML</td>
<td>&gt;=80</td>
<td>High</td>
<td>Very good</td>
<td>Requires a large set of correlation function as a-priori information; offers the best multipath mitigation performance in harsh environments</td>
</tr>
<tr>
<td>PT variants (PT(Diff2), PT(TK))</td>
<td>&gt;=80</td>
<td>High</td>
<td>Good at high $C/N_0$</td>
<td>Sensitive to noise-dependent threshold choice</td>
</tr>
<tr>
<td>C/$N_0$ based two-stage delay tracker</td>
<td>5-9</td>
<td>Low</td>
<td>At low $C/N_0$, performance same as nEML, and at moderate-to-high $C/N_0$, performance same as HRC or MGD</td>
<td>Alleviates the false lock problem of HRC or MGD</td>
</tr>
<tr>
<td>TK with nEML</td>
<td>7</td>
<td>Moderate</td>
<td>Typically slightly better than nEML</td>
<td>Like HRC or MGD, false lock problem exists</td>
</tr>
</tbody>
</table>

Figure 9.1: Comparison of multipath mitigation techniques.
ditional correlator. The proposed combined technique, C/N$_0$-based two-stage delay tracker offered a better tracking accuracy than its individual counterpart (i.e., nEML and HRC), and also alleviated the false lock problem of HRC and MGD. The other combining technique TK with nEML DLL showed slightly better performance than nEML, but it also suffered from the false lock problem like HRC and MGD.

To summarize the discussion, it can be said that RSSML and other advanced multipath mitigation techniques proposed in this thesis are more suitable for professional receivers due to their relatively high complexity; whereas for mass-market receivers, SBME and the C/N$_0$-based two-stage delay tracker are the best trade-off between performance and complexity.

Parts of the Author’s work have been used as a basis of further research in other research units over the world. For example, the multipath mitigation techniques proposed in [P2] have been cited in [23], [81], [82], [84], [100], [107], [108], [109] and [132]. The SBME multipath mitigation technique presented in [P5] and the C/N$_0$-based two-stage delay tracker presented in [P6] have also been employed and evaluated in [56]. An enhanced version of SBME has been selected to be implemented in the prototype receiver of the EU FP7 research project ‘GRAMMAR’ [44].

As the GNSS research area is fast evolving with many potential applications, it remains a challenging topic for future research to investigate the feasibility of the proposed novel techniques with the multitude of signal modulations, spreading codes, and spectrum placements that are (or are to be) proposed. Although the simulation tools used in this thesis were designed to meet the actual conditions, it would be really meaningful to analyze the performance of some of the proposed most promising multipath mitigation techniques (for example, RSSML, PT and SBME) on some measurement-based satellite-to-receiver multipath channel models, for example, the multipath channel models proposed in [70], [71], [117] and [118].
Bibliography


[38] G. Lachapelle. Advanced GNSS theory and applications. ENGO 625 lecture notes, University of Calgary, Fall 2007.


Publications
Publication P1

M. Z. H. Bhuiyan, E. S. Lohan and M. Renfors, “Peak Tracking Algorithm for Galileo-Based Positioning in Multipath Fading Channels” in Proc. of IEEE International Conference on Communications, pp.: 5927 - 5932, 24-28 June 2007, Glasgow, Scotland.

DOI= http://dx.doi.org/10.1109/ICC.2007.982

Copyright ©2007 IEEE. Reprinted with permission.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the Tampere University of Technology's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org. By choosing to view this material, you agree to all provisions of the copyright laws protecting it.
Peak Tracking algorithm for Galileo-based positioning in multipath fading channels

Mohammad Zahidul H. Bhuiyan, Elena Simona Lohan, and Markku Renfors
Institute of Communications Engineering, Tampere University of Technology
P.O.Box 553, FIN-33101, Finland;
{mohammad.bhuiyan; elena-simona.lohan; markku.renfors}@tut.fi

I. ABSTRACT

Line-of-Sight (LOS) delay estimation with high accuracy is a pre-requisite for reliable location via satellite systems. The future European satellite positioning system, Galileo, uses spread-spectrum signals modulated via Binary-Offset-Carrier (BOC) modulation. The receiver for a BOC-modulated spread spectrum signal has to cope not only with multipath effects, but also with possible lost of lock due to additional peaks in the envelope of the correlation function. Traditionally, code tracking is implemented at the receiver side via feedback delay locked loops. Feedforward methods have also been presented as alternatives for increased delay estimation accuracy, especially for short-delay multipaths. The increase in the delay estimation accuracy is typically counter-balanced by a faster Mean Time to Lose Lock (MTLL). In this paper we introduce a new algorithm, namely the Peak Tracking (PT) algorithm, which combines the feedback technique with the feedforward technique, in such a way that it increases the delay estimation accuracy while preserving a good MTLL.

II. INTRODUCTION

Code synchronization is a fundamental pre-requisite for the good performance of a spread spectrum receiver. The signals specified for the future European satellite system, Galileo, are spread spectrum signals, employing various types of BOC modulation [5]. Among them, Open Service signal, which is the signal of interest here, uses sine BOC(1,1) modulation, which signifies that the signal at chip rate \( f_c \) is multiplied with a rectangular sub-carrier with frequency rate equal to the chip rate [1], [5]. The main algorithms used (in literature and receiver implementations) for Galileo code tracking are those used for other CDMA receivers, such as GPS receivers, which are based on what is typically called a feedback delay estimator (because they use a feedback loop). The most known feedback delay estimators are the Delay Locked Loops or Early-Minus-Late (EML) loops [3], [4], [8]. The classical EML fails to cope with multipath propagation [11]. Therefore, several enhanced EML-based techniques have been introduced in order to mitigate the effect of multipaths, especially in closely spaced path scenarios. One class of these enhanced EML techniques is based on the idea of narrowing the spacing between early and late correlators, i.e., narrow EML (nEML) [2], [6], [10]. Another class of enhanced EML structures, denoted by High Resolution Correlator (HRC) uses an increased number of correlators at the receiver: besides the early, in-prompt and late correlator, a very-early and a very-late correlator are added, for better coping with code multipath mitigation for medium and long delay multipath as compared to the conventional EML [10]. The feedback loops typically have a reduced ability to deal with closely spaced path scenarios under realistic assumptions (such as the presence of errors in the channel estimation process), a relatively slow convergence, and the possibility to lose the lock (i.e., start to estimate the delays with high estimation error) due to the feedback error propagation.

Alternatively, various feedforward approaches have been proposed in the literature and they have been summarized for Galileo signals in [9]. While improving the delay estimation accuracy, these approaches might require more correlators than EML approaches and are sensitive to the noise-dependent threshold choice.

Our paper introduces a new delay tracking algorithm, the Peak tracking (PT) algorithm, which takes the advantages of both feedback and feedforward techniques, and combines them in such a way to reduce the LOS delay estimation error, while preserving a good mean-time-to-lose lock. Simulation results in multipath fading channels are included, in order to compare the performance of the proposed algorithm with the performance of various feedback and feedforward algorithms (which are briefly reviewed here).

III. BENCHMARK DELAY TRACKING ALGORITHMS AND TERMINOLOGY

The aim of the development of the novel PT algorithm is to find such an algorithm that fully utilizes the advantages of both feedforward and feedback techniques and improves the fine delay estimation. PT uses the adaptive threshold obtained from the feedforward loop in order to determine the competitive delays, i.e., the delays which are competing as being the actual delay (i.e., the delay of the first arriving path). The adaptive threshold is based on the estimated noise variance of the absolute value of the Auto-Correlation Function (ACF) between the received signal and the locally generated reference signal. At the same time, PT explores the advantage of feedback loop by calculating weight factors based on the previous estimation in order to take decision about the actual delay. However, the utilization of feedback loop is always a challenge since there is a chance to propagate the delay error.
to subsequent estimations. Therefore, the delay error should remain in tolerable range (for example, less than or equal to half of the width of main lobe) so that the advantage from feedback loop could be properly utilized.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{ideal_acf.png}
\caption{Ideal ACF of sine BOC(1,1) modulated signal}
\end{figure}

For a sine BOC(1,1) modulated signal, the width of main lobe of the envelope of an ideal ACF of the locally generated reference signal is about 0.7 chips as seen from Fig. 1 (here the real part of ACF is shown; the envelope width can be seen from the positive values of ACF). Thus, we assume in what follows a maximum allowable delay error as less than or equal to half of the width of the main lobe (i.e., 0.7/2 = 0.35 chips). This means that, if the delay error is higher (in absolute value) than 0.35 chips, the lock is considered to be lost and the acquisition and tracking processes should be restarted.

The terminologies used in the PT algorithm are defined in this section. We also review here the other delay estimation techniques selected for comparison with our algorithm. Among the feedback delay tracking algorithms, we have selected the narrow EML and the HRC [2], [6], [10]. Both are well-known from literature and are not described again here. It suffices to say that nEML uses three correlators (early, in-prompt and late), with an early-late spacing of 0.1 chips, while HRC uses 5 correlators (very early, early, in-prompt, late and very late), with an early-late spacing of 0.1 chips and a very early-very late spacing of 0.2 chips. Among the feedforward delay tracking algorithms, we have selected the Matched Filter (MF) and the Second-Order Derivative (Diff2) algorithms, which are described in the next sub-sections, and which represent also parts of the building blocks of PT algorithm. The procedure of the proposed PT algorithm is explained with some illustrative figures in section IV.

A. AACF Peak and Matched Filter (MF) algorithm

In this context, the term AACF peak is defined as any local maximum point from the Absolute value of the ACF, which is greater than a specific threshold (i.e., \(ACF_{\text{Thresh}}\), as explained in section III-D). The AACF peaks \(AACF_{\text{Peak}}\) are actually the normalized amplitudes of local maximum points of the AACF and they can be obtained using the following equation:

\[
AACF_{\text{Peak}} = \forall x_i \{(x_i \in ACF) \land (x_i \geq x_{i-1}) \land (x_i \geq x_{i+1}) \land (x_i \geq ACF_{\text{Thresh}})\};
\]

\[
i = 2, 3, \cdots, l_{AACF} - 1
\]

where \(AACF\) stands for the absolute of the auto-correlation function between the received signal and the locally generated reference signal, \(\land\) is the intersection (‘and’) operator, and \(l_{AACF}\) is the length of the set \(AACF\). Above, it was assumed that the samples of AACF are denoted via \(x_i\). In what follows, we refer to this method as Matched Filter (MF) method, by analogy with [9].

B. Diff2 Peak and Diff2 method

Second-order derivative (Diff2) peak is defined as any local maximum point of the second-order derivative of AACF, which is greater than a specific threshold (i.e., \(Diff2_{\text{Thresh}}\), as explained in section III-E). The Diff2 peaks \(Diff2_{\text{Peak}}\) are also normalized with respect to the maximum value of the second-order derivative of AACF. We have:

\[
Diff2_{\text{Peak}} = \forall x_i \{(x_i \in Diff2) \land (x_i \geq x_{i-1}) \land (x_i \geq x_{i+1}) \land (x_i \geq Diff2_{\text{Thresh}})\};
\]

\[
i = 2, 3, \cdots, l_{Diff2} - 1
\]

where \(Diff2\) is the second-order derivative of the AACF and \(l_{Diff2}\) is the length of the set \(Diff2\). Since the maxima of AACF corresponds also to maxima in its second order derivative, the Diff2 method simply estimates as LOS delay the delay of the first \(Diff2_{\text{Peak}}\).

In Fig. 2, AACF peak and Diff2 peaks are marked according to the definition described earlier. Fig. 2 represents a plot for 2 path Rayleigh fading channel model with fixed path separation of 0.15 chips. Here, the path powers are [-2 0] dB and CNR is considerably high, i.e., 100 dB-Hz, in order to emphasize the multipath channel effect. According to Fig. 2, it is visible that the first Diff2 peak corresponds to the delay of the LOS path whereas the first and only AACF peak corresponds to the delay of the second path which is 0.15 chips away from the first path. The experience from the simulation emphasizes the fact that Diff2 could distinguish very closely spaced paths whereas MF algorithm may fail to distinguish very closely spaced paths. However, Diff2 operator is very sensitive to noise specially in low CNR, and therefore, Diff2 is expected to have poor performance in low CNR scenarios.

C. Noise Threshold

The noise Threshold \(N_{\text{Thresh}}\), used for the computation of \(ACF_{\text{Thresh}}\) and \(Diff2_{\text{Thresh}}\), is obtained by estimating the noise level of the AACF. The noise level is estimated by taking the mean of out-of-peak values of the AACF according to the following equation:

\[
N_{\text{Thresh}} = \text{Mean} \{\forall x_i \{(x_i \in ACF_{\text{OutsideRect}})\};
\]

\[
i = 2, 3, \cdots, l_{AACF_{\text{OutsideRect}}} - 1
\]
D. ACF (or MF) Threshold

ACF Threshold ($ACF_{\text{Thresh}}$) is basically computed from the estimated noise threshold $N_{\text{Thresh}}$ obtained from feedforward loop and a weight factor $W_{AACF}$ using the following equation:

$$ACF_{\text{Thresh}} = \max\{ACF_{\text{Peak}}\}W_{AACF} + N_{\text{Thresh}}, \quad (4)$$

where $ACF_{\text{Peak}}$ and $N_{\text{Thresh}}$ were defined in sections III-A and III-C, respectively and $W_{AACF}$ is defined as follows:

$$0.3 \leq W_{AACF} \leq 0.35 \quad (5)$$

$W_{AACF}$ was chosen very carefully taking into consideration the side lobes of ACF of sine BOC(1,1) modulated signal. Eq. 4 represents an ideal non-coherent ACF of sine BOC(1,1) modulated signal where the side lobe peaks correspond to the value 0.25, approximately. Therefore, $W_{AACF}$ could be chosen according to eq. (5) in order to avoid side lobe peaks being considered as the competitive peaks. Definition of competitive peaks is presented in section III-F.

where $ACF_{\text{OutsideRect}}$ is the set consists of those ACF points which fall outside the rectangular window shown in Fig. 3 and $l_{ACF_{\text{OutsideRect}}}$ is the length of the set $ACF_{\text{OutsideRect}}$. The rectangular window is chosen such that it can enclose the side lobe peaks of the ACF, due to BOC modulation, and also due to multipath effects. Hence, the width of the rectangular window should not be less than 2 chips. In this example case, the width of the rectangular window was set to 2.4 chips (based on an estimated of multipath delay spread according to the detected local peaks of ACF) and the value for $N_{\text{Thresh}}$ was approximately 0 because of considerably high CNR (i.e., 100 dB-Hz). We recall here that a high CNR value is used just for illustration purposes, in order to show the multipath channel effect.

$ACF_{\text{Thresh}}$ is basically computed from the estimated noise threshold $N_{\text{Thresh}}$ obtained from feedforward loop and a weight factor $W_{AACF}$ using the following equation:

$$ACF_{\text{Thresh}} = \max\{ACF_{\text{Peak}}\}W_{AACF} + N_{\text{Thresh}}, \quad (4)$$

where $ACF_{\text{Peak}}$ and $N_{\text{Thresh}}$ were defined in sections III-A and III-C, respectively and $W_{AACF}$ is defined as follows:

$$0.3 \leq W_{AACF} \leq 0.35 \quad (5)$$

$W_{AACF}$ was chosen very carefully taking into consideration the side lobes of ACF of sine BOC(1,1) modulated signal. Eq. 4 represents an ideal non-coherent ACF of sine BOC(1,1) modulated signal where the side lobe peaks correspond to the value 0.25, approximately. Therefore, $W_{AACF}$ could be chosen according to eq. (5) in order to avoid side lobe peaks being considered as the competitive peaks. Definition of competitive peaks is presented in section III-F.

where $ACF_{\text{OutsideRect}}$ is the set consists of those ACF points which fall outside the rectangular window shown in Fig. 3 and $l_{ACF_{\text{OutsideRect}}}$ is the length of the set $ACF_{\text{OutsideRect}}$. The rectangular window is chosen such that it can enclose the side lobe peaks of the ACF, due to BOC modulation, and also due to multipath effects. Hence, the width of the rectangular window should not be less than 2 chips. In this example case, the width of the rectangular window was set to 2.4 chips (based on an estimated of multipath delay spread according to the detected local peaks of ACF) and the value for $N_{\text{Thresh}}$ was approximately 0 because of considerably high CNR (i.e., 100 dB-Hz). We recall here that a high CNR value is used just for illustration purposes, in order to show the multipath channel effect.

$ACF_{\text{Thresh}}$ is basically computed from the estimated noise threshold $N_{\text{Thresh}}$ obtained from feedforward loop and a weight factor $W_{AACF}$ using the following equation:

$$ACF_{\text{Thresh}} = \max\{ACF_{\text{Peak}}\}W_{AACF} + N_{\text{Thresh}}, \quad (4)$$

where $ACF_{\text{Peak}}$ and $N_{\text{Thresh}}$ were defined in sections III-A and III-C, respectively and $W_{AACF}$ is defined as follows:

$$0.3 \leq W_{AACF} \leq 0.35 \quad (5)$$

$W_{AACF}$ was chosen very carefully taking into consideration the side lobes of ACF of sine BOC(1,1) modulated signal. Eq. 4 represents an ideal non-coherent ACF of sine BOC(1,1) modulated signal where the side lobe peaks correspond to the value 0.25, approximately. Therefore, $W_{AACF}$ could be chosen according to eq. (5) in order to avoid side lobe peaks being considered as the competitive peaks. Definition of competitive peaks is presented in section III-F.

$ACF_{\text{Thresh}}$ is basically computed from the estimated noise threshold $N_{\text{Thresh}}$ obtained from feedforward loop and a weight factor $W_{AACF}$ using the following equation:

$$ACF_{\text{Thresh}} = \max\{ACF_{\text{Peak}}\}W_{AACF} + N_{\text{Thresh}}, \quad (4)$$

where $ACF_{\text{Peak}}$ and $N_{\text{Thresh}}$ were defined in sections III-A and III-C, respectively and $W_{AACF}$ is defined as follows:

$$0.3 \leq W_{AACF} \leq 0.35 \quad (5)$$

$W_{AACF}$ was chosen very carefully taking into consideration the side lobes of ACF of sine BOC(1,1) modulated signal. Eq. 4 represents an ideal non-coherent ACF of sine BOC(1,1) modulated signal where the side lobe peaks correspond to the value 0.25, approximately. Therefore, $W_{AACF}$ could be chosen according to eq. (5) in order to avoid side lobe peaks being considered as the competitive peaks. Definition of competitive peaks is presented in section III-F.
F. Competitive Peaks concept

A Competitive Peak ($C_{Peak}$) can be obtained using the following equation:

$$C_{Peak} = \{(AACF_{Peak}) \cup (Diff_2^2_{Peak})\}$$  (8)

where $AACF$ and $Diff_2$ were defined in sections III-A and III-B, respectively; and, $\hat{L}$ is the cardinality of the set $C_{Peak}$. This means that we combine the delay estimates given by MF and Diff2 and form a set of 'competitive' delays, from which the final estimate will be selected.

Since, for GNSS applications, the point of interest is to find the delay of the first arriving path (i.e., the LOS path), therefore, it would be enough to consider only the first 5 competitive peaks (in their order of arrival). Hence, we assume that: $\max(\hat{L}) = 5$.

Fig. 5 shows the competitive peaks for the same path profile as in Fig. 2 and 3. The competitive peaks are obtained using eq. (8). As it can be seen from Fig. 5, for this particular example, there are in total two competitive peaks which compete to be considered as being the actual delay of the LOS path.

![Competitive peaks of PT algorithm](image)

Fig. 5. Competitive peaks of PT algorithm

In this example, the first competitive peak corresponds to the delay of the first arriving path whereas the second competitive peak corresponds to the delay of the second arriving path which is 0.15 chips apart from the first path.

IV. PROCEDURE OF PT ALGORITHM

The general architecture of PT algorithm is shown in Fig. 6. In what follows, the step by step procedure of PT algorithm is presented.

A. Step 1: Noise Estimation

$N_{Thresh}$ is estimated according to section III-C, which is then used to determine $AACF_{Thresh}$ and $Diff_2^2_{Thresh}$ with the help of eqs. (4) and (6). These thresholds are then provided as input to the next step.

B. Step 2: Competitive Peak Generation

Step 2(a): Look for AACF peak(s) in AACF domain using eq. (1).

Step 2(b): Look for Diff2 peak(s) in Diff2 domain using eq. (2).

Step 2(c): Find competitive peak(s) using eq. (8).

The competitive peak(s) obtained from step 2 are then fed into steps 3(a), 3(b) and 3(c) in order to assign weights in each sub-step for each particular competitive peak.

