Dipl.-Ing. Philipp Berglez

# Development of a multi-frequency software-based GNSS receiver

Dissertation

For obtaining the academic degree of
Doktor der technischen Wissenschaften

Doctoral Programme of Technical Sciences
Doctoral School Geosciences, Class of Geodesy


**Institute of Navigation**
**Graz University of Technology**

**Supervisor**
**Univ.-Prof. Dr.h.c.mult. Dr. Bernhard Hofmann-Wellenhof**

**Examination Committee**
**Univ.-Prof. Dr.h.c.mult. Dr. Bernhard Hofmann-Wellenhof (TU Graz)**
**Univ.-Prof. Dr. Gernot Kubin (TU Graz)**
**A.o. Univ.-Prof. Dr.techn. Robert Weber (TU Vienna)**

Graz, February 2013

# Abstract

The steadily growing number of users of global navigation satellite systems (GNSS) demand a high level of accuracy and robustness from their receivers to meet the requirements of new applications. A single GNSS system for its own does not provide the desired performance for all applications. However, already in the near future, the combination of existing and upcoming systems, satellites, and signals will yield higher accuracy and robustness. Thus, a new level of quality and service will be achieved. In order to fully exploit these new evolution steps, more sophisticated receiver architectures have to be developed and deployed. In the recent past a concept has been elaborated where the hardware within a GNSS receiver is replaced as far as possible by software for the objective of cost- and time-effective product redesign. Up to now, the development of software-based GNSS receivers was focused on single frequency and single system solutions. The GNSS evolution programs, however, literally call for multi-frequency and multi-system implementations in software. The scientific challenges of the development and validation of a multi-frequency software-based GNSS receiver, capable of processing GPS and Galileo signals are elaborated within this thesis in a first step. Thereby, the focus is put on the receiver architecture and on signal processing tasks, i.e., acquisition and tracking, of different GNSS signals. Secondly, the scientific findings are used to explain the implementation of a software-based receiver using C/C++ programming language. The receiver is designed to use digital intermediate frequency signals as an input. This approach provides a high flexibility regarding the implementation of new signals and algorithms but at the same time allows to use reproducible scenarios what facilitates scientific research. Real-data recordings of GPS L2C, GPS L1, and GIOVE-A/B signals and simulated GPS and Galileo signals have been used to show the potential of the new receiver architecture. In a third step the problem of combining different systems and signals is addressed in order to implement new error mitigation and correction strategies and, thus, to obtain a more accurate and robust position solution. In this context the problem of combining different coordinate and time reference frames is explained and solved step-by-step. The algorithm for a combined processing of multi-system observations is analyzed and tested using GPS and Galileo signals. One of the major error sources influencing the position accuracy is the ionosphere. Different strategies, including a new one, for mitigating the impact of the ionospheric effect by using dual-frequency measurements are listed and based on the software-based receiver successfully tested. The combination of dual-frequency observations increases the measurement noise what can be evaluated using the reproducible scenarios of the developed receiver. The measurement noise is analyzed and a smoothing algorithm is proposed for reducing the noise and, thus, again enhancing the accuracy. Depending on the characteristics and on the conditioning

of the observations a weighting strategy has to be chosen to avoid systematic errors in the position computation. The scientific discussion and using the software receiver implementation raise the awareness of the correct strategy which is then validated using different test scenarios. This research study provides a deep insight into the development and validation of a multi-frequency software-based GNSS receiver. The knowledge is impart to build receiver architectures, modules for signal processing tasks, and algorithms for high position performance in software for optimized utilization of multi-GNSS signals.