C. Step 3(a): Weight Based on Peak Height

Assign weight(s) ($a_i$); $i = 1, \ldots, \hat{L}$ based on the competitive peak height(s) using the following equation:

$$a_i = \frac{T_{AACF}(\tau_i) + T_{Diff_2}(\tau_i)}{2}; \ i = 1, \ldots, \hat{L},$$  (9)

where $T_{AACF}$ and $T_{Diff_2}$ are the AACF and Diff2 correlation values, respectively, corresponding to a competitive peak:

$$T_{AACF}(\tau_i) = AACF(\tau_i); \ i = 1, \ldots, \hat{L}$$  (10)

$$T_{Diff_2}(\tau_i) = Diff_2(\tau_i); \ i = 1, \ldots, \hat{L}$$  (11)

D. Step 3(b): Weight Based on Peak Position

Assign weight(s) $b_i$, $i = 1, \ldots, \hat{L}$ based on peak positions in AACF: the first peak is more probable than the second one; the second one is more probable than the third one and so on. This is based on the assumption that typical multipath channel has decreasing power-delay profile. In the simulation, the following weights were used:

$$[b_1 \ b_2 \ b_3 \ b_4] = [10 \ 8 \ 6.2 \ 5.5 \ 5],$$  (12)

where $b_i$, $i = 1, \ldots, \hat{L}$ denotes the weight factor for $i^{th}$ peak. It is logical to assign higher weights for the first few competitive peaks as compared to later peaks since the target is to find the delay of the first path. However, the weights are found through trial-and-error method based on extensive analysis of the Monte Carlo simulation results.

![Mapping of weights](image)

Fig. 7. Mapping of weights ($c_i, i = 1, \ldots, \hat{L}$) based on previous estimation

E. Step 3(c): Weight Based on Previous Estimation

Assign weight(s) $c_i$, $i = 1, \ldots, \hat{L}$ based on the feedback from the previous estimation: the closer the competitive peak is from the previous estimation, the higher the weight would be for that particular competitive peak. Fig. 7 illustrates the mapping of weight(s) ($c_i$) based on the previous estimation. For example, for a delay difference of 0.1 chips from the previous estimation, the weight factor $c_1 = 10$, and for a delay difference of 0.2 chips, the weight factor $c_3 = 9$, and so on.
F. Step 4: Compute Decision Variable

The decision variable (regarding which of the competitive peaks will be declared as LOS peak), $d_i, i = 1, \ldots, L$ is computed according to the following equation:

$$d_i = a_i b_i c_i; \quad i = 1, \ldots, L$$ (13)

G. Step 5: Find Estimated Delay of the LOS Path

The LOS delay can then be obtained via:

$$\hat{\tau}_{\text{LOS}} = \arg \max_{i=1:L} (d_i)$$ (14)

Table I summarizes the weights assigned in the example path profile shown in Fig. 5. In this example case, there are two competitive peaks meaning that we need to assign weights only for those two peaks. In assigning the weights $c_i, i = 1, \ldots, L$, PT assumes that there is no initial error present from the previous estimation. In step 4, the algorithm simply computes the decision variable $d_i, i = 1, \ldots, L$ using eq. (13) for each competitive peak. And, finally, in step 5, PT algorithm selects the peak which has the maximum value for decision variable $d_i$. In our case, it is $d_1$. Therefore, in this example case, the first competitive peak corresponds, correctly, to the delay of the LOS path.

| TABLE I |
| Assignment of weights for Fig. 5 |

<table>
<thead>
<tr>
<th>1st Competitive Peak</th>
<th>$d_1$</th>
<th>$d_2$</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>10</td>
<td>12</td>
<td>96</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

V. Simulation Results

The PT algorithm has been compared with the following feedback and feedforward algorithms: narrow EML (nEML), HRC, MF, and Diff2 (explained above). Nakagami-m distributed channel profiles with 2 and 3 paths were used for all the algorithms (the average path powers are given in each figure’s caption). Nakagami-m distribution was chosen according to the results reported in [7]. The paths are assumed to be separated randomly with a maximum of 0.2 chips separation between successive paths (i.e., closely spaced paths). The simulation assumed no initial delay error coming from the acquisition stage. However, it is assumed that the LOS delay is changing in time. Two models were used here: either LOS delay is increasing linearly, with a slope of 0.05 chips per blocks of $N_c N_{nc}$ ms, where $N_c$ is the coherent integration length and $N_{nc}$ is the non-coherent integration length, or the LOS delay is oscillating randomly (uniform distribution between $-0.1$ and $+0.1$ chips) around an initial point. For each particular CNR, the total number of Monte Carlo iterations was set to $N_{\text{stats}}$, between 1500 and 5000, and the average statistics were computed over $N_{\text{stats}}$ points. The Mean Time to Lose Lock (MTLL) is the average time spent until the error becomes higher than 0.35 chips, while the Root Mean Square Error is computed over those delay errors which are less than 0.35 chips (i.e., only for in-lock condition).

The results are shown in Figs. 8 and 9, for 2 and 3-path fading channels, different LOS delay error variation, and infinite receiver bandwidth (however, more recent results, not included here due to lack of space, also showed similar conclusions in bandwidth-limited situations). An oversampling factor $N_s = 12$ samples per BOC interval was used in the simulations. Increased accuracy in the delay estimates can be obtained, for example, via interpolation, and is a topic of future research. According to Figs. 8 and 9, PT distinctively shows the best performance in terms of both RMSE and MTLL as compared to the other delay estimation algorithms. Similar results have been obtained in other 2 to 4-path channel profiles (from the point of view of PT algorithm). We notice here that MF performs better than Diff2 in 2 path channel profile, but in 3 path channel profile and in high CNRs, Diff2
outperforms MF in terms of RMSE. HRC algorithm is also highly sensitive to low CNR values. In this situation, HRC tends to converge to a false lock point, and thus RMSE and MTLL are highly deteriorating. The relative performance of the algorithms is dependent on the channel profiles, but the proposed PT algorithm tends to preserve its RMSE advantage over all the other algorithms at moderate to high CNRs. At low CNR, narrow correlator seems the best choice (both from the point of view of RMSE and MTTL).

VI. CONCLUSIONS

In this paper, a novel Peak Tracking (PT) delay estimation algorithm has been proposed for sine BOC(1,1)-modulated PRN codes, as those used for Galileo signals. PT algorithm combines both the feedback and feedforward techniques in such a way that it increases the delay estimation accuracy while preserving a good MTLL. Among all the considered algorithms, the proposed PT algorithm distinctively shows the best performance at moderate to high CNRs in terms of RMSE, while preserving a good MTLL. Among all the considered algorithms, the proposed PT algorithm distinctively shows the best performance at moderate to high CNRs in terms of RMSE, while preserving a good MTLL.

ACKNOWLEDGMENT

This work was carried out in the project “Advanced Techniques for Personal Navigation (ATENA)” funded by the Finnish Funding Agency for Technology and Innovation (Tekes). This work was also supported by the Academy of Finland. The authors also want to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

Publication P2


DOI= http://dx.doi.org/10.1155/2008/863629

Copyright ©2010 Mohammad Zahidul H. Bhuiyan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
Research Article

Code Tracking Algorithms for Mitigating Multipath Effects in Fading Channels for Satellite-Based Positioning

Mohammad Zahidul H. Bhuiyan, Elena Simona Lohan, and Markku Renfors

Institute of Communications Engineering, Tampere University of Technology, P.O. Box 553, 33101 Tampere, Finland

Correspondence should be addressed to Elena Simona Lohan, elena-simona.lohan@tut.fi

Received 28 February 2007; Accepted 30 September 2007

The ever-increasing public interest in location and positioning services has originated a demand for higher performance global navigation satellite systems (GNSSs). In order to achieve this incremental performance, the estimation of line-of-sight (LOS) delay with high accuracy is a prerequisite for all GNSSs. The delay lock loops (DLLs) and their enhanced variants (i.e., feedback code tracking loops) are the structures of choice for the commercial GNSS receivers, but their performance in severe multipath scenarios is still rather limited. In addition, the new satellite positioning system proposals specify the use of a new modulation, the binary offset carrier (BOC) modulation, which triggers a new challenge in the code tracking stage. Therefore, in order to meet this emerging challenge and to improve the accuracy of the delay estimation in severe multipath scenarios, this paper analyzes feedback as well as feedforward code tracking algorithms and proposes the peak tracking (PT) methods, which are combinations of both feedback and feedforward structures and utilize the inherent advantages of both structures. We propose and analyze here two variants of PT algorithm: PT with second-order differentiation (Diff2), and PT with Teager Kaiser (TK) operator, which will be denoted herein as PT(Diff2) and PT(TK), respectively. In addition to the proposal of the PT methods, the authors propose also an improved early-late-slope (IELS) multipath elimination technique which is shown to provide very good mean-time-to-lose-lock (MTLL) performance. An implementation of a noncoherent multipath estimating delay locked loop (MEDLL) structure is also presented. We also incorporate here an extensive review of the existing feedback and feedforward estimation algorithms for direct sequence code division multiple access (DS-CDMA) signals in satellite fading channels, by taking into account the impact of binary phase shift keying (BPSK) as well as the newly proposed BOC modulation, more specifically, sine-BOC(1,1) (SinBOC(1,1)), selected for Galileo open service (OS) signal. The state-of-art algorithms are compared, via simulations, with the proposed algorithms. The main focus in the performance comparison of the algorithms is on the closely spaced multipath scenario, since this situation is the most challenging for estimating LOS component with high accuracy in positioning applications.

Copyright © 2008 Mohammad Zahidul H. Bhuiyan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Today, with the glorious advance in satellite navigation and positioning technology, it is possible to pinpoint the exact location of any user anywhere on the surface of the globe at any time of day or night. Since its launch in the 1970s, the United States (US) Navstar global positioning system (GPS), has become the universal satellite navigation system and reached full operational capability in 1990s [1]. This has created a monopoly, resulting in technical, political, strategic and economic dependence for millions of users [2]. In recent years, the rapid improvement and lowered price of computing power have allowed the integration of GPS chips into small autonomous devices such as hand-held GPS receivers, personal digital assistants (PDAs), and cellular phones, increasing the speed of its consumption 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 1990s [3]. This led to two major GNSS decisions: the modernization of the current US GPS, known as GPS II, and the independent European effort to create its own GNSS, known as Galileo [4, 5]. These two systems are now being finalized and are expected to be available to the public by the end of the decade. It is anticipated that once
the new European satellite navigation system Galileo is operational, the vast majority of all user receivers sold will be both GPS and Galileo capable [2]. The benefits of receiving signals from both constellations include improved accuracy, reliability, and availability [2].

Galileo signals, as well as GPS signals, are based on direct-sequence code division multiple access (DS-CDMA) technique. Spread spectrum systems are known to offer frequency reuse, better multipath diversity, better narrowband interference rejection, and potentially, better capacity compared to narrowband techniques [6]. On the other hand, code and frequency synchronization are fundamental prerequisites for the good performance of the receiver. These two tasks pose several problems in the presence of mobile wireless channels, due to the various adverse effects of the channel, such as the multipath propagation, the possibility of having the line-of-sight (LOS) component obstructed by closely spaced non-line-of-sight (NLOS) components, or even the absence of LOS, and the high level of noise (especially in indoor scenarios). Moreover, the fading statistics of the channel and the possible variations of the oscillator clock limit the coherent integration time at the receiver (i.e., the receiver filters which are used to smooth the various estimates of channel parameters cannot have the bandwidth smaller than the maximum Doppler spread of the channel without introducing significant errors in the estimation process) [7–11]. The Doppler shift induced by the satellite is also prone to deteriorate the receiver performance, unless correctly estimated and removed. Moreover, the fading behavior of the channel paths induces a certain Doppler spread, directly related to the terminal velocity. Typical GNSS receivers estimate jointly the code phase and the Doppler shifts or spreads via a two-dimensional search in time-frequency plane. The delay-Doppler estimation is usually done in two stages: acquisition (or coarse estimation), followed by tracking (or fine estimation). The acquisition and tracking stages will be treated here together, assuming implicitly that the frequency-time search space is reduced, for example, via some assistance data (e.g., Doppler assistance, knowledge of previous delay estimates, etc.). In this situation, the delay estimation problem can be seen as a tracking problem (i.e., very accurate delay estimates are desired) with initial code misalignment of several chips or tens of chips and initial Doppler shift not higher than a few tens of Hertz.

One particular situation in multipath propagation is the situation when LOS component is overlapping with one or several closely spaced NLOS components [7, 9–16] making the delay estimation process more difficult. This closely spaced path scenario is most likely to be encountered in indoor positioning applications or in outdoor urban environments, and is the main focus of our paper.

The main algorithms used for GPS and Galileo code tracking, providing a certain sufficiently small Doppler shift, are based on what is typically called a feedback delay estimator and are implemented based on a feedback loop. The most known feedback delay estimators are the delay lock loops (DLLs) or early-minus-late (EML) loops [13, 17–21]. The classical EML fails to cope with multipath propagation [6]. Therefore, several enhanced EML-based techniques have been introduced in order to mitigate the effect of multipaths, especially in closely spaced path scenarios. One class of these enhanced EML techniques is based on the idea of narrowing the spacing between early and late correlators, that is, narrow EML (nEML) [22–24]. Another class of enhanced EML structures uses a modified reference waveform for the correlation at the receiver, that narrows the main lobe of the cross-correlation function, at the expense of deterioration of signal power. Examples belonging to this class are the high resolution correlator (HRC) [24], the strobe correlators [23, 25], the pulse aperture correlator (PAC) [26] and the modified correlator reference waveform [23, 27]. One other similar tracking structure is the multiple gate delay (MGD) correlator [28–30], where the number of early and late gates and the weighting factors used to combine them in the discriminator are parameters of the model. While coping better with the ambiguities of BOC correlation function, MGD may have poorer performance in multipaths than the narrow EML correlator and is very sensitive to the parameters chosen in the discriminator function (i.e., weights and number of correlators) [31].

One more feedback code tracking structure is the early-late-slope (ELS) [32] correlator, also known as multipath elimination technique (MET), which is based on two correlator pairs at both sides of the correlation function’s central peak with parameterized spacing. Based on these two correlator pairs, the slopes of early and late sides of the correlation function can be computed and then, the intersection point will be used for pseudorange correction. However, simulation results performed in [23] showed that ELS technique is outperformed by HRC from the point of view of the multipath error envelopes (MEE), for both BPSK and SinBOC(n,n)-modulated signals. ELS is also outperformed by narrow correlator for very closely spaced paths (i.e., below 0.1 chip separation) and for paths spaced at about 1/2th of the envelope of the correlation function (i.e., 1 chip spacing for BPSK signals and 0.5 chip spacing for SinBOC(n,n) [23]).

The feedback loops typically have a reduced ability to deal with closely spaced path scenarios under realistic assumptions (such as the presence of errors in the channel estimation process), a relatively slow convergence, and the possibility to lose the lock (i.e., they may start to estimate the LOS delay with high estimation error) due to the feedback error propagation. Alternatively, various feedforward approaches have been proposed in the literature and they have been summarized in [8]. While improving the delay estimation accuracy, these approaches are sensitive to the noise-dependent threshold choice.

One of the most promising feedback code tracking algorithms is the multipath estimating delay lock loop (MEDLL) [15, 33], implemented by NovAtel for GPS receivers. The MEDLL is a method for mitigating the effects due to multipath within the receiver tracking loops. The MEDLL does this by separating the incoming signal into its LOS and multipath components. Using the LOS component, the unbiased measurements of code and carrier phase can be made. Performance evaluation of narrow EML, wide EML (i.e., EML correlator with chip spacing of 1 chip) and
MEDLL in terms of multipath mitigation capability is presented in [34] for GPS C/A codes. The MEDLL shows better performance than narrow and wide EML DLLs, but it does not completely eliminate all multipath errors. Especially multipath signals with small relative delays are difficult to eliminate. However, the advantage of MEDLL is that it reduces the influence of multipath signals by estimating both LOS and multipath parameters. However, the performance analysis of MEDLL has not been much studied for SinBOC(1,1) modulated signals. Moreover, classical MEDLL is based on a maximum-likelihood search, which is computationally extensive. Here, we reduce the search space, by using a noncoherent MEDLL approach and by incorporating a phase search unit, based on statistical distributions of multipath phases. We will also include MEDLL in our performance comparison, as a benchmark algorithm.

Another feedforward technique is the slope differential (SD) approach, a recent multipath mitigation scheme based on the slope difference of the prompt correlator output (the correlation is computed between the received signal and the locally generated reference signal). This technique was first proposed in [35]. Advantage of the SD scheme is that it does not require a high speed digital signal processing for the narrow early-late spacing since it employs only the prompt correlator unlike standard DLLs [35]. In [35], it is assumed that the amplitude of the LOS signal is always larger than the amplitude of the multipath signal, which is a rather limiting assumption from the point of view of realistic multipath propagation scenarios. Therefore, a slightly modified approach, named as second-order differentiation (Diff2), is proposed in [31]. Unlike SD, Diff2 computes an adaptive threshold based on the estimated noise variance of the channel obtained from the feedforward loop in order to estimate the delay of the first arriving path. Because of this adaptive threshold, Diff2 is able to estimate the first path delay even in multipath profiles where the first path power is less than or equal to the consecutive path powers [31].

The matched filter (MF) concept is another popular feedforward technique which is extensively studied in [8, 36, 37]. MF is based on a threshold computation which is determined according to the channel condition provided by the feedforward loop. At first, the noise level is estimated and then a linear threshold is computed based on the noise variance plus some weight factor obtained from the feedforward loop. The choice of the weight factor is dependent on the modulation type. For SinBOC(1,1)-modulated signals, it has to be chosen such that the side lobe peaks of the envelope of the correlation function (CF) between the received signal and locally generated reference signal can be compensated. Therefore, the first peak of MF which is above the linear threshold corresponds to the estimated delay of the first path.

Another very promising code tracking algorithm is Teager-Kaiser (TK)-based delay estimation algorithm. The principle and the properties of the TK-based delay estimation algorithm are described in detail in [38, 39]. TK approach proved to give the best results for WCDMA scenarios in the presence of overlapping paths [9, 38]. According to [8], the performance of this algorithm is also very promising in closely spaced multipath scenarios for LOS delay estimation of SinBOC(1,1)-modulated signal in terms of accuracy and complexity. However, the best results performed with TK estimator were obtained with infinite receiver bandwidth. The presence of bandwidth limiting filter affects adversely the performance of TK estimator.

The purpose of our paper is two-fold: first, to propose an improved early-late-slope (IELS) technique, which increases the MTLL and decreases the RMSE compared with the narrow correlator, and secondly, to introduce two peak-tracking-based techniques with optimized parameters, that provide the best LOS estimation accuracy among the other studied algorithms. Additionally, the step-by-step implementation of a noncoherent MEDLL with incorporated phase estimation is given for both SinBOC and BPSK signals. In our improved ELS (IELS) technique, we propose two major updates to the basic ELS model. The first update is the adaptation of random spacing between the early and the late correlator pairs. This is mainly because of the fact that the random spacing between the early and late correlator pairs will generally provide more accuracy in order to draw slopes in the early and late sides of the correlation function as compared to fixed spacing, especially when fading channel model is concerned. The second update is the utilization of the feedforward information in order to determine the most appropriate peak on which the IELS technique should be applied. The peak tracking (PT) algorithms, as mentioned above, combine the advantages of feedback and feedforward techniques, in such a way that the delay estimation accuracy is increased, while still preserving a good mean time to lose lock (MTLL). We remark that the basic ideas of a peak tracking-like algorithm have been introduced by the authors in [40]. However, in [40], the PT was using only the second-order derivative estimates and its parameters were chosen empirically. Moreover, the algorithm presented in [40] is valid only for Galileo SinBOC(1,1)-modulated signals, while the work here is valid for both GPS and Galileo signals. We also explain here the choice of all the PT parameters and we introduce also the PT-with-TK algorithm.

Simulation results in multipath fading channels are included, in order to compare the performance of the proposed algorithm with the performance of various feedback and feedforward algorithms (some of them have been already mentioned in this introductory chapter, and the rest of them, which are less known or new, are explained in Sections 2 and 3). The procedure of PT algorithm is detailed in Section 4. The last two sections are dedicated to the simulation results and conclusions.