# Zusammenfassung

Die stetig wachsende Anzahl von Nutzern globaler Satellitennavigationssysteme (GNSS) verlangt nach höhere Genauigkeit und Robustheit ihrer Empfänger, um die Anforderungen neuer Anwendungen erfüllen zu können. Ein einzelnes System bietet nicht die notwendige Leistungsfähigkeit für alle Anwendungen. In naher Zukunft aber eröffnet die wachsende Anzahl von Satellitensystemen und Signalen die Möglichkeit diese miteinander zu kombinieren, um dadurch eine höhere Genauigkeit und Robustheit zu erzielen und einen noch höheren Qualitäts- und Servicestandard zu erreichen. Bevor aber ein maximaler Profit aus diesen Neuerungen gezogen werden kann, müssen neue und komplexere Satellitennavigationsempfänger entwickelt werden. In den letzten Jahren entstand die Idee, die Aufgaben der Hardware innerhalb eines GNSS Empfängers so weit wie möglich in die Software zu verlagern, um damit leichter und schneller auf Veränderungen hinsichtlich neuer Systeme und Signale reagieren zu können. Dies erlaubt es zudem auch eine Produktneuentwicklung mit einem hohen Maß an Kosten- und Zeiteffektivität durchzuführen. Die Entwicklung von Software-basierten GNSS Empfängern hat sich in der Vergangenheit auf die Verwendung einer Frequenz und eines einzelnen Systems konzentriert. Die neuen Evolutionsprogramme der GNSS Systeme rufen buchstäblich nach Implementierungen in Software, die die Signale mehrerer Frequenzen und Systeme ausschöpfen. Die wissenschaftlichen Herausforderungen bei der Entwicklung und Validierung eines mehrfrequenz Software-basierten GNSS Empfängers, der GPS und Galileo Signale verarbeiten kann, werden in einem ersten Schritt in dieser Dissertation erarbeitet. Dabei wird spezielles Augenmerk auf die Architektur des Empfängers und die Signalprozessierung, d.h. Signalakquise und Signalverfolgung, unterschiedlicher GNSS Signale gelegt. Im zweiten Schritt werden die wissenschaftliche Ergebnisse genützt, um die Implementierung des Software-basierten Empfängers in der Programmiersprache C/C++ zu erläutern. Als Dateneingang dienen dem Empfänger digitale Zwischenfrequenz-Signale. Dieser Ansatz bietet eine hohe Flexibilität bei der Einbindung neuer Signale und Algorithmen, erlaubt es aber gleichzeitig reproduzierbare Szenarien durchlaufen zu lassen, was insbesondere die wissenschaftliche Forschung erleichtert. Aufgezeichnete Signale von GPS L2C, GPS L1 und GIOVE-A/B wie auch simulierte digitale GPS und Galileo Signale werden dazu verwendet das Potential der neuen Empfängerkonzepte zu zeigen. In einem dritten Schritt werden die Schwierigkeiten bei der Kombination unterschiedlicher Systeme und Signale adressiert, um in weiterer Folge neue Strategien für Fehlerverminderung und Fehlerkorrektur implementieren zu können. Dadurch können Positionslösungen mit höherer Genauigkeit und Robustheit erzielt werden. Im Rahmen der Diskussion zur Kombination von Systemen werden die Herausforderungen von unterschiedlichen Koordinaten- und Zeitreferenzsysteme erklärt und Schritt für Schritt gelöst. Die

Algorithmen zur Positionsberechnung auf Basis von Mehr-Systemmessungen werden mittels GPS und Galileo Signalen analysiert und getestet. Einer der größten Fehleranteile bei der Positionslösung liegt im Einfluss der Ionosphäre. In dieser Arbeit werden verschiedene Strategien zur Reduktion des ionosphärischen Einflusses auf Basis der Verwendung von Zweifrequenz-Messungen diskutiert. Ein neuer Ansatz für eine kombinierte Behandlung von Mehrfrequenz- und Einfrequenz-Messungen wird präsentiert und mittels dem Software-basierten Empfänger getestet. Durch die Kombination mehrerer Frequenzen erhöht sich das Rauschen in den Messungen. Dies kann mittels der reproduzierbaren Szenarien und Signale mit dem Software-basierten Empfänger einfach nachgewiesen werden. Dieser Effekt, der im Detail beschrieben wird, kann durch die Verwendung einer Codeglättung mittels Phasenbeobachtungen reduziert werden, wodurch eine Verbesserung der Positionsgenauigkeit erzielt wird. Durch die Kombination unterschiedlicher Messungen mehrerer Signale und ihrer unterschiedlichen Konditionierung muss eine adäquate Strategie für die Gewichtung der Messungen gewählt werden, um systematische Fehler in der Positionsberechnung zu vermeiden. Durch die wissenschaftliche Betrachtung und durch den Einsatz des Software-basierten Empfängers kann eine optimale Strategie für die Gewichtung identifiziert werden, die anschließend mittels unterschiedlicher Szenarien auch validiert werden kann. Im Rahmen dieser Arbeit wird die Entwicklung und Analyse eines mehrfrequenz Software-basierten GNSS Empfänger präsentiert. Es wird das Wissen vermittelt wie Empfängerarchitekturen, Module für die Signalprozessierung und Algorithmen für Positionen hoher Genauigkeit und Robustheit in Software umgesetzt werden können, um die Signale mehrere GNSS Systeme optimale nützen zu können.