2. SIGNAL AND CHANNEL MODEL

In what follows, the continuous-time model is adopted for clarity purpose. The signal $s(t)$ transmitted from one satellite, with pseudorandom (PRN) code can be written as:

$$s(t) = \sqrt{E_b} p_{mod}(t) \otimes c(t),$$

where $E_b$ is bit energy, $p_{mod}(t)$ is the modulation waveform (e.g., BPSK for C/A GPS code or SinBOC(1,1) for L1 Galileo signals), and $c(t)$ is the spread navigation data.
(spreading is done with a pseudorandom code of chip interval \( T_c \) and spreading factor \( S_F \)):

\[
c(t) = \sum_{n=\infty}^{\infty} b_n \sum_{k=1}^{S_F} c_{k,n} \delta(t - nT_cS_F - kT_c). \tag{2}
\]

Above \( \delta(\cdot) \) is the Dirac unit pulse, \( b_n \) is the \( n \)th data bit (for pilot channels, \( b_n = 1, \forall n \)) and \( c_{k,n} \) is the \( k \)th chip (±1 valued) corresponding to the \( n \)th spread bit.

The modulation waveform for BPSK and SinBOC-modulated signals\(^1\) can be written as [41]:

\[
\rho_{mod}(t) = p_{TB}(t) \otimes \sum_{i=0}^{N_s-1} \delta(t - iT_B), \tag{3}
\]

where \( N_s \) is BOC modulation order: \( N_s = 1 \) for BPSK modulation\(^2\) and \( N_s = 2f_{sc}/f_s \) where \( f_{sc} \) is the subcarrier frequency and \( f_s \) is the carrier frequency for SINBoc modulation, \( T_B = T_c/N_s \) is the BOC interval, and \( p_{TB}(t) \) is the pulse shaping filter (e.g., for unlimited bandwidth, \( p_{TB}(t) \) is a rectangular pulse of width \( T_B \) and unit amplitude).

The received signal \( r(t) \), for multipath propagation and Doppler shift introduced by the channel is

\[
r(t) = \sqrt{E_b} \sum_{n=-\infty}^{\infty} b_n \sum_{k=1}^{S_F} c_{k,n} \sum_{l=1}^{L} a_{l,n} e^{j\phi_{l,n}} n\delta(t - nT_cS_F - kT_c - \tau_l) e^{-j2\pi f_{d}^{\hat{t}} t} + \eta(t), \tag{4}
\]

where \( L \) is the number of channel paths, \( a_{l,n} \) is the \( l \)th path amplitude during \( n \)th code epoch, \( \phi_{l,n} \) is the \( l \)th path phase during \( n \)th code epoch, and \( \tau_l \) is the \( l \)th path delay (typically assumed to be slowly varying or constant within the observation interval) and \( \eta(t) \) is a wideband additive noise, incorporating all sources of interferences over the channel. Assuming that the signal is sampled at \( N_c \) samples per chip (for BPSK) or per BOC interval (for BOC modulation), then the power spectral density of \( \eta(\cdot) \) can be written as \( \eta(t) \sim \eta_0/N_c \sqrt{N_s} \), where \( \eta_0 \) is the noise power in 1 kHz bandwidth (i.e., bandwidth corresponding to one code epoch).

At the receiver side, the incoming signal \( r(t) \) is correlated with a replica (reference signal) \( s_{ref}(t) \) of the modulated PRN code. The correlation output \( R(\cdot) \) can be written as:

\[
R(\tau, \hat{\tau}, \hat{f}_D) = E\left(r(\tau) \otimes s_{ref}(\tau)\right) = E\left(\int_{N_cT_c} r(t)s_{ref}(\tau - t)dt\right), \tag{5}
\]

where the correlation is performed over one spreading length of duration \( S_F T_c \) (this corresponds to 1 millisecond for GPS and Galileo), \( E(\cdot) \) is the expectation operator with respect to the random variables (e.g., PRN code, channel effects, etc.), and

\[
S_{ref}(\tau, \hat{\tau}, \hat{f}_D) = p_{mod}(t) \otimes c(t) \otimes \delta(t - \hat{\tau}) e^{-j2\pi f_{d}^{\hat{t}} t}, \tag{6}
\]

is the reference modulated PRN code with a code phase \( \hat{\tau} \) and Doppler shift \( \hat{f}_D \).

Since the main focus in this paper is the multipath tracking, we will assume in what follows that there is only a small residual Doppler error after the acquisition process \( \Delta f_D = f_D - \hat{f}_D \). Also, if we assume ideal codes and pilot channel-based estimation (or data removed before the correlation process), then \( E(c(t) \otimes c(t)) = \delta(t) \). With these assumptions, after several manipulations and by replacing (1) to (4) into (5) we get:

\[
\begin{align*}
R(\tau, \hat{\tau}, \hat{f}_D, n) &= \sqrt{E_b} \sum_{i=1}^{L} a_{i,n} e^{j\phi_{i,n}} R_{mod}(\tau - \tau_1 + \hat{\tau}) \mathcal{F}(\Delta f_D) + \tilde{\eta}(\tau, n), \\
\tilde{\eta}(\tau, n) &= \eta(t) S_F T_c e^{-j2\pi f_{d}^{\hat{t}} t} + \eta(t),
\end{align*}
\]

where \( R_{mod}(\tau) = p_{mod}(t) \otimes p_{mod}(t) \) is the autocorrelation of the modulation waveform (including BPSK or BOC modulation and pulse shaping and whose detailed expression can be found in [41]), and \( \mathcal{F}(\Delta f_D) = \sin((\pi\Delta f_D S_F T_c) e^{-j\pi n\Delta f_D S_F T_c} \) is a deterioration factor due to small residual Doppler errors (and it was obtained via integrating \( e^{-j2\pi f_{d}^{\hat{t}} t} \) over one code epoch). The filtered noise \( \tilde{\eta}(\tau, n) \) power spectral density (PSD) \( \tilde{N}_0 \) depends on the PSD of the modulation waveform, \( G_{mod}(f) \) via

\[
\tilde{N}_0 = N_0 G_{mod}(f) = N_0 G_{BPSK/BOC}(f) |P_{filter}(f)|^2, \tag{8}
\]

where the BPSK and BOCS PSD are given by\(^3\) [41]:

\[
G_{BPSK}(f) = T_c \sin^2(\pi f T_c), \tag{9}
\]

and, respectively:

\[
G_{BOC}(f) = \frac{1}{T_c} \left( \frac{\sin(\pi f T_c/S_F)}{\pi f S_F} \right)^2, \tag{10}
\]

In (8), \( P_{filter}(f) \) is the transfer function of the pulse shaping filter. For example, for infinite bandwidth, \( P_{filter}(f) = 1 \) over the bandwidth of interest.

In a practical receiver, in order to cope with noise, coherent and noncoherent integration of the correlation function might be used. The output after coherent integration over \( N_c \) code symbols is

\[
R(\tau, \hat{\tau}, \hat{f}_D) = \frac{1}{N_c} \sum_{n=1}^{N_c} R(\tau, \hat{\tau}, \hat{f}_D, n). \tag{11}
\]

\(^1\) The formulas for CosBOC modulations can be found in [41] and they are not reproduced here for sake of compactness.

\(^2\) BPSK can be seen as a particular case of BOC modulation, as shown in [41].

\(^3\) For simplicity, only the expression for sine BOC of even BOC modulation order is shown, such as for SinBOC(1,1); the other formulas can be found in [41].
The matched filter (MF) output after \( N_{nc} \) noncoherent integration blocks become:

\[
J_{MF}(\tau, \hat{\tau}, f_D) = \frac{1}{N_{nc}} \sum_{n=1}^{N_{nc}} R(\tau, \hat{\tau}, f_D) R^*(\tau, \hat{\tau}, f_D). \tag{12}
\]

Under the assumption of zero-mean additive noise \( \tilde{\eta}(\cdot) \), (12) becomes

\[
J_{MF}(\tau, \hat{\tau}, f_D) \approx E_b \sum_{l=1}^{L} \sum_{l_i=1}^{L_l} |a_l| \epsilon_l |\tilde{\eta}(\phi_{l}, \phi_{l_i})| R_{mod}(\tau - \Delta \hat{\tau}_l) \\
\times R_{mod}(\tau - \Delta \hat{\tau}_{l_i}) |F(\Delta f_D)|^2 + |\tilde{\eta}(\tau)|^2, \tag{13}
\]

where \( a_l = E(a_{l,n}) \) and \( \phi_{l,i} = E(\phi_{l,n}) \) are the average amplitude and phase values of path \( l \) over one code symbol interval, \( \Delta \hat{\tau}_l = \tau_l - \hat{\tau} \), and \( \tilde{\eta}(\tau) \) is the filtered and averaged noise (after coherent and noncoherent integration).

### 3. CODE TRACKING ALGORITHMS

#### 3.1. Early late slope technique (ELS)

The early-late-slope (ELS) multipath mitigation technique can be easily explained by having a look at the signal’s autocorrelation function (ACF) [32]. The general idea is to determine the slope at both sides of the ACF’s central peak. Once both slopes are known, they can be used to compute a pseudorange correction that can be applied to the measured pseudorange. This multipath mitigation technique has temporarily been used in some of NovAtel’s GPS receivers, where it has been called multipath elimination technology (MET).

The principle of forming pseudorange corrections is illustrated in Figure 1. Here, \( R(\tau) \) can be, for example, the coherent correlation function of (11) or the noncoherent output \( J_{MF}(\tau, \hat{\tau}, f_D) \) of (12). It can be noticed that the autocorrelation peak is distorted due to the influence of the multipath signal. The slope of the correlation function on the early side of the peak is \( a_1 \), and \( a_2 \) is the slope of the late side of the peak. The spacing between the early and late correlators is \( d \). Using the slope information the following error function can be derived to accurately estimate how much the correlators need to be moved so that they are centered on the peak [32]:

\[
T = \frac{y_1 - y_2 + (d/2)(a_1 + a_2)}{a_1 - a_2}, \tag{14}
\]

where \( T \) is the tracking error. This is actually the \( \tau \)-coordinate of the intersection of the two straight lines (i.e., the slopes \( a_1 \) and \( a_2 \)). \( T \) will equal zero when the two correlators are positioned equidistant on each side of the peak. When \( T \) is non-zero it can be used to feed back to the hardware to keep the early and late correlators centered on the peak.

#### 3.2. Improved early late slope technique (IELS)

In our improved ELS (IELS) technique, there are two major updates to the basic ELS model. The first update is the adaptation of random spacing between the early correlators (i.e., the spacing between \( E_1 \) and \( E_2 \)) and also between the two late correlators (i.e., the spacing between \( L_1 \) and \( L_2 \)). The reason is quite straightforward. The random spacing between the correlators will generally be more appropriate than fixed spacing to draw correct slopes in both the early and late sides of the correlation function, especially when fading channel model is concerned. The second update is the utilization of the feedforward information in order to determine the most appropriate peak on which the IELS technique should be applied. Unlike BPSK, BOC-modulated signal has side peaks with nonnegligible magnitudes. Therefore, there should be a fair way to get rid of these side peaks not being considered as the central peak. Similar with PT algorithm (that will be explained in Section 3.4), an IELS threshold is computed based on the estimated noise variance provided from the feedforward structure. The chosen IELS peak is the one which is above the threshold level as well as the closest to the previous estimation.

#### 3.3. Multipath estimating delay lock loop (MEDLL) implementation

Multipath estimating delay lock loop (MEDLL) is mainly designed to reduce both code and carrier multipath errors by estimating the parameters (i.e., amplitudes, delays, phases) of LOS plus multipath signals [15, 33]. The MEDLL of NovAtel uses several correlators (e.g., 6 to 10) per channel in order to determine accurately the shape of the multipath-corrupted correlation function. Then, a reference correlation function is used in a software module in order to determine the best combination of LOS and NLOS components (i.e., amplitudes, delays, phases and number of multipaths). An important aspect of the MEDLL is an accurate reference correlation function which could be constructed by averaging measured correlation functions over a significant amount of total averaging time [33].

The classical MEDLL approach involves the decomposition of correlation function into its direct and multipath components. The MEDLL estimates the amplitude \( a_l \), delay parameters (amplitudes, delays, phases), along with the most suitable number of multipaths. The description of the MEDLL algorithm is postponed until Section 3.4.
(τ) and phase (φ) of each multipath component using maximum likelihood criteria. Each estimated multipath correlation function is in turn subtracted from the measured correlation function. After the completion of this process, only the estimate of the direct path correlation function is left. Finally, a standard EML DLL is applied to the direct path component and an optimal estimate of the code tracking error is obtained [34, 42]. There are several implementations possible for MEDLL algorithm. Here, we chose a noncoherent MEDLL to make the implementation of the algorithm much faster and comparable with the complexity and the implementation of the other discussed algorithms. The steps used in our MEDLL implementation for BPSK and SinBOC(1,1) signals are summarized below.

(1) Find the maximum peak of $R(τ)$ (where $R(τ) = f_{\text{MF}}(τ, \hat{τ}, \bar{φ})$ from (12)) and its corresponding delay $\hat{τ}_1$, amplitude $\hat{a}_1$ and phase $\hat{φ}_1$. However, we have to mention that the phase information is lost due to noncoherent integration, thus we recover it by generating random (uniformly distributed in $[0, 2\pi]$) phases $\phi_k$ and by choosing that one corresponding to the minimum square error of the residual function $R^{(k)}(τ)$, $k$ being the iteration index (see next step). In practice, $N_{\text{random}} = 50$ phases proved to give accurate enough results (with no so significant computational burden).

(2) Subtract the contribution of the calculated peak, in order to have a new approximation of the correlation function $R^{(1)}(τ) = R(τ) - |\hat{a}_1 R_{\text{mod,ideal}}(t - \hat{τ}_1)e^{j\hat{φ}_1}|^2$. Here $R_{\text{mod,ideal}}(·)$ is the reference correlation function for a BPSK or SinBOC-modulated signal, in the absence of multipath (which can be, e.g., computed only once, according to ideal codes [41], and stored at the receiver). We remark that the choice of phase is not important during the first step (due to the squared absolute value), and, thus the phase estimation can be ignored during the first step. Find out the new peak of the residual function $R^{(1)}(·)$ and its corresponding delay $\hat{τ}_2$, the amplitude $\hat{a}_2$ and phase $\hat{φ}_2$. Subtract the contribution of the first two peaks from $R(τ)$ and find a new estimate of the first peak, as the peak of the residual $R^{(2)}(τ) = R(τ) - |\hat{a}_1 R_{\text{mod,ideal}}(t - \hat{τ}_1)e^{j\hat{φ}_1} + \hat{a}_2 R_{\text{mod,ideal}}(t - \hat{τ}_2)e^{j\hat{φ}_2}|^2$. The reestimated values of the delay and amplitude of first peak are rewritten in $\hat{τ}_1$ and $\hat{a}_1$, respectively. For more than two peaks, once the two first peaks are found, the search for the $l$th peak is based on the residual $R^{(l)}(τ) = R(τ) - |\sum_{m=1}^{l-1} \hat{a}_m R_{\text{mod,ideal}}(t - \hat{τ}_m)e^{j\hat{φ}_m}|^2$, $l \geq 3$. The procedure is continued iteratively until all desired peaks are estimated (see next steps).

(3) The previous step is repeated until a certain criterion of convergence is met, that is, when residual function is below a threshold (e.g., set from 0.4 to 0.5 here) or until the moment when introducing a new delay does not improve the performance (in the sense of root mean square error between the original correlation function and the estimated correlation function).

Ignoring completely the phase information and keeping only the amplitude estimates is also possible for MEDLL implementation, in order to decrease the computational burden. However, slight deterioration of performance is noticed, as seen in the illustrative MEDLL examples of Figure 2 (“Old MEDLL” method refers to the situation when the phase information is not taken into account, while the “New MEDLL” method refers to the situation when the phase information is searched for, in a random manner as explained above, with $N_{\text{random}} = 50$). Here, $N_r$ is the oversampling factor (a chip interval has 40 samples in this example). By increasing the number of random points $N_{\text{random}}$, the “New MEDLL” would approach the performance of coherent MEDLL.

### 3.4. Peak tracking algorithm

The motivation behind the development of the new PT algorithm was to find such an algorithm that fully utilizes the advantages of both feedforward and feedback techniques and improves the fine delay estimation. PT utilizes the adaptive threshold obtained from the feedforward loop in order to determine some competitive delays, that is, the delays which are competing as being the actual delay (i.e., the delay of the first arriving path). The adaptive threshold is based on the estimated noise variance of the absolute value of the correlation function between the received signal and the locally generated reference signal. At the same time, PT explores the advantage of feedback loop by calculating some weight factors based on the previous estimation in order to take decision about the actual delay. However, the utilization of feedback loop is always a challenge since there is a chance to propagate the delay error to subsequent estimations. Therefore, the delay error should remain in tolerable range (e.g., less than or equal to half of the width of main lobe of the envelope of the correlation function) so that the advantage from feedback loop could be properly utilized.

For a SinBOC(1,1)-modulated signal, the width of main lobe of the envelope of an ideal CF between the locally generated reference signal and the received code is about 0.7 chips as shown in Figure 3. Thus, when we deal with SinBOC(1,1) signals, we assume in what follows a maximum allowable delay error less than or equal to half of the width of the main lobe (i.e., 0.7/2 = 0.35 chips). This means that, if the delay error is higher (in absolute value) than 0.35 chips, the lock is considered to be lost and the acquisition and tracking processes should be restarted. For BPSK signals, the maximum delay error will be 1 chip (since the width of the main lobe is 2 chips).

The details of the PT algorithm are given in Section 4. Among the feedforward delay tracking algorithms, the matched filter (MF), the second-order differentiation (Diff2), and the Teager-Kaiser (TK) algorithms are described in the next subsections. Diff2 and TK algorithms represent also parts of the building blocks of PT algorithm with Diff2 technique, denoted herein as PT(Diff2) and of PT algorithm with TK technique (PT(TK)).

### 3.5. MF peak and MF technique

In this context, the term MF peak is defined as any local maximum point in the CF squared envelope that is greater than
or equal to a specific threshold (i.e., MF\text{Thresh}, as explained in Section 3.7.1). The MF peak (\text{MF}\text{Peak}) is actually the normalized amplitude of local maximum point of the CF squared envelope, which can be obtained using the following equation:

$$\text{MF}\text{Peak} = \forall x_i \{ (x_i \in \text{MF}) \land (x_i \geq x_i - 1) \land (x_i \geq x_i + 1) \land (x_i \geq \text{MF}\text{Thresh}) \};$$ \hspace{1cm} (15)

where ECF here stands for the squared envelope (squared absolute value) of the correlation function between the received signal and the locally generated reference signal: $\text{ECF} = \text{E}_{\text{MF}}(\tau, \hat{\tau}, \hat{f}_D)$ (see (12)), $\land$ is the intersection and operator, and $l_{\text{MF}}$ is the length of the set MF. Above, it was assumed that the samples of ECF are denoted via $x_i$. In what follows, we refer to this method as matched filter (MF) method, by analogy with [8].

### 3.6. Diff 2 peak and Diff 2 techniques

Second-order differentiation (Diff2) peak is defined as any local maximum of the second-order derivative of the ECF, that is greater than or equal to a specific threshold (i.e., Diff2\text{Thresh}). The Diff2 peak (Diff2\text{Peak}) is also normalized with respect to the maximum value of the second-order derivative of the ECF. We have:

$$\text{Diff2}\text{Peak} = \forall x_i \{ (x_i \in \text{Diff2}) \land (x_i \geq x_i - 1) \land (x_i \geq x_i + 1) \land (x_i \geq \text{Diff2}\text{Thresh}) \};$$ \hspace{1cm} (16)

where Diff2 is the second-order differentiation of $\text{E}_{\text{MF}}(\tau, \hat{\tau}, \hat{f}_D)$ from (12), $l_{\text{Diff2}}$ is the length of the set Diff2. Since the local maxima of ECF are also seen in the maxima of its second-order derivative, the Diff2 method includes the MF estimates, but it can also detect closely-spaced paths.

In Figure 4, MF peaks and Diff2 peaks are marked according to the definition described before. Figure 4 represents a plot for 2 path Nakagami-m fading channel model with $m = 0.5$ for SinBOC(1,1)-modulated signal. In this example case, decaying power delay profile (PDP) is used with a multipath separation of about 0.75 chips. carrier to noise ratio (CNR) is considerably high, that is, 100 dB-Hz, in order to emphasize the multipath channel effect. In Figure 4,
the peaks marked in the two subplots correspond to the channel delays and also to the MF and Diff 2 peaks (both MF and Diff 2 algorithms estimate correctly the channel paths in this example).