# Contents

# Acknowledgments

Finishing this thesis would not have been possible without guidance and help. Therefore I would like to express my gratitude to several people who contributed to this thesis by supporting and motivating me during the last years.

First of all, I would like to thank my supervisor Prof. Bernhard Hofmann-Wellenhof. He encouraged me to base this work on several of TeleConsult Austria's projects, believed in my capabilities, and provided me with all the support and flexibility which helped me to finish this thesis.

I would like to thank Prof. Gernot Kubin and Prof. Robert Weber for reviewing my thesis as advisors and providing valuable comments and suggestions.

I am very thankful to Elmar Wasle, Franz Weimann, and Andreas Pedroß for carefully reading the first version of this thesis and for their numerous discussions, remarks, and suggestions.

I have to thank my colleagues at TeleConsult Austria GmbH and the Institute of Navigation for several fruitful discussions, especially to Bernadette, Christoph, and Florian for sharing a good time at the job and their friendship.

I would like to thank Katrin for her encouragement and support during the last years, and showing me that there is more than work.

Last but not least, I would like to express my deepest gratitude to my family for giving me the opportunity to complete my academic studies, for their great support over all the past years, and their love.

Graz, February 2013 Philipp Berglez

# Abbreviations

| | |
|---|---|
| ACF | autocorrelation function |
| A/D | analog to digital |
| ADC | analog-to-digital converter |
| AGC | automatic gain control |
| AGNSS | assisted (or aided) GNSS |
| AltBOC | alternative binary offset carrier |
| BER | bit error ratio |
| BOC | binary offset carrier |
| BPS | bits per second |
| BPSK | binary phase-shift keying |
| C/A | coarse/acquisition |
| CAF | cross ambiguity function |
| CCIR | Comité Consultatif International des Radiocommunications |
| CBOC | composite binary offset carrier |
| CCF | crosscorrelation function |
| CCS | crosscorrelation sequence |
| CDMA | code division multiple access |
| C/NAV | civil navigation message |
| $C/N_0$ | carrier-to-noise-power-density ratio |
| CODE | Centre for Orbit Determination in Europe |
| CPU | central processing unit |
| CRC | cyclic redundancy check |
| CS | commercial service |
| CPPR | correlation power peak ratio |
| DCB | differential correction bias |
| DFT | discrete Fourier transform |
| DLL | delay locked loop |
| DoD | Department of Defense |
| DOP | dilution of precision |
| DoT | Department of Transportation |
| DSP | digital signal processing |
| DSSS | direct sequence spread spectrum |
| ECEF | earth-centered, earth-fixed (coordinates) |
| EGNOS | European geostationary navigation overlay service |
| ESA | European Space Agency |
| EU | European Union |
| FDMA | frequency division multiple access |

| | |
|---|---|
| FEC | forward error correction |
| FFT | fast Fourier transform |
| FFTW | fastest Fourier transform in the west (C subroutine library) |
| FhG-IIS | Fraunhofer Gesellschaft – Institute of Integrated Circuits |
| FLL | frequency locked loop |
| F/NAV | freely accessible navigation message |
| FOC | full operational capability |
| FPGA | field programmable gate array |
| GBAS | ground-based augmentation system |
| GDOP | geometric dilution of precision |
| GGTO | GPS to Galileo time offset |
| GIPSIE® | GNSS multisystem performance simulation environment |
| GIPSIE®–IFS | GIPSIE® – intermediate frequency simulator |
| GIPSIE®–SCS | GIPSIE® – satellite constellation simulator |
| GIOVE | Galileo in-orbit validation element |
| GLONASS | Global Navigation Satellite System (Global'naya Navigatsionnaya Sputnikovaya Sistema) |
| GNSS | global navigation satellite system |
| GNU | GNUs not Unix |
| GPL | GNU general public license |
| GPS | Global Positioning System |
| GPST | GPS system time |
| GST | Galileo system time |
| GTRF | Galileo terrestrial reference frame |
| HDL | hardware description language |
| HOW | hand-over word |
| I&D | integrate and dump |
| IDFT | inverse discrete Fourier transform |
| IF | intermediate frequency |
| I/NAV | integrity navigation message |
| IOC | initial operational capability |
| IOV | in-orbit validation |
| ISC | inter-signal correction |
| ITRF | international terrestrial reference frame |
| ITRS | international terrestrial reference system |
| ITU | International Telecommunication Union |
| ITU-R | ITU, Radiocommunication (sector) |
| KML | keyhole markup language |
| LC | linear combination |
| LFSR | linear feedback shift register |
| LNA | low-noise amplifier |
| LO | local oscillator |
| MBOC | multiplexed binary offset carrier |
| Mcps | Mega chips per second |

| | |
|---|---|
| MEO | medium earth orbit (satellite) |
| MGD | multiple gate delay |
| NAV | navigation message |
| NAVSTAR | navigation system with timing and ranging |
| NCO | numerically controlled oscillator |
| NMEA | National Marine Electronics Association |
| OS | open service |
| PC | personal computer |
| PDF | probability density function |
| PDOP | position dilution of precision |
| PLL | phase locked loop |
| PPS | precise positioning service |
| PRN | pseudorandom noise |
| PRS | public regulated service |
| PSD | power spectral density |
| PSK | phase-shift keying |
| PVT | position, velocity, and time |
| QPSK | quadrature phase-shift keying |
| RAM | random access memory |
| RF | radio frequency |
| RINEX | receiver independent exchange format |
| RNSS | radionavigation satellite system |
| ROC | receiver operating characteristic |
| RTK | real-time kinematics |
| SAR | search and rescue |
| SBAS | space-based augmentation system |
| SDR | software-defined radio |
| S/N | signal-to-noise ratio |
| SoL | safety of life |
| SPS | standard positioning service |
| SSB | single sideband |
| SV | space vehicle |
| SVID | space vehicle identification |
| TCA | TeleConsult Austria GmbH |
| TDOP | time dilution of precision |
| TEC | total electron content |
| TGD | time group delay |
| TLM | telemetry word |
| TOW | time of week |
| US | United States (of America) |
| USRP | universal software radio peripheral |
| UTM | universal transverse Mercator |
| WAAS | wide-area augmentation system |
| WGS-84 | World Geodetic System 1984 |
| XOR | exclusive-or |

# Chapter 1

# Introduction

This dissertation deals with the development of a software-based multi-frequency global navigation satellite system (GNSS) receiver. The term GNSS denotes navigation systems based on a global constellation of satellites, which emit ranging signals used for positioning. The steadily growing segment of GNSS users requires higher accuracy and robustness from their receivers to fulfill the requirements of new applications. One system for its own cannot provide the level of performance for all applications, particular those demanding high position accuracy and integrity information. The strategy in order to meet the new requirements is to modernize current GNSS and to develop new systems. The modernization of the existing United States (US) Global Positioning System (GPS) and Russian Global Navigation Satellite System (GLONASS), the evolution of new systems like the European Galileo or the Chinese Beidou, and the availability of space-based augmentation systems (SBASs) or ground-based augmentation systems (GBASs) bring numerous systems and signal players into the field of navigation. In the near future the increasing number of satellites and signals give rise to the possibility of combining various constellations and signals to provide a new level of services. Due to this rapid evolution, GNSS receiver architectures have to be adapted to the new possibilities and, thus, the receiver development has rather started than finished.