3.7. Noise thresholds for MF and Diff 2 algorithms

Noise threshold ($N_{\text{Thresh}}$) is obtained based on the noise level of the ECF. The noise level is estimated by taking the mean of out-of-peak values of the ECF. The out-of-peak values are all the ECF points which fall outside the rectangular window shown in Figure 5. The rectangular window is chosen such that it contains all side lobe peaks of the ECF, due to BOC modulation, as well as multipath effects (we have to assume a maximum delay spread of the channel, but this choice proved not to be so critical). Hence, the width of the rectangular window should not be less than 2 chips. In this example case, the width of the rectangular window was 2.4 chips.

3.7.1. MF threshold

MF Threshold ($M_{\text{FThresh}}$) is basically computed from the estimated noise threshold $N_{\text{Thresh}}$ and a weight factor $W_{MF}$ using the following equation:

$$M_{\text{FThresh}} = \max\{M_{\text{FPeak}} \cdot W_{MF} + N_{\text{Thresh}}\}, \quad (17)$$

where $M_{\text{FPeak}}$ and $N_{\text{Thresh}}$ were defined above and $W_{MF}$ is defined as follows (the exact choice within these intervals proved not to be critical):

$$0.1 \leq W_{MF} \leq 0.15 \quad \text{for BPSK},$$

$$0.3 \leq W_{MF} \leq 0.35 \quad \text{for SinBOC(1,1)}, \quad (18)$$

$W_{MF}$ was chosen optimized empirically (e.g., based on the levels of the side lobes of ECF of SinBOC(1,1)-modulated signal). Figure 6 represents an ideal ECF for SinBOC(1,1)-modulated signal where the side lobe peak have approximately the value 0.25. Therefore, $W_{MF}$ could be chosen according to (18) in order to avoid side lobe peaks being considered as the competitive peaks. Definition of competitive peaks is presented in Section 3–1.
3.7.2. Diff 2 threshold

Diff 2 threshold (Diff 2Thresh) is computed from the estimated noise threshold $N_{\text{Thresh}}$ and a weight factor $W_D$ via:

$$\text{Diff } 2\text{Thresh} = \max \{ \text{Diff } 2\text{Peak} \} W_D + N_{\text{Thresh}},$$

where Diff 2Peak and $N_{\text{Thresh}}$ were defined above and $W_D$ is defined as follows (based on the second-order derivative values of an ideal ECF):

$$0.22 \leq W_D \leq 0.3 \quad \text{for BPSK},$$
$$0.37 \leq W_D \leq 0.5 \quad \text{for SinBOC}(1, 1).$$

The second-order differentiation of ECF is very sensitive to noise which emphasizes the fact that the weight factor $W_D$ should be chosen higher than the weight factor $W_{\text{MF}}$ chosen for MFThresh. That is why the weight factor $W_D$ is slightly greater than $W_{\text{MF}}$.

3.8. Teager-kaiser (TK) peaks and TK technique

The nonlinear quadratic TK technique was first introduced for measuring the real physical energy of a system [43]. Since its introduction, it has widely been used in various speech processing and image processing applications and, more recently, it has also been applied in code division multiple access (CDMA) applications [38, 39, 44]. It was found that this nonlinear technique exhibits several attractive features such as simplicity, efficiency and ability to track instantaneously-varying spatial modulation patterns [45]. Teager-Kaiser operator is chosen in the context of this paper because it proved to give the best results in delay estimation process when used with other CDMA type of signals, as explained in [38, 39, 44].

Teager-Kaiser operator $\Psi_{\text{TK}}(\cdot)$ to a real or complex continuous signal $x(t)$ is given by [39]:

$$\Psi_{\text{TK}}(x(t)) = \dot{x}(t)x^*(t) - \frac{1}{2}\left[\ddot{x}(t)x^*(t) + x(t)\dddot{x}(t)\right].$$

For discrete signals $x(n)$, TK operator is defined as [39]:

$$\Psi_{\text{TK}}(x(n)) = x(n-1)x^*(n-1) - \frac{1}{2}\left[x(n-2)x^*(n) + x(n)x^*(n-2)\right].$$

TK peak is defined as any local maximum of the Teager-Kaiser operator applied to the ECF, that is greater than or equal to a specific threshold (i.e., TKThresh):

$$\text{TKPeak} = \forall x_i \{ (x_i \in \text{TK}) \land (x_i \geq x_i - 1) \land (x_i \geq \text{TKThresh}) \};$$

where $\text{TK} = \Psi_{\text{TK}}(J_{\text{MF}}(\tau, \hat{\tau}, \hat{f_D}))$ is the TK operator applied to ECF and $I_{\text{TK}}$ is the length of the set TK.

Above, TK Threshold (TKThresh) is computed similarly with MF and Diff 2 thresholds:

$$\text{TKThresh} = \max \{ \text{TKPeak} \} W_{\text{TK}} + N_{\text{Thresh}},$$

where $W_{\text{TK}}$ weight was obtained from the TK applied to an ideal ECF and by optimization based on simulations, that is,

$$0.25 \leq W_{\text{TK}} \leq 0.3 \quad \text{for BPSK},$$
$$0.3 \leq W_{\text{TK}} \leq 0.32 \quad \text{for SinBOC}(1, 1).$$

3.9. Competitive peaks concept

The competitive peaks are to be used in the proposed peak tracking algorithms. A competitive peak ($C_{\text{Peak}}$) can be obtained using the following equations:

$$C_{\text{Peak}} = \{ (\text{MFPeak}) \cup \{ \text{Diff } 2\text{Peak} \} \},$$
$$C_{\text{Peak}} = \{ (\text{MFPeak}) \cup \{ \text{TKPeak} \} \},$$

where the symbol $\cup$ is used as the union of two sets. This means that we combine the delay estimates given by MF and Diff 2, or by MF and TK, and form a set of “competitive” delays, from which the final estimate will be selected.

Since, for GNSS applications, the point of interest is to find the delay of the first arriving path (i.e., the LOS path), therefore, it would be enough to consider only the first few competitive peaks (in their order of arrival). Hence, we assume that:

$$\max (\hat{\ell}) = 5.$$
An example of peak tracking algorithm with TK technique is shown in Figure 8. Figure 4 represents a plot for 2 path Nakagami-m fading channel model with $m = 0.5$. Here, decaying power delay profile (PDP) is used with a multipath separation of about 0.75 chips. Carrier-to-noise ratio (CNR) is considerably high, that is, 100 dB-Hz, in order to emphasize the multipath channel effect. According to Figure 8, the first competitive peak corresponds to the delay of the first arriving path whereas the second competitive peak corresponds to the delay of the second arriving path which is about 0.75 chips apart from the first path.

4. DESCRIPTION OF PEAK TRACKING ALGORITHMS

The general architecture of PT algorithms (i.e., PT with Diff2 and PT with TK) is shown in Figure 9. In what follows, the step by step procedure of PT algorithms is presented.

4.1. Step 1: noise estimation

$N_{\text{Thresh}}$ is estimated according to Section 3.7, which is then used to determine $ACF_{\text{Thresh}}$, $\text{Diff2}_{\text{Thresh}}$ and $TK_{\text{Thresh}}$. These thresholds are then provided as input to the next step.

4.2. Step 2: competitive peak generation

Step 2(a): Look for MF peak(s) in ECF domain using (15).
Step 2(b): Look for Diff2 peak(s) in Diff2 domain using (16) (for PT(Diff2) method) or for TK peak(s) in TK domain using (23) (for PT(TK) method).
Step 2(c): Find competitive peak(s) using (26).

The competitive peak(s) obtained from step 2 are then fed into steps 3(a), 3(b) and 3(c) in order to assign weights in each substep for each particular competitive peak.

4.3. Step 3(a): weight based on peak height

Assign weight(s) $a_i$, $i = 1, \ldots, \hat{L}$, based on the competitive peak height(s) using the following equation:

$$a_i = \frac{[T_{\text{MF}}(\tau_i) + T_{\text{Diff2/TK}}(\tau_i)]}{2}, \quad i = 1, \ldots, \hat{L},$$

where $T_{\text{MF}}$ and $T_{\text{Diff2/TK}}$ are the MF and Diff2/TK correlation values, respectively, corresponding to a competitive peak:

$$T_{\text{MF}}(\tau_i) = \text{MF}(\tau_i), \quad i = 1, \ldots, \hat{L},$$

$$T_{\text{Diff2/TK}}(\tau_i) = \text{Diff2/TK}(\tau_i), \quad i = 1, \ldots, \hat{L}.$$  

4.4. Step 3(b): weight based on peak position

Assign weight(s) $b_i$, $i = 1, \ldots, \hat{L}$ based on peak positions in ECF distribution: the first peak is more probable than the second one, the second one is more probable than the third one and so on. This is based on the assumption that typical multipath channel has decreasing power-delay profile. In the simulation, the following weights were used based on peak positions:

$$[b_1, b_2, b_3, b_4, b_5] = [1.0, 0.8, 0.6, 0.4, 0.2],$$

where $b_i$, $i = 1, \ldots, \hat{L}$, denotes the weight factor for $i$th peak; that is, $b_1$ is the weight for 1st peak, $b_2$ is the weight for 2nd peak, and so on. It is logical to assign higher weights for the first few competitive peaks as compared to later peaks since the objective is to find the delay of the first path. Figure 10 represents the assignment of weights based on peak position.

4.5. Step 3(c): weight based on previous estimation

Assign weight(s) $c_i$, $i = 1, \ldots, \hat{L}$ based on the feedback from the previous estimation: the closer the competitive peak is
4.6. **Step 4: compute the decision variable**

The decision variable (regarding which of the competitive peaks will be declared as LOS peak), $d_i, i = 1, \ldots, \hat{L}$ is computed according to the following equation:

$$d_i = a_i \cdot b_i + c_i, \quad i = 1, \ldots, \hat{L},$$  \hspace{1cm} (34)

which means that the first two weights $a_i$ and $b_i$ have higher weight than the third weight. This decision variable was also optimized empirically, via simulations.

4.7. **Step 5: find estimated delay of the LOS path**

The LOS delay can then be obtained via:

$$\hat{\tau}_{LOS} = \arg \max_{i=1: \hat{L}} (d_i).$$  \hspace{1cm} (35)

Table 1 summarizes the weights assigned in the example path profile shown in Figure 7 for PT(Diff2). Similar results were obtained with PT(TK). In this example case, there are two competitive peaks meaning that we need to assign weights only for those two peaks. In assigning weights for $c_i, i = 1, \ldots, \hat{L}$, PT assumes that there is no initial error present from the previous estimation. In step 4, the algorithm simply computes the decision variable $d_i, i = 1, \ldots, \hat{L}$ using (34) for each competitive peak. And, finally, in step 5, PT(Diff2) algorithm selects the peak which has the maximum value for the decision variable $d_i$. In our case, it is $d_1$. Therefore, in this example, the first competitive peak corresponds to the true delay of the LOS path.
The comparison in terms of multipath error envelopes between the different considered algorithms is shown in Figure 11 for 2-path static channels with linear path amplitude values 1 and 0.8. The upper envelope is obtained for in-phase paths, and the lower envelope is obtained for –180 degrees phase shift between the paths. The receiver bandwidth here was set to 8 MHz and the noncoherent correlators were used. We recall the following notations used in the figures’ captions:

(i) nEML = narrow early-minus-late correlator (with 0.1 chip early-late spacing);
(ii) HRC = high resolution correlator (with 0.1 chip early-late spacing and 0.2 chip very early-very late chip spacing);
(iii) IELS = improved early late slope;
(iv) MEDLL = multipath estimating delay locked loop (implemented with phase information);
(v) Diff2 = second-order derivative-based algorithm;
(vi) TK = Teager-Kaiser-based algorithm;
(vii) MF = matched filter-based algorithm;
(viii) PT(Diff2) = peak tracking with second-order derivative stage;
(ix) PT(TK) = peak tracking with Teager-Kaiser stage.

5. MEE CURVES

Table 1: Assignment of weights for Figure 7.

<table>
<thead>
<tr>
<th></th>
<th>a1</th>
<th>b1</th>
<th>c1</th>
<th>d1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st competitive peak</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2nd competitive peak</td>
<td>0.85</td>
<td>0.8</td>
<td>0.25</td>
<td>0.93</td>
</tr>
</tbody>
</table>
Figure 13: RMSE and MTLL for SinBOC(1,1) and infinite bandwidth.

Figure 14: RMSE and MTLL for SinBOC(1,1) and 8 MHz bandwidth (Butterworth filter).
From Figure 11, the best MEE performance is obtained with the feedforward algorithms (TK, DiFF2, PT/TK, and PT/Diff2), followed by MEDLL approaches. We also remark that IELS algorithm outperforms the narrow correlator for closely spaced multipath (unlike the results reported in [23]). However, even after the improvements, ELS algorithm is outperformed by HRC method. However, MEEs are rather poor performance criteria since they assume only static 2-path channels and do not take into account the channel changes. More robust performance estimators are those based on root mean square error and mean-time-to-lose-lock values of the delay estimates, as shown in Section 6.

6. SIMULATION RESULTS

Simulation results have been carried in Nakagami-m closely spaced path scenarios for both SinBOC(1,1) and BPSK signals, and for both limited and unlimited bandwidths. The results are described in the next two subsections.

6.1. SinBOC(1,1) case

Simulation results were carried for both infinite bandwidth situation and for some severe bandwidth situation (i.e., 8 MHz double-sided bandwidth limitation). The received filter for finite bandwidth case was a 8-MHz 5th order butterworth filter with 0.1 dB passband ripples and 40 dB stopband attenuation. The received signal was sampled at \( N_r = 10 \) samples per BOC interval, \( N_r \) being the oversampling factor (i.e., here we have \( N_B/N_r = 20 \) samples per chip). The channel paths were assumed to have a Nakagami-m amplitude variation (\( m = 0.5 \)). The channel paths number was assumed to be randomly varying between 2 and 5 paths (uniform distribution), and the successive path separation was also a random variable, distributed between \( 1/(N_s N_B) \) and 0.35 chips (i.e., closely spaced path scenarios). The channel paths were assumed to obey a decaying power delay profile (PDP), with the decaying factor \( \mu = 0.2/N_s/N_{BOC} \). 8000 random points were used in the computation of RMSE and MTLL values. The RMSE was computed only on those delay estimation values which were (in absolute value) less or equal to 0.35 chips (i.e., half of the main lobe of the ECF). The MTLL was computed as the average number of delay estimates whose error (in absolute value) was less than 0.35 chips. The RMSE values are plotted in meters here, by using the relationship \( \text{RMSE}[\text{m}]= \text{RMSE}[\text{chips}]/c T_c, \) where \( c \) is the speed of light. It was also assumed that the initial delay estimate coming from the acquisition stage is accurate to 0 chip delay error, but that the LOS delay is randomly varying in time (with a uniform distribution of \( \pm 0.05 \) chips around the previous delay). Thus, the purpose of the delay tracking unit is to keep the lock, that is, to follow these delay variations. A coherent integration time of \( N_c = 20 \) ms was used, followed by noncoherent integration on \( N_{nc} = 4 \) blocks.

A snapshot of estimated LOS delay versus the true LOS delay for two of the considered algorithms is shown in Figure 12 for SinBOC(1,1) signal and 8 MHz receiver bandwidth.

Simulation results for SinBOC(1,1) case are shown in Figures 13 and 14, for infinite and finite bandwidth, respectively.

As seen in Figures 13 and 14, the best RMSE performance for both SinBOC(1,1) is achieved by MEDLL and PT(Diff2) approaches. However, MEDLL has quite poor MTLL performance (it tends to lose lock faster), thus the best tradeoff between RMSE and MTLL is achieved, by PT(Diff2) and HRC algorithms. PT(TK) has poorer performance than PT(Diff2) algorithm.

6.2. BPSK case

For BPSK signals, similar parameters as for SinBOC(1,1) signal were used in the simulations. The only differences were a higher oversampling factor \( N_r = 16 \), for an increased accuracy (since the number of samples per chip is \( N_r \) for BPSK case, while, for SinBOC(1,1), it was \( 2N_c \)) and the successive path spacing of 1 chip (since the width of the ECF is 2 chips for BPSK). Also, the RMSE values are computed over the delay estimates which are at most half of the main lobe width apart from the true LOS estimate, which corresponds to 1 chip for BPSK case. The statistics were also done for \( N_{\text{rand}} = 8000 \) random realizations (each random realization has a length of \( 8 N_c N_{nc} \) milliseconds), thus the best MTLL that can be achieved by our simulations is \( N_{\text{rand}} N_c N_{nc} \) (here, this corresponds to 640 s).

A snapshot of estimated LOS delay versus the true LOS delay for two of the considered algorithms is shown in Figure 15 for BPSK signal and 8 MHz receiver bandwidth. RMSE and MTLL values for BPSK case are shown in Figures 16 and 17, for infinite and finite bandwidth case, respectively.
Similar conclusions as for SinBOC (1,1) signals are also drawn from Figures 16 and 17, but this time PT(Diff2) outperforms MEDLL also in RMSE values. The differences between infinite bandwidth and limited bandwidth situations are quite small, which means that the algorithms work similarly well in both situations.

7. CONCLUSIONS

In this paper, novel peak tracking delay estimation algorithms have been proposed for Galileo and GPS signals. PT algorithms combine the feedback and feedforward delay estimators in order to achieve reduced RMSE delay error and
promising mean time to lose lock. Among all the considered algorithms, the best tradeoff between RMSE and MTLL was achieved via PT(Diff 2) algorithm, which decreased considerably the delay estimation error at moderate to high CNRs, while still preserving a better MTLL compared with other feedforward tracking approaches. We also presented an improved early-late-slope technique which outperforms the narrow correlator especially in the presence of short delay multipath, where the classical ELS was showing worse results. We have as well presented a reduced complexity implementation of a noncoherent MEDLL, where the phase information was searched for via statistical assumptions. Extensive simulation results in both limited and unlimited receiver bandwidth have been presented, including 9 feedback and feedforward delay tracking algorithms. We have also shown that at small CNRs (e.g., up to 25 dB-Hz for a coherent integration time of 20 milliseconds), narrow correlator is still the best choice among the considered algorithms. However, its performance is still far from accurate. Better results can be achieved via increasing CNRs (or, alternatively) increasing $N_s$, with a combined feedback-feedforward approach.

ACKNOWLEDGMENTS

This work was carried out in the project “Advanced Techniques for Personal Navigation (ATENA)” funded by the Finnish Funding Agency for Technology and Innovation (Tekes). This work has also been partially supported by the Academy of Finland.

REFERENCES


Publication P3

Copyright ©2009 IEEE. Reprinted with permission.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the Tampere University of Technology's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org. By choosing to view this material, you agree to all provisions of the copyright laws protecting it.
Multipath mitigation performance of multi-correlator based code tracking algorithms in closed and open loop model

Mohammad Zahidul H. Bhuiyan, Xuan Hu, Elena Simona Lohan, and Markku Renfors
Department of Communications Engineering, Tampere University of Technology
P.O.Box 553, FIN-33101, Finland;
{mohammad.bhuiyan; xuan.hu; elena-simona.lohan; markku.renfors}@tut.fi

Abstract—Multipath is one of the dominant error sources in satellite-based positioning. It is a well known fact that the conventional code tracking loop suffers from performance degradation due to the presence of multipath. In past years, several advanced signal processing techniques have been devised to mitigate the multipath induced errors. One particular class of these signal processing techniques is the multi-correlator based feed-forward approach. The multipath performance of these feed-forward techniques has been extensively studied in open loop configuration by the authors in [1], [9]. The goal of this paper is to analyze the performance of these multi-correlator based feed-forward techniques in closed loop configuration, i.e., in the presence of an NCO and a loop filter, in multipath channels. Additionally, the delay tracking methods previously studied for Binary Phase Shift Keying (BPSK) and Sine Binary Offset Carrier (SinBOC) signals are also studied here with the newly proposed Multiplexed BOC modulation.