## 1.1 Motivation

The traditional satellite navigation receiver architecture is based on special hardware chips. The major drawback is that accounting for new specifications is difficult and results most of the time in a redesign of the hardware. The idea arose to replace the tasks of the hardware as far as possible by software for the purpose of easy and especially cost-saving redesign. Due to the required high computational effort for processing GNSS signals, the idea became feasible not before the first powerful microprocessors appeared. The first software-based receivers were implemented about one decade ago. Nowadays the developers try to use the increased computing power, especially the increased computational performance in cell phones or hand held devices, for signal processing with the advantage of producing less expensive GNSS receivers. Furthermore software-based receivers are especially useful for research and development activities. The implementation and testing of new algorithms

and strategies becomes easier and saves time, especially considering signal processing models and methods. Advanced algorithms are designed to enhance the performance which will open new fields of applications. A software-based receiver offers the possibility to access all different processes within a receiver. Thus, the integration with other complementary navigation systems can be easily achieved. For a deep integration with another navigation system, for example the use of raw GNSS observation data is necessary and in return the data of the complementary navigation system are used to steer the algorithms of the GNSS receiver to improve its performance. Since conventional hardware-based receivers are not suitable for this kind of integration, software-based receivers can fill this gap and can be used easily instead. Future innovation in receiver design, in navigation sensor integration techniques, and in meeting application requirements will be directly linked to the software receiver architecture (Morton 2007). Software-based receivers will also become more important on the global market due to cost optimization and miniaturization. The development is just at the beginning and will evolve rapidly in the future and, furthermore, will be part of the general market evolution.
The development of software-based GNSS receivers in the past concentrated on single frequency and single system solutions. Hardly any dual-frequency and dual-system and/or multi-frequency implementations have been developed so far. By using more than one GNSS and more than one frequency an improvement of accuracy and integrity becomes feasible and, thus, the range of applications is widened. An improvement of accuracy, due to the reduction of ionospheric errors, is achieved when processing multi-frequency measurements of one system. The availability and integrity is increased by using two different systems. These enhancements are needed, for example, within safety critical applications on land, on sea, in the air, or in space.
This thesis presents the development and the analysis of a multi-frequency software-based GNSS receiver, capable of processing two GPS and one Galileo frequencies. The performed research covers the following topics:

- Software-based signal processing approach in order to reduce the hardware components;
- Acquisition and tracking methods for processing different GNSS signals;
- Ionosphere-free pseudorange combination to improve the position accuracy;
- Combined GPS and Galileo position solution;
- Use of simulated and real-world data for verification and performance analysis.

The current implementation of GPS and Galileo marks a first step and builds the basis for future work. Besides the further adaptation of receiving other GNSS signals, the software-based receiver platform serves as an ideal scientific playground for the investigation of signal processing techniques and, furthermore, for the integration with other navigation sensors.

## 1.2 Thesis outline

Chapter 1 states the motivation for the thesis and explains the structure of it. A state-of-the-art analysis shows the current software-based receiver market and highlights the innovative elements and the related work of this thesis.
Chapter 2 starts with a short introduction to GNSS signals and systems. The different signal structures of GPS and Galileo, as well as their navigation data are described. Furthermore the different pseudorandom noise (PRN) sequences and their correlation properties are discussed in detail. Special focus is put on the civil and open-service signals of GPS and Galileo.
Chapter 3 gives an overview of the software-based receiver structure and a short introduction to the GNSS front-end module, which affects the sampled raw data characteristics. The basic modules and their interrelation within the developed receiver are discussed briefly.
Chapter 4 describes the GNSS signal acquisition in detail. The implementation of the parallel code phase search acquisition method is described in more detail. The zero-padding strategy, needed for the acquisition of several civil signals, is discussed. Acquisition results of different systems and signals show the performance of this module.
Chapter 5 deals with code and carrier tracking and the data demodulation. Different strategies for tracking GPS and Galileo signals are described. In case of Galileo signal tracking the multiple gate discriminator technique is investigated. The performance of the developed module is discussed by showing selected results of simulated and real-world data.
Chapter 6 discusses the computation of the raw observables and the recovery of the ephemerides data. The topics of cyclic redundancy check (CRC), forward error correction (FEC), and block de-interleaving are also described in detail.
Chapter 7 provides information about the positioning algorithm for a combined GPS and Galileo receiver. The problem of combining different systems and signals is addressed in order to implement new error mitigation and correction strategies and, thus, to obtain a more accurate and robust position solution. In this context the problem of combining different coordinate and time reference frames is explained and solved step-by-step. Algorithms for processing dual-frequency measurements to eliminate the ionospheric error are presented.
Chapter 8 shows different test cases and results using both simulated and real-world data. Real-data recordings of GPS L2C, GPS L1, and GIOVE-A/B signals and simulated GPS and Galileo signals are used to show the potential of the new receiver architecture. Special focus lies on the performance evaluation of the dual-frequency measurements. Finally, the execution time and processing load are discussed.
Chapter 9 concludes this thesis and provides an outlook to future topics.

## 1.3 State-of-the-art

Many different definitions of a software-defined radio (SDR) can be found in literature, e.g., in Buracchini (2000) or Mitola (1995). Following Won et al. (2006) the widely accepted SDR definition in the field of GNSS is a receiver in which internal digital signal processing

is carried out by a programmable processor applying software techniques. According to Petovello and Lachapelle (2008) the term SDR is used in a more general sense, whereas the term software-based is referred to those receivers which carry out all the internal signal processing by software techniques (Won et al. 2006). Three different main SDR categories exist (Won et al. 2006), as shown in Figure 1.1.

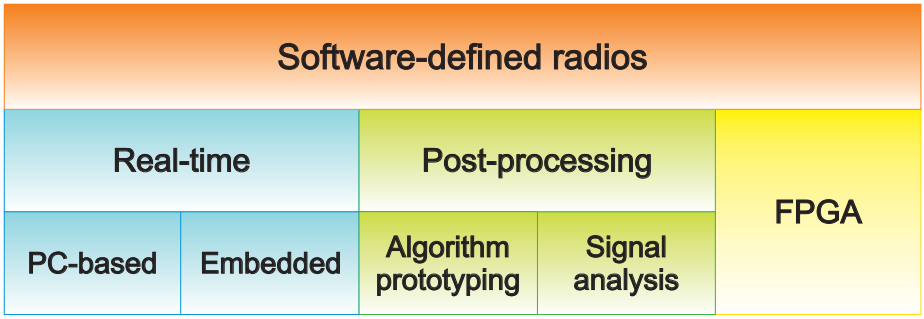



Figure 1.1: Software-defined radio categories

The majority of software-based receivers is located in the subgroup algorithm prototyping and is thus designed for post-processing. Testing and validating newly developed algorithms using real or realistic signals is often done by small software tools or some lines of source code. Although most of them do not cover the whole receiver operations and are far from real-time capabilities, some of them might be denoted as software-based receivers. Another field of application of a post-processing software receiver is GNSS signal analysis, where new signals are studied, unpublished codes decrypted, or satellite failures investigated. However, the GNSS SDR boom really started with the development of real-time processing capabilities. At the beginning this was possible using special digital signal processing (DSP) tools and later using conventional personal computers (PCs). Today the trend is towards specialized processors for embedded applications. The hardware environments (processor speed, memory, available hard disk space) vary from PC-based to embedded systems and so does the overall software design. Most of the PC approaches use C++ or even MATLAB as development language, while in the embedded sector C or assembler is used. Due to limited processing capabilities high-end multi-frequency receivers will have a PC-based approach or can be found in the third category. The term FPGA refers to an integrated circuit which is designed to be configured by the customer or designer after manufacturing. The FPGA is generally programmed using a hardware description language (HDL). Sometimes field programmable gate array (FPGA) based receivers can be programmed in the field in a C-like language and are used in real-time multi-frequency GNSS receiver developments (Overbeck et al. 2010). More discussion on the different groups can be found in Won et al. (2006).
In the following, existing implementations of software-defined GNSS receivers are presented. Note that some terminologies will be used which will be explained later in this thesis. The list is in alphabetical order of products and manufactures, without being exhaustive, but covering the most important players in the market.