I. INTRODUCTION

Multipath signal propagation remains a dominant source of error in Global Navigation Satellite System (GNSS) positioning. In the past years, lots of work have been achieved to improve the multipath rejection performance of the receivers. In order to reduce multipath error due to the presence of reflected signals, several approaches have been used. Among them, the use of special multipath limiting antennas (i.e., choke ring or multi-beam antennas), the post-processing techniques to reduce carrier multipath, the carrier smoothing to reduce code multipath, and the code tracking algorithms based on receiver internal correlation technique (i.e., narrow Early-Minus-Late [2] or High Resolution Correlator [10]) are the most prominent approaches. In this paper, our focus is limited to the correlation based multipath mitigation techniques. The most known code tracking algorithm is the traditional Early-Minus-Late (EML), which is composed of 1 chip spacing between early and late correlator pair. The traditional EML has limited multipath mitigation capability, and therefore, several enhanced EML-based techniques have been introduced, specially to mitigate closely spaced multipath. One class of these enhanced EML techniques is based on the idea of narrowing the spacing between the early and late correlators, i.e., narrow EML (nEML), provided that sufficient front-end bandwidth is assured [2]. Another enhanced version of this type of structure is the High Resolution Correlator (HRC), that uses an increased number of correlators for better coping with medium-to-long delay multipath [5], [10]. Alternatively, several feed-forward techniques have been introduced in the literature in past few years [1], [9]. While improving the delay estimation accuracy, these techniques require a larger number of correlators than the traditional Delay Locked Loop (DLL), and they are sensitive to the noise-dependent threshold choice. Among the feed-forward techniques, two most competitive ones, previously proposed by the authors, are selected herein for performance comparison. These are Peak Tracking, based on $2^{nd}$ order Differentiation (PT(Diff2)), and Teager-Kaiser (TK), the details of which can be obtained in [1]. The performance of these multi-correlator based techniques [1], [4], [9] have been extensively studied in fading multipath environment with open loop model, i.e., in the absence of a Numerically Controlled Oscillator (NCO) and a loop filter. In this paper, we present the performance of these multi-correlator based tracking algorithms in closed loop model, i.e., in the presence of an NCO and a loop filter, in multipath channels. The main novelty of this paper comes from building the link between feed-forward and feedback approaches, and showing that the feed-forward approaches previously proposed have a great potential also when used in closed loop configuration. Additionally, the delay tracking methods previously studied for Binary Phase Shift Keying (BPSK) and Sine Binary Offset Carrier (SinBOC) modulated signals are also studied here with the newly proposed Multiplexed Binary Offset Carrier (MBOC) modulation.

This paper is organized as follows. The overview of MBOC modulation is presented in Section III followed by a description about the implemented closed loop model in Section IV. The multipath tracking performance via semi-analytical Multipath Error Envelopes (MEE) in closed loop model are presented in Section V. The simulation results for both closed and open loop models are shown in Sections VI and VII respectively. Finally, section VIII draws some general conclusions based on the obtained results.

II. OVERVIEW OF MBOC MODULATION

The GIOVE-B, the second Galileo satellite, launched on April 27, 2008 started transmitting the Galileo L1 signal using a specific optimized wave-form, MBOC, that will be interoperable with the L1C signal to be used in future Block III.
GPS satellites, in accordance with the July 2007 agreement between the European Union and the United States [11]. The MBOC modulation enables receivers to obtain significantly better multipath mitigation performance than BPSK and SinBOC(1,1) modulations. The multipath improvement of MBOC modulation over SinBOC(1,1) is shown in [11] with the transmitted GIOVE-B signal.

The MBOC(6,1/11) power spectral density is a mixture of SinBOC(1,1) and SinBOC(6,1) spectra. The MBOC(6,1/11) spectrum can be generated by a number of different time waveforms that allows flexibility in implementation. The Time-Multiplexed BOC (TMBOC) implementation interfaces SinBOC(6,1) and SinBOC(1,1) spreading symbols in a regular pattern, whereas Composite BOC (CBOC) uses multilevel spreading symbols formed from the weighted sum (or difference) of SinBOC(1,1) and SinBOC(6,1) spreading symbols, interpolated to form a constant modulus composite signal [3]. Following the BOC model and derivations of [7], the composite CBOC signal which is used here can be written as:

\[
s_{CBOC}(t) = w_1 s_{SinBOC(1,1), held}(t) \pm w_2 s_{SinBOC(6,1)}(t)
\]

Above, when the 2 right-hand terms are added, additive CBOC or CBOC(‘+’) is formed; when the 2 terms are subtracted, we have the inverse CBOC or CBOC(‘-’) implementation. Alternatively, CBOC(‘+/-’) implementation can be used, when odd chips are CBOC(‘+’) modulated and even chips are CBOC(‘-’) modulated [3]. In eqn. 1, \( N_{B_1} = 2 \) is the BOC modulation order for SinBOC(1,1) signal, \( N_{B_2} = 12 \) is the BOC modulation order for SinBOC(6,1) signal, the term \( s_{SinBOC(1,1), held} \) represents that SinBOC(1,1) signal is passed through a hold clock in order to match the higher rate of SinBOC(6,1); and \( w_1 \) and \( w_2 \) are amplitude weighting factors such that \( w_1 = \sqrt{10/11} = 0.9535 \) and \( w_2 = \sqrt{1/11} = 0.3015 \), and \( c(t) \) is the pseudorandom code as defined in eqn. 2.

\[
c(t) = \sqrt{E_b} \sum_{n=-\infty}^{\infty} b_n \sum_{m=1}^{S_F} c_{m,n} p_{T_{\delta_2}}(t - nT_c S_F - mT_c), \quad (2)
\]

where \( b_n \) is the \( n \)-th code symbol, \( E_b \) is the code symbol energy, \( S_F \) is the spreading number or number of chips per code symbol \( (S_F = 1023) \), \( c_{m,n} \) is the \( m \)-th chip corresponding to the \( n \)-th symbol, \( T_c \) is the chip rate, and \( p_{T_{\delta_2}}(\cdot) \) is a rectangular pulse of support \( T_c/N_{B_2} \) and unit amplitude. In eqn. 1, the first term comes from the SinBOC(1,1) modulated code (held at rate \( 12/T_c \) in order to match the rate of the second term), and the second term comes from a SinBOC(6,1) modulated code.

In TMBOC implementation, the whole signal is divided into blocks of \( N \) code symbols and \( M < N \) of \( N \) code symbols are SinBOC(1,1) modulated, while \( N - M \) code symbols are SinBOC(6,1) modulated. Using similar derivations as in [7], we can obtain the formula for TMBOC waveform. An equivalent unified model of CBOC and TMBOC modulations can be derived using the facts that \( M, N << \infty \) and that, since \( w_1, w_2 \) are amplitude coefficients and \( M, N - M \) define the power division between SinBOC(1,1) and SinBOC(6,1), we may set the following relationship between \( w_1, w_2 \) and \( M, N \):

\[
w_1 = \sqrt{M/N} \quad \text{and} \quad w_2 = \sqrt{N/M}. \quad (3)
\]

Therefore, in accordance with [8], the unified model can be written as:

\[
s_{TMBOC}(t) = w_1 c_1(t) \oplus s_1(t) \oplus p_{T_{\delta_2}}(t) + w_2 c_2(t) \oplus s_2(t) \oplus p_{T_{\delta_2}}(t) \quad (3)
\]

where \( \oplus \) is the Dirac pulse, \( \oplus \) is the convolution operator, \( c_1(t) \) is the code signal without pulse shaping:

\[
c_1(t) = \sqrt{E_b} \sum_{n=-\infty}^{\infty} b_n \sum_{m=1}^{S_F} c_{m,n} \delta(t - nT_c S_F - mT_c), \quad (4)
\]

and \( s_1(t), s_2(t) \) are SinBOC-modulated parts (with associated hold block when needed), given by:

\[
s_1(t) = \sum_{i=0}^{N_{B_1}-1} \sum_{k=0}^{N_{B_2}-1} (-1)^i \delta(t - i\frac{T_c}{N_{B_1}} - k\frac{T_c}{N_{B_2}}), \quad (5)
\]

and, respectively:

\[
s_2(t) = \sum_{i=0}^{N_{B_2}-1} (-1)^i \delta(t - i\frac{T_c}{N_{B_2}}) \quad (6)
\]

In our simulations, we use MBOC modulation with \( M = 10 \) and \( N = 11 \), since MBOC combines SinBOC(6,1) and SinBOC(1,1) spreading symbols with a 1/10 average power ratio. In [8], the equivalence between CBOC and TMBOC implementations was discussed.

III. MULTI-CORRELATOR STRUCTURE AND THEIR CLOSED LOOP IMPLEMENTATION

Compared with the conventional EML tracking loop, where only 3 correlators are used (i.e. Early, Prompt and Late), here, in the multi-correlator based structure, we generate a bank of correlators (in this implementation, we use 81 correlators with 0.05 chips spacing between successive correlators) as presented in Fig. 1. This large number of correlators is needed in order to include the feed-forward techniques in the comparison, because feed-forward techniques make use of these correlators for estimating the channel properties while taking decision about the code delay [1]. Some of these correlators can be kept inactive or unused, for example when EML and HRC tracking loops are used. After the necessary front-end processing, and after the carrier has wiped-off, the received post-processed signal is passed through a bank of correlators. As shown in Figure 1 the NCO and PRN generator block
produces a bank of early and late versions of replica codes based on the delay of the Line-Of-Sight (LOS) signal \( \hat{\tau} \), the correlator spacing \( \Delta \), and the number of correlators \( N \). In case of EML tracking loop, the corresponding early-late spacing is equal to \( 2\Delta \). The received signal is correlated with each replica in the correlator bank, and the output of the correlator bank is a vector of samples in the correlation envelope. Therefore, we obtain the correlation values for the range of \( \pm N\Delta \) chips from the prompt correlator, where \( N \) is the number of correlators and \( \Delta \) is the correlator spacing between successive correlators. The various code tracking algorithms (named as Discriminator in Fig. 1) utilize the correlation values as input, and generate the estimated LOS delay as output, which is then smoothed by a loop filter. In accordance with [6], the implemented code loop filter is a 1st order filter, whose function can be written as:

\[
\hat{\tau}(k+1) = \hat{\tau}(k) + \gamma d(k)
\]

where \( \gamma \) is calculated based on loop filter bandwidth, \( B_n \). A DLL loop bandwidth of 1 Hz is used, assuming that carrier aiding is always available [10].

IV. MULTIPATH ERROR ENVELOPES IN CLOSED LOOP MODEL

The multipath tracking performance is studied first via Multipath Error Envelopes (MEE) for static 2 path channel in closed loop model. In MEE analysis, the noise free environment is always considered, and the focus is on the multipath induced delay errors. Typically, a two path model is used to generate the MEE curves as in [3], [10]. The multipath amplitude is 3 dB less than the LOS path amplitude. The MEE curves are obtained for two extreme phase variations (i.e., 0 and 180 degrees) of multipath signal with respect to LOS component. The multipath delays are varying from 0 to 1.5 chips with a step size of 0.05 chips. The MEE simulations were carried out for BPSK, SinBOC(1,1) and MBOC modulated signals. A more representative curve known as Running Average Error (RAE) is presented here in accordance with [3]. RAE is computed from the area enclosed within the multipath error and averaged over the range of the multipath delays from zero to the plotted delay values.

Four different code tracking algorithms are analyzed for performance comparison. Among them, nEML and HRC are conventional delay tracking algorithms. The narrow correlator or nEML is the first approach to reduce the influence of code multipath that uses a chip spacing of 0.05 or 0.1 chips (less than 1 chip) depending on the available front-end bandwidth [2]. HRC uses an increased number of correlators (i.e. 5 correlators) for better coping with medium-to-long delay multipath [5], [10].

The other two algorithms are based on feed-forward technique, known as Peak Tracking based on 2nd order Differentiation (PT(Diff2)) and Teager Kaiser (TK). Both of these algorithms utilize the adaptive threshold computed from the estimated noise variance of the channel in order to decide on the correct code delay [1]. These feed-forward algorithms first generate competitive peaks which are above the computed adaptive threshold. In PT(Diff2), the competitive peaks are then multiplied by some optimized weighting factors, which are assigned based on the peak power, the peak position and the delay difference of the peak from the previous delay estimate. Finally, PT(Diff2) selects the peak which has the maximum weight as being the best LOS candidate. In contrast to PT(Diff2), TK only considers the delay difference of the peak from the previous delay estimate while taking decision on the LOS delay [1].

Running average error results for BPSK, SinBOC(1,1) and MBOC modulated signals are shown in Figs 2, 3 and 4, respectively. For all three different types of modulation, PT(Diff2) and TK can mitigate the multipath effects completely for...
all possible multipath delays in the absence of noise. The simulation results in the presence of noise are to be shown in Section V. Their curves are overlapping in Figs 2-4. Among the other two algorithms, HRC outperforms nEML in all three cases as observed from the RAE curves. The multipath improvement of MBOC signal over SinBOC(1,1) signal can be noticed for nEML and HRC. This is mainly because of the fact that MBOC modulated signal has slightly steeper auto-correlation main lobe, as opposed to the auto-correlation main lobe of SinBOC(1,1) signal.

The statistics were computed for $N_{\text{rand}} = 2000$ random realizations for each particular Carrier-to-Noise-Ratio (CNR). The Root-Mean-Square-Errors (RMSE) are plotted in metres, by using the relationship $\text{RMSE}_m = \text{RMSE}_{\text{chips}} c T_c$; where $c$ is the speed of light, $T_c$ is the chip duration, and $\text{RMSE}_{\text{chips}}$ is the RMSE in chips.

RMSE vs. CNR plots for all three modulations are shown in Fig. 5. The feed-forward algorithms showed superior performance for all three modulation types in moderate-to-high CNR conditions (i.e., from around 30 dB-Hz onward). Among the other two algorithms, HRC shows better multipath mitigation capability than nEML only in good CNR conditions, since HRC is more sensitive to noise.

VI. SIMULATION RESULTS IN OPEN LOOP MODEL

Simulations were carried out in the same simulation environment as mentioned in Table I but in open loop configuration, i.e., without any NCO and loop filter, having one shot estimates. RMSE vs. CNR plots for open loop model are shown in Fig. 6. As in closed loop implementation, here also, the feed-forward algorithms perform the best for all three modulation types in moderate-to-high CNR conditions. Similar conclusion (as like closed loop implementation) can be drawn for HRC and nEML.
Fig. 5. RMSE vs. CNR plots for in closed loop model

Fig. 6. RMSE vs. CNR plots for in open loop model
VII. CONCLUSIONS

In this paper, the feed-forward techniques, previously proposed by the authors in [1], are implemented in closed loop configuration. The multipath performance of these algorithms along with the conventional DLLs for three different modulation types, including the newly proposed MBOC modulation, are presented in terms of running average error and RMSE. The results are then compared with the open loop counterpart for observation. It is shown that the feed-forward algorithms provide much better multipath mitigation performance than the traditional DLLs in moderate-to-high CNRs for all three modulated signals in both closed and open loop models. It is also shown that the results in closed loop and open loop configurations are similar (closed loop results slightly better), which points out the fact that the feed-forward algorithm analysis can be quite conveniently (i.e., faster) asserted in open loop configuration. The multipath improvement of MBOC signal over BPSK and SinBOC(1,1) signals is also evident from the simulation results (i.e. smaller RMSE values for MBOC signal as compared to other two signals).

ACKNOWLEDGEMENT

This work was carried out in the project “Future GNSS Applications and Techniques (FUGAT)” funded by the Finnish Funding Agency for Technology and Innovation (Tekes). This work has also been supported by the Academy of Finland, which is gratefully acknowledged.

REFERENCES


Copyright ©2009 Nordic Institute of Navigation. Reprinted with permission.

DOI= http://dx.doi.org/10.1109/ISCAS.2010.5537801

Copyright ©2010 IEEE. Reprinted with permission.
A Slope-Based Multipath Estimation Technique for Mitigating Short-Delay Multipath in GNSS Receivers

Mohammad Zahidul H. Bhuiyan, Elena Simona Lohan and Markku Renfors
Department of Communications Engineering, Tampere University of Technology
P.O. Box 553, FIN-33101, Finland
Email: {mohammad.bhuiyan, elena-simona.lohan, markku.renfors}@tut.fi

Abstract—The everlasting public interest on location and positioning services has originated a demand for a high performance Global Navigation Satellite System (GNSS), such as the Global Positioning System or the future European satellite navigation system, Galileo. The performance of GNSS is subject to several errors, such as ionosphere delay, troposphere delay, receiver noise and multipath. Among all these errors, multipath is the main limiting factor in precision-oriented GNSS applications. In order to mitigate the multipath influence on navigation receivers, the multipath problem has been approached from several directions, including the development of novel signal processing techniques. Many of these techniques rely on modifying the tracking loop discriminator in order to make it resistant to multipath. These techniques have proved very efficient against multipath having a medium or large delay with respect to the Line-Of-Sight (LOS) signal. In general, the multipath errors are largely reduced for multipath delays greater than around 0.1 chips (which is about 29.3 meters for Galileo E1 Open Service (OS) signal). Theoretically, this constitutes a remarkable improvement as compared to simpler techniques such as narrow Early-Minus-Late (nEML) tracking loop. However, in practice, most of the multipath signals enter the receiver with short-delay with respect to LOS signal, making most of these mitigation techniques partially ineffective. In this paper, we propose a new multipath estimation technique, namely the Slope-Based Multipath Estimation (SBME), which is capable of mitigating the short-delay multipath (i.e., multipath delays less than 0.35 chips) quite well compared with other state-of-the-art mitigation techniques, such as the nEML and the High Resolution Correlators (HRC). The proposed SBME first derives a multipath estimation equation by utilizing the correlation shape of the ideal normalized correlation function of a Binary Phase Shift Keying (BPSK)- or Multiplexed Binary Offset Carrier (MBOC)-modulated signal, which is then used to compensate for the multipath bias of a nEML tracking loop. It is worth to mention here that the SBME requires an additional correlator at the late side of the correlation function, and it is used in-conjunction with a nEML tracking loop. The multipath performance of the above-mentioned mitigation techniques is presented for Galileo E1 OS and GPS L1 C/A signals from theoretical as well as simulation perspective.

I. INTRODUCTION

Multipath propagation remains a dominant source of error in Global Navigation Satellite System (GNSS) positioning. Several approaches have been used in order to reduce the multipath error. Among them, the use of special multipath limiting antennas (i.e., choke ring or multi-beam antennas), the post-processing techniques to reduce carrier multipath, the carrier smoothing to reduce code multipath, and the code tracking algorithms based on receiver internal correlation technique are the most prominent approaches [1], [2]. In this paper, our focus is limited to the correlation based multipath mitigation techniques, since they are the most widely used in commercial GNSS receivers. The most known correlation-based code tracking algorithm is the traditional narrow Early-Minus-Late (nEML), which is based on the idea of narrowing the spacing between the early and late correlators, provided that sufficient front-end bandwidth is assured [1]. Another enhanced version of this type of structure is the High Resolution Correlator (HRC), which uses a higher number of correlators (i.e., 5 complex correlators) for better coping with medium-to-long delay multipath in good Carrier to Noise density ratio (C/N0) [2]. Another category of techniques relies on an estimation of the parameters (delay, amplitude and phase) of the Line-Of-Sight (LOS) signal along with all other multipath components. Among these techniques, Multipath Estimating Delay Lock Loop (MEDLL) [3], and Reduced Search Space Maximum Likelihood (RSSML) estimator [4] achieve a significant performance improvement against multipath at the expense of a higher complexity. Most of these techniques suffer from the fact that they are partially ineffective against short-delay multipath. This is a strong limitation since most of the multipath signals tend to be close-in, short-delay type in practice [5].

The main motivation for this research comes from the well-known property that the signal amplitude of a GNSS signal is highly correlated with the multipath error in the code-phase measurement, as mentioned in [5]. This property is even more effective for short-delay multipath, since the sensitivity of the signal amplitude to multipath is maximized for short-delays.

This paper introduces a novel Slope-Based Multipath Estimation (SBME) technique, which attempts to compensate the multipath error contribution of a nEML tracking loop by
utilizing the slope information of an ideal normalized correlation function. More specifically, the tracking error affecting an nEML tracking loop is estimated in an independent module on the basis of different signal amplitude measurements (i.e., on the basis of distortion on the slope of the received correlation function with respect to ideal correlation function). The estimated multipath error is then subtracted from the code-phase measurement which yields a substantial reduction of the error, especially for short-delay multipath.

The theoretical Multipath Error Envelope (MEE) and simulation based analyses are presented here for short-delay multipath profiles in order to compare the performance of the proposed mitigation technique with two other state-of-the-art DLLs. The performance of these tracking algorithms is analyzed for the newly proposed Multiplexed Binary Offset Carrier (MBOC) modulation for Galileo Open Service (OS) and modernized GPS, along with the existing Binary Phase Shift Keying (BPSK) modulation used in GPS L1 C/A signal. Among the variants of MBOC modulation, Time-Multiplexed BOC (TMBOC) modulation is chosen for modernized GPS L1C signal, whereas Composite BOC (CBOC) modulation is chosen for Galileo E1 signal. In the analysis, we will use a CBOC(-) implementation as an example case, since all MBOC variants have only slight variations of the correlation lobes, and since CBOC(-) is the modulation of choice for Galileo pilot channel (E1C), which is the most likely to be used for high accuracy tracking. Further details on CBOC(-) and other MBOC implementations can be found in [6].