- The Aerospace Research and Technology Centre (CTAE) is providing an open source receiver, called GPS-SDR, based on universal software radio peripheral (USRP) and

SiGe Semiconductor components. It is able to process GPS L1 and Galileo E1 signals. The C++ source code is available at *http://www.ctae.org/sdr/*.

- Baseband Technologies offers a GPS L1 C/A-code PC-based or embedded-based software receiver (*http://www.basebandtech.com/images/Technical Paper - Next Generation Software-based GPS Receiver for Real-World Applications.pdf*).

- A GPS software receiver toolkit is provided by the Data Fusion Corporation. The toolkit is available as MATLAB toolkit, as C toolkit, or as DSP/C toolkit and provides GPS L1 processing capabilities (*http://www.datafusion.com/gps/brochure.pdf*).

- The fastgps project is another open source project featuring GPS L1 C/A signals and is available at SourceForge (*http://fastgps.sourceforge.net*).

- Fastrax Software GPS was developed for high-volume consumer devices (e.g., personal navigation devices, mobile internal devices, smart phones, or laptops). According to *http://www.fastraxgps.com* it is able to process GPS L1 signals.

- GPS Laboratory at Cornell University developed a GPS L1/L2 software receiver written in MATLAB, called GPSL (*http://gps.ece.cornell.edu/SFRCX.html*).

- The University of Catalunya has developed a GNSS open source software receiver, capable of processing GPS L1 C/A signals. The receiver is FPGA based and uses an open source software-defined radio platform (Universal Software Radio Platform) for the signal processing. The GNU general public license (GPL) radio project library configures the platform and the data processing is done by the GPS-SDR open source project software (*http://www.cttc.es/resources/doc/090608-gnssdemo-64145.pdf*).

- Regarding observable processing and data management the GPS Toolkit (GPSTk) from the University of Texas has to be mentioned. GPSTk is an open source project providing comprehensive libraries and sets of utilities for GPS research, analysis, and development. The GPSTk suite is a platform-independent software written in C++ able to process GPS L1 and L2 raw observations (*http://www.gpstk.org*).

- The University FAF Munich developed a software receiver completely in software (Visual C++ / assembler). The ipexSR (Institute of Geodesy and Navigation PC-based Experimental Software Receiver) is capable of processing multi-frequency GNSS signals in real-time or post-processing. The FPGA-based receiver is capable of processing GPS L1/L2/L5, Galileo E1/E5/E6 as well as GLONASS G1/G2 signals and can operate in real-time kinematics (RTK) mode (*http://www.ifen.unibw-muenchen.de/news/ipexSRInfo.pdf*).

- NAMURU II GPS receiver is an FPGA based receiver capable of processing GPS L1 and L2 signals (*http://www.dynamics.co.nz/index.php?main page=page&id=11*).

- NGene is a full software receiver developed by the Instituto Superiore Mario Boella (ISMB) and Politecnico di Torino. It is able to process GPS L1, EGNOS L1, and Galileo E1 signals in real-time (Fantino et al. 2009).

- An open source software (OpenSourceGPS) is provided by Clifford Kelley for processing GPS L1 signals using an embedded system for signal processing (*http://www.home.earthlink.net/∼cwkelley/*).

- An open architecture software GNSS receiver has been written in C and C++ by the Center for Remote Sensing, Inc. (CRS). The receiver components are based on CRS's Windows-based IMPULSE software and can be ported to DSP, FPGA, or single chip for real-time applications. Two different versions exist. While SGS-CA-001-01 is able to process GPS L1 C/A, SGS-DF-002-01 is a dual-frequency P-code receiver (*http://www.cfrsi.com/products/gpssystemsoftrx.html*).