II. SLOPE BASED MULTIPATH ESTIMATION TECHNIQUE

The signal amplitude measured in a GNSS receiver is the reading of the in-prompt correlation value. The signal amplitude can also be computed with respect to the correlation value of any late side correlators \( R_{s+1} \) (\( l \) being any integer). For example, let \( R_{s+1} \) be the correlation value at \( +l \Delta EL \) chips delay from the prompt correlator, where \( \Delta EL \) is the early-late correlator spacing.

Let \( A_{R_{s+1}} \) be the amplitude of the signal with respect to the correlation value \( R_{s+1} \), which is \( +l \Delta EL \) chips delayed from the prompt correlator with a correlation value of \( A_{R_0} \). All the correlation values \( A_{R_{s+1}} \) and \( A_{R_0} \) will represent an identical signal amplitude in case no multipath is present, i.e., when the correlation peak is not distorted by any multipath.

We can express the late slope of the normalized correlation function of any modulation in terms of \( R_{s+1}, A_{R_{s+1}}, l \) and \( \Delta EL \), as follows:

\[
\frac{R_{s+1} - A_{R_{s+1}}}{\frac{\Delta EL}{2}} = m_{MOD\_TYPE}
\]

(1)

where \( m_{MOD\_TYPE} \) is the late slope of the normalized correlation function of any modulation \( MOD\_TYPE \) (i.e., either BPSK or CBOC(-) or any other type of modulation). For example, in case of ideal normalized correlation function and infinite receiver bandwidth, \( m_{BPSK} = -1 \) and \( m_{CBOC(-)} = -5.3847 \). In case of CBOC(-), this value corresponds to the first late slope with respect to the main peak. This also implies the fact that \( R_{s+1} \) should also lie on the same slope of the received correlation function (i.e., can be anywhere within 0.083 chips from the prompt correlator if we decide to use \( m_{CBOC(-)} = -5.3847 \)).

Therefore, we can determine the value of the signal amplitude \( A_{R_{s+1}} \) by the following equation:

\[
A_{R_{s+1}} = R_{s+1} - m_{MOD\_TYPE} \frac{l \Delta EL}{2}
\]

(2)

In case when more than one path is present, all these estimates for the signal amplitude (i.e., the prompt correlation value) are corrupted, and the multipath errors from different estimators are not the same. Empirically, it is found that a good estimate of the multipath error of a narrow Early-Minus-Late (nEML) tracking loop can be obtained by using an appropriate function of the signal amplitude measured from the prompt correlation \( (A_{R_{s+1}}) \), and from the late correlation at a delay of \( \Delta EL \) from the prompt correlator \( (A_{R_0}) \), i.e., by utilizing Eqn. (2) with \( l = +2 \).

In Fig. 1, a plot is shown for the normalized signal amplitudes computed from \( R_0 \) and \( R_{s+1} \) as a function of multipath delay for a Signal-to-Multipath-Amplitude Ratio (SMAR) of 6 dB for a BPSK modulation. It is sufficient to focus on the in-phase and the out-of-phase combination of multipath, since they contribute the largest multipath error.

From Fig. 1, it can be seen that the differences between the two normalized amplitudes is moderately steady for multipath delays from 0.1 to 1 chip. This inspires the fact that we can estimate the multipath error caused by nEML tracking loop by proper scaling of the difference between the two normalized amplitudes (i.e., \( A_{R_{s+1}} \) and \( A_{R_0} \)).

Fig. 1. Normalized amplitudes (i.e., \( A_{R_{s+1}} \) and \( A_{R_0} \)) for 2 path signal with path amplitude [0 -6] dB for BPSK modulation

Fig. 2 presents this difference between the two normalized amplitudes, scaled by 0.46, where 0.46 is the optimized coefficient. The scaling factor 0.46 is a coefficient computed in least square sense to minimize the differences between nEML error envelope and the estimated multipath error as shown in Fig. 3.

Apparently, from Fig. 2, it can be seen that the quantity \( \frac{A_{R_{s+1}} - A_{R_0}}{A_{R_0}} \) constitutes a very good estimate of the actual error. The agreement between the actual nEML multipath
error and the estimated error by the above quantity is best for short delays. Therefore, the multipath error affecting a nEML tracking loop can be written as follows:

$$MP_{SBME} = c_{MOD,\text{TYPE}} \frac{\Delta R_{12} - \Delta R_0}{R_0}$$

(3)

where $MP_{SBME}$ is the multipath error in units of chips, $c_{MOD,\text{TYPE}}$ is the optimization coefficient for any modulation of type $MOD,\text{TYPE}$. The optimization coefficients and the first late slopes of the ideal normalized correlation function for BPSK and CBOC(-) modulated signals are listed in Table I. Finally, substituting $\Delta R_{12}$ and $\Delta R_0$ in Eqn. (3) with $R_{1+2}$ and $R_0$, we can write the following:

$$MP_{SBME} = -c_{MOD,\text{TYPE}} \left(1 - \frac{R_{1+2} - m_{MOD,\text{TYPE}} \Delta R_{12}}{R_0}\right)$$

(4)

Finally, substituting $\Delta R_{12}$ and $\Delta R_0$ in Eqn. (3) with $R_{1+2}$ and $R_0$, we can write the following:

$$MP_{SBME} = -c_{MOD,\text{TYPE}} \left(1 - \frac{R_{1+2} - m_{MOD,\text{TYPE}} \Delta R_{12}}{R_0}\right)$$

(4)

The multipath tracking performance is studied first via Multipath Error Envelopes (MEE) for static two path channel. In MEE analysis, the noise free environment is always considered, and the focus is on the multipath induced delay errors. Typically, a two path model is used to generate the MEE curves as in [6]. The multipath amplitude is 6 dB less than the LOS path amplitude. The MEE curves are obtained for two extreme phase variations (i.e., 0 and 180 degrees) of multipath signal with respect to LOS component. The multipath delays are varying from 0 to 1.4 chips with a step size of 0.04 chips. The MEE simulations were carried out for BPSK and CBOC(-) modulated signals. A more representative curve known as Running Average Error (RAE) is presented here in accordance with [6]. RAE is computed from the area enclosed within the multipath error and averaged over the range of the multipath delays from zero to the plotted delay values. Running average error results for BPSK and CBOC(-) modulated signals are shown in Fig. 4. It can be seen from Fig. 4 that SBME outperforms HRC and nEML for multipath delays less than 0.1 chips for BPSK, and multipath delays less than 0.35 chips for CBOC(-) signal in this ideal no-noise scenario.

### III. THEORETICAL ANALYSIS

The multipath tracking performance is studied first via Multipath Error Envelopes (MEE) for static two path channel. In MEE analysis, the noise free environment is always considered, and the focus is on the multipath induced delay errors. Typically, a two path model is used to generate the MEE curves as in [6]. The multipath amplitude is 6 dB less than the LOS path amplitude. The MEE curves are obtained for two extreme phase variations (i.e., 0 and 180 degrees) of multipath signal with respect to LOS component. The multipath delays are varying from 0 to 1.4 chips with a step size of 0.04 chips. The MEE simulations were carried out for BPSK and CBOC(-) modulated signals. A more representative curve known as Running Average Error (RAE) is presented here in accordance with [6]. RAE is computed from the area enclosed within the multipath error and averaged over the range of the multipath delays from zero to the plotted delay values. Running average error results for BPSK and CBOC(-) modulated signals are shown in Fig. 4. It can be seen from Fig. 4 that SBME outperforms HRC and nEML for multipath delays less than 0.1 chips for BPSK, and multipath delays less than 0.35 chips for CBOC(-) signal in this ideal no-noise scenario.

### Table I: Slopes and coefficients for different modulations

<table>
<thead>
<tr>
<th>MOD_TYPE</th>
<th>$m_{MOD,\text{TYPE}}$</th>
<th>$c_{MOD,\text{TYPE}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPSK</td>
<td>-1</td>
<td>0.46</td>
</tr>
<tr>
<td>CBOC(-)</td>
<td>-5.3847</td>
<td>0.05</td>
</tr>
</tbody>
</table>

### IV. SIMULATION RESULTS

Simulations are carried out in short-delay (i.e., less than 0.35 chips delay from LOS signal) multipath scenarios for BPSK and CBOC(-) signals. The simulation profile is
summarized in Table II. Rayleigh fading channel model is used in the simulation. The number of channel path is fixed to 2 with random path separation between 0.04 and 0.35 chips. The channel paths are assumed to obey a decaying Power Delay Profile (PDP), where the amplitude of the second path $\alpha_2$ is exponentially decaying with respect to the amplitude of the first path $\alpha_1$ and to the path separation: $\alpha_2 = \alpha_1 e^{-\mu x_{\max}}$, where $x_{\max}$ is the path separation and $\mu$ is a path decaying coefficient (here $\mu = 0.1$ chips). The received signal was sampled at $N_e = 24$ and 2 samples per BPSK and CBOC(-) interval, respectively. $N_e$ varies in order to have the same number of samples per chip for both the modulations.

The received signal duration is 800 milliseconds (ms) or 0.8 s for each particular C/N0 level. The tracking errors are computed after each $N_eN_{nc}$ msc (in this case, $N_eN_{nc} = 20$ ms) interval. In the final statistics, the first 600 ms are ignored in order to remove the initial error bias that may come from the acquisition stage. We run the simulations for 100 random realizations, which give a total of $10 \times 100 = 1000$ statistical points, for each C/N0 level. The Root-Mean-Square-Errors (RMSE) are plotted in meters, by using the relationship $RMSE_m = RMSE_{\text{chips}}cT_s$; where $c$ is the speed of light, $T_s$ is the chip duration, and $RMSE_{\text{chips}}$ is the RMSE in chips. RMSE vs. C/N0 plots are shown in Fig. 5. The SBME showed the best overall performance as compared to nEML and HRC in short-delay multipath scenarios. HRC is very sensitive to noise due to the extra pair of discriminations between the very-early and very-late gates, as compared to nEML and SBME.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path Profile</td>
<td>2 path Rayleigh channel</td>
</tr>
<tr>
<td>Path Power</td>
<td>Decaying PDP, $\mu = 0.1$ chips</td>
</tr>
<tr>
<td>Path Spacing</td>
<td>Random between 0.04 &amp; 0.35 chips</td>
</tr>
<tr>
<td>Path Phase</td>
<td>Random between 0 and $2\pi$</td>
</tr>
<tr>
<td>Early-Late Spacing, $\Delta_{EL}$</td>
<td>0.0833 chips</td>
</tr>
<tr>
<td>Coherent Integration, $N_c$</td>
<td>20 ms</td>
</tr>
<tr>
<td>Non-coherent Integration, $N_{nc}$</td>
<td>1 block</td>
</tr>
<tr>
<td>Oversampling Factor, $N_e$</td>
<td>$[24, 2]$</td>
</tr>
<tr>
<td>Initial Delay Error</td>
<td>0.0833 chips</td>
</tr>
<tr>
<td>Loop Filter Bandwidth</td>
<td>1 Hz</td>
</tr>
<tr>
<td>Loop Filter Order</td>
<td>1st order</td>
</tr>
<tr>
<td>Front-end Bandwidth</td>
<td>Infinite</td>
</tr>
</tbody>
</table>

V. CONCLUSIONS

In this paper, a novel multipath mitigation technique, SBME was proposed and implemented in ideal infinite bandwidth case. The multipath performance of the newly proposed algorithm along with the conventional DLLs was studied in terms of running average error along with the simulated root-mean-square-error for short-delay multipath scenarios. It was shown that the SBME provided the best overall performance as compared to nEML and HRC in short-delay multipath scenarios. The authors are currently working on the adaptation of SBME in band-limited case, which is very demanding for GNSS mass-market receivers.

REFERENCES


Publication P6


Copyright ©ENC GNSS 2010. Reprinted with permission.
Publication P7


DOI= [http://dx.doi.org/10.1155/2010/412393](http://dx.doi.org/10.1155/2010/412393)

Copyright ©2010 Hindawi Publishing Corporation. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
Research Article

Advanced Multipath Mitigation Techniques for Satellite-Based Positioning Applications

Mohammad Zahidul H. Bhuiyan and Elena Simona Lohan

Department of Communications Engineering, Tampere University of Technology, P.O. Box 553, 33101 Tampere, Finland

Correspondence should be addressed to Mohammad Zahidul H. Bhuiyan, mohammad.bhuiyan@tut.fi

Received 9 June 2010; Revised 17 August 2010; Accepted 20 October 2010

Copyright © 2010 M. Z. H. Bhuiyan and E. S. Lohan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Multipath remains a dominant source of ranging errors in Global Navigation Satellite Systems (GNSSs), such as the Global Positioning System (GPS) or the future European satellite navigation system Galileo. Multipath is generally considered undesirable in the context of GNSS, since the reception of multipath can make significant distortion to the shape of the correlation function used for time delay estimation. However, some wireless communications techniques exploit multipath in order to provide signal diversity though in GNSS, the major challenge is to effectively mitigate the multipath, since we are interested only in the satellite-receiver transit time offset of the Line-Of-Sight (LOS) signal for the receiver’s position estimate. Therefore, the multipath problem has been approached from several directions in order to mitigate the impact of multipath on navigation receivers, including the development of novel signal processing techniques. In this paper, we propose a maximum likelihood-based technique, namely, the Reduced Search Space Maximum Likelihood (RSSML) delay estimator, which is capable of mitigating the multipath effects reasonably well at the expense of increased complexity. The proposed RSSML attempts to compensate the multipath error contribution by performing a nonlinear curve fit on the input correlation function, which finds a perfect match from a set of ideal reference correlation functions with certain amplitude(s), phase(s), and delay(s) of the multipath signal. It also incorporates a threshold-based peak detection method, which eventually reduces the code-delay search space significantly. However, the downfall of RSSML is the memory requirement which it uses to store the reference correlation functions. The multipath performance of other delay-tracking methods previously studied for Binary Phase Shift Keying-(BPSK-) and Sine Binary Offset Carrier-(SinBOC-) modulated signals is also analyzed in closed loop model with the new Composite BOC (CBOC) modulation chosen for Galileo E1 signal. The simulation results show that the RSSML achieves the best multipath mitigation performance in a uniformly distributed two-to-four paths Rayleigh fading channel model for all three modulated signals.

1. Introduction

Multipath remains a dominant source of ranging errors in Global Navigation Satellite Systems (GNSSs), such as the Global Positioning System (GPS) or the future European satellite navigation system, Galileo. Several approaches have been used in order to reduce the multipath error. Among them, the use of special multipath-limiting antennas (i.e., choke ring or multibeam antennas), the postprocessing techniques to reduce carrier multipath, the carrier smoothing to reduce code multipath, and the code-tracking algorithms based on receiver internal correlation technique are the most prominent approaches [1]. In this paper, our focus is limited to the correlation-based multipath mitigation techniques, since they are the most widely used in commercial GNSS receivers. The classical correlation-based code tracking structure used in GNSS is based on a feedback delay estimator and is implemented via a feedback loop. The most known feedback-delay estimator is the Delay Lock Loop (DLL) or Early-Minus-Late (EML) loop, where two correlators spaced at one chip from each other are used in the receiver in order to form a discriminator function, whose zero crossings determine the path delays of the received signal [2–7]. The classical EML fails to cope with multipath propagation [1]. Therefore, several enhanced EML-based techniques have been introduced in the literature for the last two decades in order to mitigate the impact of multipath, especially in closely spaced path scenarios. One class of these enhanced
EML techniques is based on the idea of narrowing the spacing between the early and late correlators, that is, narrow EML (nEML) or narrow correlator [1, 8, 9]. The choice of correlator spacing depends on the receiver’s available front-end bandwidth along with the associated sampling frequency [10]. Correlator spacings in the range of 0.05 to 0.2 chips are commercially available for nEML-based GPS receivers [11].

Another family of discriminator-based DLL variants proposed for GNSS is the so-called Double-Delta (ΔΔ) technique, which uses more than 3 correlators in the tracking loop (typically, 5 correlators: two early, one in prompt and two late) [8]. ΔΔ technique offers better multipath rejection in medium-to-long delay multipath [9, 12] in good Carrier-to-Noise-density ratio (C/N₀) (and two late) [8]. Couple of well-known particular cases of ΔΔ technique are the High Resolution Correlator (HRC) [9], the Strobe Correlator (SC) [8, 13], the Pulse Aperture Correlator (PAC) [14], and the modified correlator reference waveform [8, 15]. One other similar tracking structure is the Multiple Gate Delay (MGD) correlator [16–19], where the number of early and late gates and the weighting factors used to combine them in the discriminator are the parameters of the model and can be optimized according to the multipath profile as illustrated in [12]. While coping better with the ambiguities of BOC correlation function, the MGD provides slightly better performance than the nEML at the expense of higher complexity and is sensitive to the parameters chosen in the discriminator function (i.e., weights, number of correlators, and correlator spacing) [12, 19]. In [12], it is also shown that ΔΔ technique is a particular case of MGD implementation.

Another tracking structure closely related to ΔΔ technique is the Early1/Early2 (E1/E2) tracker, initially proposed in [20] and later described in [8]. In E1/E2 tracker, the main purpose is to find a tracking point on the correlation function that is not distorted by multipath. The first step is to locate two correlators on the early slope of the correlation function. The correlation values of these two early correlators are then compared with the correlation values of an ideal reference correlation function. Finally, a delay-correction factor is computed based on the measured and reference correlation values of E1 and E2 correlators. As reported in [8], E1/E2 tracker shows some performance improvement over ΔΔ technique only for very short delay multipath for GPS L1 C/A signal (i.e., BPSK signal).

Another feedback-tracking structure is the Early-Late-Slope (ELS) [8], which is also known as Multipath Elimination Technique (MET) [21]. The ELS is based on two correlator pairs at both sides of the correlation function’s central peak with parameterized spacing. Once both slopes are known, they can be used to compute a pseudorange correction that can be applied to the pseudorange measurement. However, simulation results performed in [8] showed that ELS is outperformed by HRC with respect to Multipath Error Envelopes (MEEs), for both BPSK and SinBOC(1,1) modulated signals.

A new multipath-estimation technique, named as A-Posteriori Multipath Estimation (APME), is proposed in [22], which relies on a posteriori estimation of the multipath error tracking. Multipath error is estimated independently in a multipath-estimator module on the basis of the correlation values from the prompt and very late correlators. The performance in multipath environment reported in [22] is comparable with that of the SC: slight improvement for very short delays (i.e., delays less than 20 meters), but rather significant deterioration for medium delays. A similar slope-based multipath mitigation strategy, named as Slope-based Multipath Estimator (SBME), was proposed by the authors in [23]. SBME first derives a multipath estimation equation by utilizing the correlation shape of the ideal normalized correlation function, which is then used to compensate for the multipath bias of an nEML tracking loop. SBME requires an additional correlator at the late side of the correlation function, and it is used in-conjunction with an nEML tracking loop. It is reported in [23] that SBME has superior multipath mitigation performance than nEML in closely spaced two paths channel model.

The conventional techniques, discussed so far can be classified based on their correlator requirements as shown in Figure 1. For clarity reason, we use the notation correlator in this paper in order to represent complex correlator (i.e., one complex correlator is equivalent to two correlators needed for in-phase and quad-phase channels).

One of the most promising advanced multipath mitigation techniques is the Multipath Estimating Delay Lock Loop (MEDLL) [24–26] implemented by NovAtel for GPS receivers. The MEDLL uses many correlators in order to determine accurately the shape of the multipath corrupted correlation function. Then, a reference function is used in a software module in order to determine the best combination of LOS and Non-LOS (NLOS) components (i.e., amplitudes, delays, phases, and number of multipath). However, MEDLL provides superior long-delay multipath mitigation performance than nEML at the cost of expensive multicorrelator-based tracking structure. MEDLL is considered as a significant evolutionary step in the receiver-based attempt to mitigate-multipath. Moreover, MEDLL has stimulated the design of different maximum likelihood-based implementations for multipath mitigation. One such variant is the noncoherent MEDLL, developed by the authors, as described in [27]. Classical MEDLL is based on a maximum likelihood search, which is computationally extensive. The authors implemented a noncoherent version of MEDLL that reduces the search space by incorporating a phase-search unit, based on statistical distribution of multipath phases. However, the performance of this suggested approach depends on the number of random phases considered; this means that the larger the number is, the better the performance will be. But this will also increase the processing burden significantly. The results reported in [27] show that the noncoherent MEDLL provides very good performance in terms of Root-Mean-Square-Error (RMSE), but has a rather poor Mean-Time-to-Lose-Lock (MTLL) as compared to the conventional DLL techniques.

A new technique to mitigate multipath by means of correlator reference waveform was proposed in [28]. This technique, referred to as second derivative correlator, generates a signal correlation function which has a much narrower width than a standard correlation function and
is, therefore, capable of mitigating multipath errors over a much wider range of secondary path delays. The narrowing of correlation function is accomplished by using a specially designed code reference waveform (i.e., the negative of the second-order derivative of correlation function) instead of the ideal code waveform used in almost all existing receivers. However, this new technique reduces the multipath errors at the expense of a moderate decrease in the effective Signal-to-Noise Ratio (SNR) due to the effect of narrowing the correlation function. A similar strategy, named as Slope Differential (SD), is based on second order derivative of the correlation function [29]. It is shown in [29] that this technique has better multipath performance than nEML and Strobe Correlator. However, the performance measure was solely based on the theoretical MEE curves, thus its potential benefit in more realistic multipath environment is still an open issue.

A completely different approach to mitigate multipath error is used in NovAtel’s recently developed vision correlator [30]. The Vision Correlator (VC) is based on the concept of Multipath Mitigation Technique (MMT) developed in [31]. It can provide a significant improvement in detecting and removing multipath signals as compared to other standard multipath-resistant code-tracking algorithms (e.g., PAC of NovAtel). However, VC has the shortcoming that it requires a reference function shape to be used to fit the incoming data with the direct path and the secondary path reference signals. The reference function generation has to be accomplished a priori, and it must incorporate the issues related to Radio Frequency (RF) distortions introduced by RF front end.

Several advanced multipath mitigation techniques were also proposed by the authors in [27, 32]. While improving the delay-estimation accuracy, these techniques require a higher number of correlators than the traditional DLL, and they are sensitive to the noise-dependent threshold choice. Among these advanced techniques, two most competitive ones, previously proposed by the authors, are selected herein for performance comparison. These are Peak Tracking, based on 2nd-order Differentiation (PT(Diff2)), and Teager-Kaiser- (TK-) based delay estimation, the details of which can be found in [27].

Many correlation-based multipath mitigation techniques exist, but even the most promising ones (e.g., nEML, HRC, PT(Diff2), etc.) are not good enough for closely spaced multipath environment, which is a key motivation for present-day researchers (as is the case in this research) to come up with new innovative techniques. The purpose of this paper is twofold: first, to propose a novel maximum likelihood-based Reduced Search Space Maximum Likelihood (RSSML) delay estimator as an advanced multipath mitigation technique, mostly designed for harsh multipath environment (where there can be more than two strong closely spaced paths) and second, to analyze the performance of other contemporary multipath mitigation techniques (both conventional and advanced techniques) under the same unified simulation model. Additionally, the authors also develop a combined TK- and nEML-based approach, named here as TK+nEML, which is less complex than TK, while at the same time provide better multipath mitigation than nEML. The motivation for such a combined approach will be discussed in more detail in Section 4.2. The authors remark here that the basic idea of RSSML was first introduced in [33], where RSSML was implemented for two paths channel with infinite bandwidth assumption. Moreover, the version of RSSML presented in [33] was not optimized in terms of memory, since it required a large set of correlation functions for all possible delays in a predefined code delay window range. The RSSML with its current version requires a large set of correlation functions only for the prompt correlator, and it is also adapted for finite bandwidth assumption for any number of paths up to four. Simulation results in fading multipath environment are included in this paper in order to compare the performance of the proposed techniques with the various conventional DLLs and other developed advanced techniques (which are briefly reviewed here). The performance of these techniques are analyzed for the newly defined Composite Binary Offset Carrier (CBOC) modulation along with the existing Binary Phase Shift Keying (BPSK) and Sine Binary Offset Carrier (SinBOC) modulations.

The rest of this paper is organized as follows. Section 2 presents the signal and channel model, followed by a description on multicorrelator-based delay-tracking structure in
Section 3. The advanced multipath mitigation techniques including the proposed RSSML are introduced in Section 4, followed by a detailed analysis on implementation issues for RSSML in Section 5. Section 6 shows the multipath performance of the selected techniques in terms of semianalytical running average error. Section 7 presents the simulation results in two-to-four paths fading channel model with finite front-end bandwidth whereas Section 8 provides a comparison between different techniques in terms of their multipath mitigation capability, relative complexity and needed a priori information. Finally, some general conclusions are drawn in Section 9, with a perspective on future research direction.

2. Signal and Channel Model

Typical GNSS signals, such as those used in GPS or Galileo, employ the Direct Sequence-Code Division Multiple Access (DS-CDMA) technique, where a Pseudorandom Noise (PRN) code from a specific satellite is spreading the navigation data over $S_F$ chips (or over a code-epoch length) [34, 35]. In what follows, a baseband model is adopted for clarity reason. The estimation of code delay in today’s receivers is typically done in digital domain using the baseband correlation samples. In the following, the time notation $t$ denotes the discrete time instant. The signal $x(t)$ transmitted from one satellite with a specific PRN code can be written as

$$x(t) = \sqrt{E_b} p_{\text{mod}}(t) \otimes c(t),$$

(1)

where $E_b$ is the bit energy, $p_{\text{mod}}(t)$ is the modulation waveform (i.e., BPSK for GPS L1 C/A code or CBOC(·) for Galileo E1C signals), and $c(t)$ is the navigation data after spreading as written below (spreading is done with a PRN code of chip interval $T_c$ and spreading factor $S_F$)

$$c(t) = \sum_{n=-\infty}^{\infty} b_n \sum_{k=1}^{S_F} \delta(t - nT_cS_F - kT_c).$$

(2)

Above $\delta(\cdot)$ is the Dirac unit pulse, $b_n$ is the $n$th data bit (for pilot channels, $b_n = 1, \forall n$), and $c_{k,n}$ is the $k$th chip ($\pm 1$ valued) corresponding to the $n$th spread bit.

The modulation waveform for BPSK or BOC can be written as [36]

$$p_{\text{mod}}(t) = p_{TB}(t) \otimes \sum_{i=0}^{N_B-1} \delta(t - iT_B),$$

(3)

where $N_B$ is BOC modulation order: $N_B = 1$ for BPSK modulation (BPSK can be seen as a particular case of BOC modulation, as illustrated in [36]) and $N_B = 2f_s/f_c$, where $f_c$ is the subcarrier frequency and $f_s$ is the carrier frequency for BOC modulation, $T_B = T_c/N_B$ is the BOC interval, and $p_{TB}(t)$ is the pulse-shaping filter (e.g., for unlimited bandwidth case, $p_{TB}(t)$ is a rectangular pulse of width $T_B$ and unit amplitude).

The signal $x(t)$ is typically transmitted over a multipath static or fading channel, where all interference sources except the multipath are lumped into a single additive Gaussian noise term $\eta(t)$

$$r(t) = \sum_{l=1}^{L} a_l x(t - \tau_l)e^{j(2\pi f_d t + \theta_l)} + \eta(t),$$

(4)

where $r(t)$ is the received signal, $L$ is the number of channel paths, $a_l$ is the amplitude of the $l$th path, $\theta_l$ is the phase of the $l$th path, $\tau_l$ is the channel delay introduced by the $l$th path (typically assumed to be slowly varying or constant within the observation interval), $f_d$ is the Doppler shift introduced by the channel, and $\eta(t)$ is a wideband additive noise, incorporating all sources of interferences over the channel. Assuming that the signal is sampled at $N_s$ samples per-chip (for BPSK) or per-BOC interval (for BOC modulation), then the power spectral density of $\eta(\cdot)$ can be written as $N_0/(N_cN_BS_F)$, where $N_0$ is the noise power in 1 kHz bandwidth (i.e., bandwidth corresponding to one code epoch). Generally, the SNR for any GNSS signal is constant within the observation interval),

$$C/N_0 \text{ [dB-Hz]} = \frac{E_b}{N_0} + 10 \log_{10}(B_w).$$

(5)

The delay tracking is typically based on the code epoch-by-epoch correlation $l(\cdot)$ between the incoming signal and the reference $x_{\text{ref}}(\cdot)$ modulated PRN code, with a certain candidate Doppler frequency $f_d$ and delay $\hat{\tau}$

$$R\left(\hat{\tau}, \hat{f}_D, m\right) = E\left(\frac{1}{T_{\text{sym}}} \int_{(m-1)T_{\text{sym}}}^{mT_{\text{sym}}} r(t)x_{\text{ref}}(\hat{\tau}, \hat{f}_D)dt\right),$$

(6)

where $m$ is the code epoch index, $T_{\text{sym}}$ is the symbol period (i.e., $T_{\text{sym}} = S_F T_c$), and $E(\cdot)$ is the expectation operator with respect to the random variables (e.g., PRN code, channel effects, etc.), and

$$x_{\text{ref}}(\hat{\tau}, \hat{f}_D) = p_{\text{mod}}(t) \otimes \sum_{n=-\infty}^{\infty} \sum_{k=1}^{S_F} \hat{b}_n c_{k,n} \times \delta(t - nT_{\text{sym}} - kT_c) \otimes p_{TB}(t)e^{-j2\pi \hat{f}_D t},$$

(7)

where $\hat{b}_n$ is the estimated data bits. For Galileo signals, a separate pilot channel is transmitted [34]. In what follows, it is assumed that data bits are perfectly estimated ($\hat{b}_n = b_n$), and removed before the correlation process. In a practical receiver, in order to cope with noise, coherent and noncoherent integration can be used. The average coherent correlation function $R_c(\hat{\tau}, \hat{f}_D)$ can be written as

$$R_c(\hat{\tau}, \hat{f}_D) = \frac{1}{N_r} \sum_{m=1}^{N_r} R(\hat{\tau}, \hat{f}_D, m),$$

(8)

where $N_r$ is the coherent integration time expressed in code epochs or milliseconds for GPS or Galileo signal, and the
noncoherently averaged correlation function \( \mathcal{R}_{nc}(\hat{\tau}, \hat{f_D}) \) can be written as

\[
\mathcal{R}_{nc}(\hat{\tau}, \hat{f_D}) = \frac{1}{N_{nc}} \sum_{N_{nc}} \frac{1}{N_c} \sum_{m=1}^{N_c} \mathbf{R}(\hat{\tau}, \hat{f_D}, m) \bigg|_{p_{nc}}
\]

where \( N_{nc} \) is the noncoherent integration time expressed in blocks of length \( N_c \) milliseconds (for clarity reason, we avoid using the block indexes for the noncoherent summations), and \( p_{nc} \) is the power index used for noncoherent summation. The most encountered variants for \( p_{nc} \) are: \( p_{nc} = 1 \) (which is the sum of absolute correlation values), and \( p_{nc} = 2 \) (i.e., which is the sum of squared-absolute correlation values). We prefer to use the later option (i.e., \( p_{nc} = 2 \)) in our simulations.

3. Multicorrelator Based Delay-Tracking Structure

Compared with the conventional EML tracking loop, where only three correlators are used (i.e., Early, Prompt and Late), here, in the multicorrelator-based structure, we generate a bank of correlators (e.g., in this implementation, we use 193 correlators with 0.0208 chips spacing between successive correlators) as presented in Figure 2. This large number of correlators is needed in order to include the advanced multipath mitigation techniques in the comparison, because these techniques make use of these correlators for estimating the channel properties while taking decision about the code delay [27]. Some of these correlators can be kept inactive or unused, for example when EML and HRC tracking loops are used. After the necessary front-end processing, and after the carrier has wiped off, the received postprocessed signal was passed through a bank of correlators. As shown in Figure 2, the NCO and PRN generator block produces a bank of early and late versions of replica codes based on the delay of the LOS signal \( \hat{\tau} \), the correlator spacing \( \Delta \), and the number of correlators \( M \). In case of EML-tracking loop, the corresponding early-late spacing is equal to \( 2\Delta \). The received signal is correlated with each replica in the correlator bank, and the output of the correlator bank is a vector of samples in the correlation envelope. Therefore, we obtain the correlation values for the range of \( \pm M\Delta \) chips from the prompt correlator, where \( M \) is the number of correlators and \( \Delta \) is the correlator spacing between successive correlators. The various code tracking techniques (named as discriminator in Figure 2) utilize the correlation values as input, and generate the estimated LOS delay as output, which is then smoothed by a loop filter. In accordance with [35], the implemented code loop filter is a 1st order filter, whose function can be written as

\[
\hat{\tau}(k + 1) = \hat{\tau}(k) + \gamma d(k),
\]

where \( \gamma \) is calculated based on loop filter bandwidth, \( B_{\text{ff}} \). A DLL loop bandwidth of 2 Hz is used in the simulation, assuming that carrier aiding is always available [9].

4. Advanced Multipath Mitigation Techniques

The advanced state-of-the-art multipath-mitigation techniques discussed in Section 1 are classified here based on their mitigation strategies, as shown in Figure 3. These advanced techniques usually require a vast number of correlators in order to estimate the channel characteristics, which are then used to mitigate the multipath effect. Several multipath-mitigation techniques introduced in past years are based on Maximum Likelihood (ML) estimation principle. Examples of ML-based techniques include MEDLL [26], MMT [31], VC [30] of NovAtel, MEDLL of Tampere University of Technology (TUT) [27], and the proposed RSSML. Among other techniques, second derivative [28], slope differential [29], and PT(Diff2) [27] are based on 2\textsuperscript{nd}-order differentiation whereas TK, PT(TK) and TK + nEML are based on Teager Kaiser operator. In the following subsections, only those algorithms are elaborated, which will later be considered for performance analysis. It is nice to mention here that a brief discussion of the remaining algorithms has already been presented in Section 1.

4.1. Teager Kaiser. The Teager Kaiser-based delay-estimation technique is based on the principle of extracting the signal energy corresponding to various channel paths via the non-linear TK operator. The output \( \Psi_{TK}(x(n)) \) of TK operator applied to a discrete signal \( x(n) \), can be defined as [38]

\[
\Psi_{TK}(x(n)) = x(n-1)x^n(n-1)
\]

\[
-\frac{1}{2}[x(n-2)x^n(n) + x(n)x^n(n-2)].
\]

The input of TK operator can be the noncoherent correlation function. The output of TK operator can indicate the presence of a multipath component more clearly than looking directly at the correlation function. According to (11), at least 3 correlation values are needed to compute TK (in prompt, early, and very early). But usually, TK-based delay estimation utilizes a higher number of correlators and is sensitive to the noise dependent threshold choice. Firstly, it computes the noise variance according to the explanation presented in Section 5.1, which is then used to compute an adaptive threshold as defined in [27]. The peaks which are above the adaptive threshold are considered as competitive peaks. Among all the competitive peaks, TK selects the delay associated to that competitive peak which has the closest delay difference from the previous delay estimate.
TK-based technique is chosen in the context of the paper since it has been proved that it can give very good results in the delay-estimation process when used with CDMA type of signals, as presented in [27, 39]. Most recently, TK has been studied also in closed-loop model for SinBOC(1,1), modulated two paths channel model, and its performance was one of the best among the considered algorithms [40]. One major limitation of TK-based technique is the fact that they are quite sensitive to the filtering stages (i.e., when infinite bandwidth is unavailable). The impact of the bandwidth limitation on TK performance is seldomly addressed in the literature, and hence, it is included in our algorithms’ list for performance analysis under bandwidth limitation.

4.2. Combined Approach: Teager Kaiser and Narrow EML

A combined simplified approach with Teager Kaiser and narrow EML is implemented in order to justify the feasibility of having an nEML discrimination after the TK operation on the noncoherent correlation function. In this combined approach, TK operator is first applied to the noncoherent correlation function, and then nEML discrimination is applied to the TK output. The motivation for this combined approach comes from the fact that, when we apply TK operation to the noncoherent correlation function, it usually makes the main lobe of the noncoherent correlation function (after TK operation) much more steeper. This eventually reduces the effect of multipath in case of TK-based nEML (TK + nEML) as compared to nEML, as illustrated in Figure 4. In Figure 4, TK + nEML has a zero crossing at 0.014 chips away from the true delay whereas nEML has a zero crossing at 0.029 chips away from the true delay. Therefore, TK + nEML has superior multipath performance (around 4.1 meters of multipath error) as compared to nEML (around 8.5 meters of multipath error) for this particular scenario. On the contrary, TK+nEML restricts the code delay search range (i.e., the range where we expect our true code delay to be located) to be much narrower as compared to nEML, which eventually increases the risk for the combined approach to lock at any of the false zero crossings, in cases when the initial coarse delays are poorly estimated.

4.3. Peak Tracking

Peak Tracking- (PT-) based techniques, namely, PT based on 2nd-order differentiation (PT(Diff2)) and PT based on Teager Kaiser (PT(TK)), are first proposed in [27, 41]. Both of these techniques utilize the adaptive threshold computed from the estimated noise variance of the channel in order to decide on the correct code delay. These advanced techniques first generate competitive peaks which are above the computed adaptive threshold.
as explained in Sections 5.1 and 5.2. For each of the competitive peak, a decision variable is formed based on the peak power, the peak position and the delay difference of the peak from the previous delay estimate. Finally, the PT techniques select the peak which has the maximum weight as being the best LOS candidate. PT(Diff2) is included in our list of multipath mitigation techniques for performance analysis to be presented in Sections 6 and 7, since it has superior multipath mitigation performance over PT(TK) [27].

4.4. Reduced Search Space Maximum Likelihood Delay Estimator. In the presence of multipath, we recall that the received signal at the input of a GNSS receiver can be expressed as in (4). We rewrite (4) below for further clarification

\[ r(t) = \sum_{l=1}^{L} a_l x(t - \tau_l) e^{i(2\pi f_s t + \phi_l)} + n(t). \]  (12)

In the above equation, as explained earlier, \( x(t) \) is the spread-spectrum code, \( n(t) \) is the white Gaussian noise, and \( a_l, \tau_l, \theta_l \) are the amplitude, delay, and phase of the \( l \)th signal, respectively. For any GNSS signal, one of the most important parameter of interest is the LOS code delay. A conventional DLL (e.g., nEML) is not able to follow the LOS code delay accurately, since it does not take into consideration the bias contributed by the multipath components. The proposed RSSML attempts to compensate the multipath error contribution by estimating the multipath parameters along with the LOS signal. If \( r(t) \) is observed for a certain time \( T_{coh} \), that is short enough to assume that the parameters are constant, then the Maximum Likelihood Estimation (MLE) theory can be applied to estimate those parameters. The MLE principle states that the estimate of a certain parameter with the smallest mean square error is the estimate that maximizes the conditional probability density function of \( r(t) \). According to MLE, RSSML calculates the estimated signal parameters (i.e., path delays, path amplitudes, and phase paths), which minimize the mean square error of \( L(\hat{\tau}, \hat{a}, \hat{\theta}) \), as specified in

\[ L(\hat{\tau}, \hat{a}, \hat{\theta}) = \int_{T_{coh}}^{t} [r(t) - s(t)]^2 dt, \]

\[ s(t) = \sum_{l=1}^{L} \hat{a}_l x(t - \hat{\tau}_l) e^{i(2\pi f_s t + \hat{\phi}_l)}. \]  (13)

Here, \( s(t) \) is the estimate of the LOS as well as multipath signals, and we assume that the Doppler shift \( f_D \) is correctly estimated by the carrier tracking loop (i.e., \( f_D = f_D \)) and that all the multipath components experience similar Doppler shift (i.e., \( f_D = f_D \)). The first assumption is valid as long as we are at the fine tracking stage (i.e., signal has been tracked for a while). Equation (13) can be solved by setting the partial derivatives of \( L(\hat{\tau}, \hat{a}, \hat{\theta}) \) to zero. The resulting equations for the \( l \)th signal can be written as follows in accordance with [25]:

\[ \hat{\tau}_l = \max_{\tau} \left\{ \text{Re} \left[ \sum_{n=1}^{L} \hat{a}_n R_{\text{ideal}}(\tau - \hat{\tau}_n) e^{i\hat{\phi}_n} e^{-j\hat{\theta}_n} \right] \right\}, \]

\[ \hat{\theta}_l = \arg \left[ \sum_{n=1}^{L} \hat{a}_n R_{\text{ideal}}(\tau_l - \hat{\tau}_n) e^{i\hat{\phi}_n} e^{-j\hat{\theta}_n} \right]. \]  (14)

In the above equations, \( R_{rx}(\tau) \) is the received down-converted correlation function, and \( R_{\text{ideal}}(\tau) \) is the ideal reference correlation function, the expression of which can be found in [36]. Generally speaking, RSSML performs a nonlinear curve fit on the input correlation function which finds a perfect match from a set of ideal reference correlation functions with certain amplitude(s), phase(s), and delay(s) of the multipath signal. Conceptually, a conventional spread-spectrum receiver does the same thing, but for only one signal (i.e., the LOS signal). With the presence of multipath signal, RSSML tries to separate the LOS component from the combined signal by estimating all the signal parameters in MLE sense, which consequently achieves the best curve fit on the received input correlation function. The total number of path components \( L \) is generally unknown to the receiver and, therefore, has to be estimated. One possible way to estimate \( L \) is to compute the mean square error for \( L = 1, 2, \ldots, L_{\text{max}} \) number of paths and select \( L \) with which we obtain the minimum mean square error. In this implementation, \( L_{\text{max}} \) is chosen such that the total number of path components does not exceed 3 (i.e., \( L_{\text{max}} = 3 \)).

In a multicorrelator-based structure, the estimated LOS delay, theoretically, can be anywhere within the code delay window range of \( \pm \tau_w \) chips, though in practice, it is quite likely to have a delay error around the previous delay estimate. The code delay window range essentially depends on the number of correlators (i.e., \( M \)) and the spacing between the correlators (i.e., \( \Delta \)) according to

\[ \tau_w = \frac{\pm (M - 1)}{2}. \]  (15)

For example, if 193 correlators are used with a correlator spacing of 0.0208 chips, then the resulting code-delay window range will be \( \pm 2 \) chips with respect to prompt correlator.
Therefore, the LOS delay estimate can be anywhere within this ±2 chips window range. The ideal noncoherent reference correlation functions are generated for up to \( L_{\text{max}} \) paths only for the middle delay index (i.e., \((M + 1)/2\)th delay index; for \( M = 193 \), the middle delay index is 97). These ideal correlation functions for the middle delay index are generated offline and saved in a lookup table in memory. In real time, RSSML reads the correlation values from the lookup table, translates the ideal reference correlation functions at the middle delay index to the corresponding candidate delay index within the code delay window, and then computes the Minimum Mean Square Error (MMSE) for that specific delay candidate. Instead of considering all possible LOS delays within a predefined code delay window as delay candidates, the search space is first reduced to some competitive peaks which are generated based on the computed noise thresholds as explained in Section 5. This will eventually reduce the processing time required to compute the MMSE (i.e., MMSE needs to be computed only for the reduced search space).

### 5. Implementation Issues for RSSML

The implementation of RSSML is discussed here for better clarification. Setting the partial derivatives of (13) to zero yields a set of nonlinear equations, as presented in (14). To overcome the difficulty of solving these equations, the RSSML generates a set of ideal noncoherent reference correlation functions for the middle correlator of a certain code-delay window range with various multipath delays, phases, and amplitudes. This means that, we generate \( s(t) \) in (13), by varying all multipath components for the middle correlator (e.g., the 97th correlator for a code-delay window range of ±2 chips with 0.0208 chips correlator spacing) of the code-delay window in order to obtain a discrete set of ideal noncoherent reference correlation functions. The set of multipath parameters can be specified as follows:

\[
\begin{align*}
A &= \forall a \{0 \leq a \leq 1\}, \\
T &= \forall \tau \{0 \leq \tau \leq 1\}, \\
\Theta &= \forall \theta \{0 \leq \theta \leq 2\pi\},
\end{align*}
\]

where \(|A| = p, |T| = q\), and \(|\Theta| = r\) are the cardinalities of the sets \( A, T, \) and \( \Theta \), respectively. The cardinality of each set will depend on the resolution of the multipath parameters within the given range. However, the complexity will increase as the cardinality of any set increases. The step-by-step procedure for RSSML is summarized below.

#### 5.1. Step 1: Noise Estimation

The correlation values for early time delays (i.e., \(< -1\) chip from the prompt correlator) are not affected by any multipath components since the multipath components are always delayed with respect to the LOS component. The noise level is estimated by taking the mean out-of-1-chip values at the early side from the prompt correlator of the normalized noncoherent correlation function as explained in Figure 5.

#### 5.2. Step 2: Competitive Peak Generation

The competitive peaks are those peaks which are generated based on the estimated noise level as obtained from step 1. A peak threshold is computed based on the estimated noise threshold plus some weighting factor as defined in [27]. The weighting factors are chosen in such a way that they reduce the possible risk that may arise due to the side lobes of the SinBOC(1,1) or CBOC(-) correlation. Therefore, the weighting factors chosen for SinBOC(1,1) and CBOC(-) modulations are slightly different from that of BPSK. It is worth to mention here that we use the same weighting factors for CBOC(-) and SinBOC(1,1) modulations, since they have almost similar correlation shape. As shown in Figure 5, in this example case, there is only one competitive peak which is above the computed peak threshold. The search space is then reduced from a large number of correlators to some competitive delay candidates (serving here as competitive peaks).

#### 5.3. Step 3: Reference Correlation Functions Reading and Translation

The RSSML first reads the ideal noncoherent reference correlation functions (which are generated offline for the middle/prompt correlator) from the lookup table. Next, it translates the correlation functions at the middle delay index to the corresponding candidate delay index within the code delay window for each competitive peak (which are already obtained from step 2). While doing the translation, RSSML truncates the ideal reference correlation values to zero which fall outside the code delay window range (i.e., correlation values outside ±2 chips from the middle correlator are truncated to zero).

#### 5.4. Step 4: MMSE Computation

The RSSML computes the MMSE for each candidate delay index corresponding to a competitive peak obtained from step 2.

#### 5.5. Step 5: LOS Delay Estimation

The candidate delay index with the lowest MMSE is chosen as the estimated LOS delay.
6. Semianalytical Running Average Error

The most typical way to evaluate the performance of a multipath mitigation technique is via Multipath Error Envelopes (MEE). Typically, two paths, either in phase or out of phase, are assumed to be present, and the multipath errors are computed for multipath delays up to 1.2 chips at maximum, since the multipath errors become less significant after that. The upper multipath error envelope can be obtained computed for multipath delays up to 1.2 chips at maximum, where the paths are in phase and the lower multipath error envelope when the paths are out of phase (i.e., 180° phase difference). In MEE analysis, several simplifying assumptions are usually made in order to distinguish the performance degradation caused by the multipath errors only. Such assumptions include zero Additive-White-Gaussian-Noise (AWGN), ideal infinite-length PRN codes, and zero residual Doppler. Under these assumptions, the correlation \( R_x(\tau) \) between the reference code of modulation type MOD (e.g., BPSK or CBOC(-)) and the received MOD-modulated signal via an L-path channel can be written as

\[
R_x(\tau) = \sum_{l=1}^{L} a_l e^{j\theta_l} R_{MOD}(\tau - \tau_l),
\]

where \( a_l, \theta_l, \tau_l \) are the amplitude, phase, and delay, respectively, of the \( l \)th path; and \( R_{MOD}(\tau) \) is the autocorrelation function of a signal with modulation type MOD. The analytical expressions for MEEs become complicated in the presence of more than two paths due to the complexity of channel interactions. Therefore, an alternative Monte-Carlo simulations-based approach is proposed herein for multipath error analysis in more than one path scenarios (i.e., for \( L \gtrsim 2 \)). First, a sufficient number of random realizations, \( N_{\text{random}} \) are generated (i.e., in the simulations, we choose \( N_{\text{random}} = 2000 \) ), and then we look at absolute mean error for each path delay over \( N_{\text{random}} \) points. The objective here is to analyze the multipath performance of various tracking techniques in the presence of more than two channel paths, which may occur in urban or indoor scenarios.

The following assumptions are made while running the simulations for generating the curves of Running Average Error (RAE). The channel follows a decaying Power Delay Profile (PDP), which can be expressed by the equation:

\[
at_l = a_l \exp^{-\mu(\tau_l - \tau_1)},
\]

where \( (\tau_l - \tau_1) \neq 0 \) for \( l > 1, \mu \) is the PDP coefficient (assumed to be uniformly distributed in the interval [0.05; 1.0], when the path delays are expressed in samples). The channel path phases \( \theta_l \) are uniformly distributed in the interval [0; 2\pi], and the number of channel paths \( L \) is uniformly distributed between 2 and \( L_{\text{max}} \), where \( L_{\text{max}} \) is set to 4 in the simulations. A constant successive path spacing \( x_{sl} \) is chosen in the range [0; 1.167] chips with a step of 0.0417 chips (which will define the multipath delay axis in the running average error curves). It is worth to mention here that the number of paths reduced to one LOS path when \( x_{sl} = 0 \). The successive path delays can be found using the formula \( \tau_l = lx_{sl} \) in chips. Therefore, for each channel realization (which is a combination of amplitudes \( \bar{a} = a_1, \ldots, a_L \), phases \( \bar{\theta} = \theta_1, \ldots, \theta_L \), fixed path spacings, and the number of channel paths \( L \)), a certain LOS delay is estimated \( \hat{\tau}_l(\bar{a}, \bar{\theta}, L) \) from the zero crossing of the discriminator function (i.e., \( D(\tau) = 0 \)), when searched in the linear range of \( D(\tau) \). The estimation error due to multipath is \( \hat{\tau}_l(\bar{a}, \bar{\theta}, L) - \tau_1 \), where \( \tau_1 \) is the true LOS path delay. The RAE curves are generated in accordance with [42]. RAE is actually computed from the area enclosed within the multipath error and averaged over the range of the multipath delays from zero to the plotted delay values. Therefore, in order to generate the RAE curves, the absolute mean error is computed for all \( N_{\text{random}} \) random points via

\[
\text{AME}(x_{sl}) = \text{mean}\left(\left| \hat{\tau}_l(\bar{a}, \bar{\theta}, L) - \tau_1 \right|\right),
\]

where \( \text{AME}(x_{sl}) \) is the mean of absolute multipath error for the successive path delay \( x_{sl} \). Now, the running average error for each particular delay in the range [0; 1.167] chips can be computed as follows:

\[
\text{RAE}(x_{sl}) = \frac{\sum_{i=1}^{i} \text{AME}(x_{sl})}{i},
\]

where \( i \) is the successive path delay index and RAE \( x_{sl} \) is the RAE for the successive path delay \( x_{sl} \). The RAE curves for three different modulations are shown in Figure 6.

7. Simulation Results

The semianalytical results from Section 6 have also been validated via simulations in fading multipath channels. Simulations have been carried out in closely spaced multipath scenarios for BPSK-, SinBOC(1,1)-, and CBOC(-)-modulated signals for a finite front-end bandwidth. The simulation profile is summarized in Table 1. Rayleigh fading channel model is used in the simulation, where the number of channel paths follows a uniform distribution between two and four. The successive path separation is random between 0.02 and 0.35 chips. The channel paths are assumed to obey a decaying PDP following (18), where \( \mu = 0.1 \) (when the path delays are expressed in samples). The received signal was sampled at \( N_r = 48, 24, \) and 4 for BPSK-, SinBOC(1,1)- and CBOC(-)-modulated signals, respectively. \( N_r \) varies in order to have the same number of samples per chip for all the three cases.

The received signal duration is 800 milliseconds (ms) or 0.8 seconds for each particular C/N0 level. The tracking errors are computed after each \( N_c N_{\text{rec}} \) ms (in this case, \( N_c N_{\text{rec}} = 20 \) ms) interval. In the final statistics, the first 600 ms are ignored in order to remove the initial error bias that may come from the delay difference between the received signal and the locally generated reference code. Therefore, for the above configuration (i.e., code loop filter parameters and the first path delay of 0.2 chips), the leftover tracking errors after 600 ms are mostly due to the effect of multipath only, as shown in Figure 7. We run the simulations for 100 random realizations, which give a total of 10 * 100 = 1000 statistical points, for each C/N0 level. The Root-Mean-Square-Errors (RMSE) of delay estimates are plotted
in meters, by using the relationship RMSE\textsubscript{m} = RMSE\textsubscript{chips}C\textsubscript{Tc}, where \( c \) is the speed of light, \( T_c \) is the chip duration, and RMSE\textsubscript{chips} is the RMSE in chips. RMSE versus \( C/N_0 \) plots for the given multipath-channel profile are shown in Figure 8. Additionally, a RMSE versus \( C/N_0 \) plot is presented in Figure 9 for SinBOC(1,1)-modulated single path signal in order to show the performance of the mitigation techniques in the absence of any multipath. In this no-multipath scenario, nEML has the best tracking performance from \( C/N_0 \) 35 dB-Hz and higher whereas RSSML showed the best tracking performance in 30 dB-Hz, and slightly worse performance than nEML from \( C/N_0 \) 35 dB-Hz and higher.

8. Performance Comparison

Table 2 shows the comparison between the different discussed techniques in terms of closely spaced multipath performance, semianalytical running average error performance, correlator requirement (in other words, code delay window length at the tracking stage), a priori information needed as input, channel estimation requirement, memory requirement, and complexity analysis as a whole. This comparison is solely based on the simulation results described in Sections 6 and 7.

It can be seen from Figure 8 that the proposed RSSML showed the best multipath performance in closely spaced
Figure 7: One snapshot of delay tracking for SinBOC(1,1) signal in 3 path fading channel with 24 MHz BW.

Table 1: Simulation profile description.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel model</td>
<td>Rayleigh fading channel</td>
</tr>
<tr>
<td>Number of paths</td>
<td>(between 2 to 4)</td>
</tr>
<tr>
<td>Path power</td>
<td>Decaying PDP with $\mu = 0.1$</td>
</tr>
<tr>
<td>Path spacing</td>
<td>Random between 0.02 and 0.35 chips</td>
</tr>
<tr>
<td>Path phase</td>
<td>Random between 0 and $2\pi$</td>
</tr>
<tr>
<td>Oversampling factor, $N_s$</td>
<td>[48, 24, 4]</td>
</tr>
<tr>
<td>E-L Spacing, $\Delta_{EL}$</td>
<td>0.0833 chips</td>
</tr>
<tr>
<td>Number of Correlators, $M$</td>
<td>193</td>
</tr>
<tr>
<td>Double-sided Bandwidth, BW</td>
<td>24 MHz</td>
</tr>
<tr>
<td>Filter type</td>
<td>FIR</td>
</tr>
<tr>
<td>Filter order</td>
<td>6</td>
</tr>
<tr>
<td>Coherent integration, $N_C$</td>
<td>20 ms</td>
</tr>
<tr>
<td>Noncoherent integration, $N_{nc}$</td>
<td>1 block</td>
</tr>
<tr>
<td>Initial delay error</td>
<td>$\pm 0.1$ chips</td>
</tr>
<tr>
<td>First path delay</td>
<td>0.2 chips</td>
</tr>
<tr>
<td>Code tracking loop bandwidth</td>
<td>2 Hz</td>
</tr>
<tr>
<td>Code tracking loop order</td>
<td>1st order</td>
</tr>
</tbody>
</table>

two to four paths fading channel model for all three modulation types. All other techniques have varying multipath performance with varying $C/N_0$ and varying modulation types. In general, PT(Diff2) performs better for SinBOC(1,1) and CBOC(−) signals whereas HRC performs better for all three modulations, but only in good $C/N_0$ (i.e., 40 dB-Hz and higher). It is interesting to note here that all the techniques except the proposed RSSML tend to show similar performance (within few meters of error bounds) in this two to four paths fading channel profile with a reasonably high PDP factor 0.1, as seen in Figure 8.

The semianalytical RAE performance is shown in Figure 6. It is obvious from Figure 6 that the proposed RSSML showed superior performance in terms of RAE as compared to other techniques in this no noise two to four paths static channel model. Among other techniques, PT(Diff2) and TK showed very good performance followed by HRC and TK + nEML. The RAE analysis is quite theoretical from two perspectives: firstly, the delay estimation is a one-shot estimate and does not really include any tracking loop in the process, and secondly, the analysis is usually carried out with ideal noise free assumption. These facts probably explain the reason why an algorithm which performs very good with respect to RAE may not necessarily provide the same performance in more realistic closed-loop fading channel model, especially in the presence of more than two channel paths. However, MEE or RAE analysis has been widely used by the research community as an important tool for analyzing the multipath performance due to simpler implementation and also due to the fact that it is hard to isolate multipath from other GNSS error sources in real life.

The complexity of any multipath mitigation technique mainly depends on the correlation structure and the implementation issues concerning channel estimation, correlator requirement, required number of mathematical operations, memory requirement, and so on. The advanced mitigation techniques are usually complex, since they generally utilize a large number of correlators for channel estimation, which are then used to estimate the first arriving path delay. Among the advanced techniques, the proposed RSSML is the most complex one, since it requires a large set of reference correlation functions which are generated offline to be used as a-priori information while estimating the code delay of first arriving path (please visit Section 5 for details). The memory size will eventually depend on few factors including the maximum number of paths to be considered, the correlator spacing, the number of correlators and the resolution of each multipath parameter (i.e., path delays, path phases, and path amplitudes). In the current MATLAB implementation, the RSSML requires approximately 14 megabytes of memory for each particular modulation with maximum number of paths set to 3, the correlator spacing set to 0.0208 chips, the number of correlators for window length of 4 chips set to 193. However, it is possible to reduce the memory requirement by adjusting the parameters appropriately. The impact of memory optimization is not analyzed here, and hence, it is kept open for future research.

9. Conclusions

Multipath is one of the major dominant sources in high-precision-oriented GNSS applications. Many receiver architectures exist in the market which employ a variety of
multipath mitigation techniques. Most of these techniques provide very good multipath mitigation for medium-to-long delay multipath. However, the multipath studies presented in most of the research papers are based on only two paths assumption, which is rather optimistic. In this study, a novel Reduced Search Space Maximum Likelihood delay estimator was proposed and the multipath performance was studied for short delay multipath where the number of paths varied between two and four. The multipath performance of the newly proposed technique along with the state-of-the-art DLLs, and other advanced techniques were presented via running average error sense and also via root-mean-square-error sense. Three different modulation types were considered including the newly proposed CBOC modulation (chosen as the modulation technique for Galileo E1 signal).

It was shown that the RSSML, in general, achieved the best multipath mitigation performance for all three different signals in this two-to-four paths closely spaced multipath profile. Simulation results show that the proposed RSSML offers a viable solution by increasing the position accuracy in the presence of closely spaced multipath, especially in dense urban areas where the number of significant paths can be higher than two. On the contrary, the proposed method increases the receiver complexity, since it is based on multicorrelator-based structure, and at the same time, it requires a good amount of memory to keep the reference noncoherent correlation functions available for computing the MMSE. Therefore, RSSML and other advanced multipath mitigation techniques presented here are more suitable for professional receivers due to their relatively high complexity.

Figure 8: RMSE versus $C/N_0$ plots for 2 to 4 paths Rayleigh fading channel in 24 MHz BW.
Table 2: Comparative performance of multipath mitigation techniques.

<table>
<thead>
<tr>
<th></th>
<th>nEML</th>
<th>HRC</th>
<th>TK + nEML</th>
<th>PT(Diff2)</th>
<th>TK</th>
<th>RSSML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closely spaced multipath performance</td>
<td>Moderate</td>
<td>Good</td>
<td>Moderate</td>
<td>Good</td>
<td>Moderate</td>
<td>Best</td>
</tr>
<tr>
<td>Running average error performance</td>
<td>Fair</td>
<td>Good</td>
<td>Good</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Best</td>
</tr>
<tr>
<td>Correlator requirement (No. of Corr.)</td>
<td>Few (3)</td>
<td>Few (5)</td>
<td>Few (5)</td>
<td>Many (100+)</td>
<td>Many (100+)</td>
<td>Many (100+)</td>
</tr>
<tr>
<td>A priori information</td>
<td>Coarse delay estimate</td>
<td>Coarse delay estimate</td>
<td>Coarse delay estimate</td>
<td>Coarse delay estimate</td>
<td>Coarse estimate delay</td>
<td>A large set of reference correlation functions</td>
</tr>
<tr>
<td>Channel estimation required?</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes (noise computation)</td>
<td>Yes (noise computation)</td>
<td>Yes (noise computation)</td>
</tr>
<tr>
<td>Memory requirement</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>High</td>
</tr>
<tr>
<td>Complexity</td>
<td>Low</td>
<td>Low</td>
<td>Moderate</td>
<td>Fair</td>
<td>Fair</td>
<td>High</td>
</tr>
</tbody>
</table>

Figure 9: RMSE versus C/N₀ plot for single path SinBOC(1,1) signal in 24 MHz BW.

whereas for mass-market receivers, nEML and HRC are still the best tradeoff between performance and complexity.

Acknowledgments

This work was carried out in the project “Future GNSS Applications and Techniques (FUGAT)” funded by the Finnish Funding Agency for Technology and Innovation (Tekes). This work has also been supported by the Academy of Finland, which is gratefully acknowledged. The authors would also like to thank Nokia Foundation and Tampere Doctoral Programme in Information Science and Engineering (TISE) for their financial support.

References


Publication P8


Copyright ©2011 GPS World. Reprinted with permission.