- The Position, Location And Navigation (PLAN) Group of the University of Calgary developed the SNRx (GNSS Software Navigation Receiver). It is a C++ class-based software receiver and utilizes GPS L1 C/A, L2C, L5 and L1C signals, Galileo E1B, E1C, E5a-I, E5a-Q, E5b-I and E5b-Q signals, and GLONASS G1 and G2 civil signals. It is designed for any available co-processors (e.g., FPGA, DSP) (*http://plan.geomatics.ucalgary.ca/software gsnrx.php*).

- Sogei offers simulation software modules in C language for acquisition, tracking and positioning of GPS, Galileo, and EGNOS L1 signals (*http://www.sogei.it/flex/cm/pages/ServeBLOB.php/L/EN/IDPagina/202*).

- The Danish GPS Center offers a software-defined receiver (softGPS project) capable of processing the GPS L1 C/A signal in post-processing mode. The complete receiver source code is written in MATLAB and is part of a book by Borre et al. (2007) (*http://kom.aau.dk/project/softgps/*).

- Based on the Danish GPS Center source code a software-defined GPS/GLONASS receiver is available at *http://gnss-sdr.ru*. The source code of the receiver is written in SCILAB, which is an open-source analogue of MATLAB.

- Intecs S.p.A offers both: a real-time software receiver and a software digital signal analysis tool. The first tool (Softrec) supports GPS and EGNOS L1 signals, the second one, named SoftrecG3, supports all GNSS navigation signals (*http://www.intecs.it/eng/prodotti dettagli.asp?IDProdotto=12*).

- SPIRIT GLONASS + GPS software is designed to run on different platforms (e.g., ARM Cortex, TI C64, or Intel Atom processor) offering a combined GPS/GLONASS L1 position solution (*http://www.spiritdsp.com/products/super-sensitive-software-gnss-receiver.php*).

- The SX-NSR software receiver developed by IFEN GmbH is able to process GPS L1/L2/L5, GLONASS G1/G2, Galileo E1/E5/E6, and SBAS L1 signals in real-time (*http://www.ifen.com/sx-snr*).

Table 1.1 summarizes the main characteristics of the existing software-defined GNSS receivers and compares them with the software-based GNSS receiver presented in this thesis.

Table 1.1: Existing software-defined GNSS receivers

| Name | Processing | | Platform | | System | | | Frequency | |
|---|---|---|---|---|---|---|---|---|---|
| | real-time | post-processing | PC | Other | GPS | GLONASS | Galileo | Single | Multi |
| Baseband Technology | x | | x | | x | | | x | |
| Cornell University | x | | x | x | x | | | | x |
| CTAE | x | | | x | x | | x | x | |
| Data Fusion Corporation | | x | x | | x | | | x | |
| FastGPS | | x | x | | x | | | x | |
| Fastrax | x | | x | x | x | | | x | |
| GPS Toolkit | | x | x | | x | | | | x |
| NAMURU II GPS | x | | | x | x | | | | x |
| NGene | x | | x | | x | | x | x | |
| OpenSourceGPS | x | | x | | x | | | x | |
| ipexSR | x | | | x | x | x | x | | x |
| IMPULSE | x | | | x | x | | | | x (P-Code) |
| Sogei | | x | x | | x | x | | x | |
| Danish GPS Center | | x | x | | x | | | x | |
| gnss-sdr.ru | | x | x | | x | x | | x | |
| Intecs S.p.A | x | | x | x | x | x | x | | x |
| SPIRIT GLONASS + GPS | x | | | x | x | x | | x | |
| SX-NSR | x | | x | | x | x | x | | x |
| University of Calgary | | | | x | x | x | x | | x |
| University Catalunya | | | | x | x | | | x | |
| Developed receiver | | x | x | | x | | x | | x |

## 1.4 Innovative elements

The innovative elements of this thesis are based on technologies and algorithms of signal processing and satellite navigation, which are tailored to the task of processing open-service GPS and Galileo signals in order to obtain a combined dual-frequency GPS and Galileo user position. As shown in Sect.1.3 and Table 1.1 most of the existing software-based receivers are GPS single-frequency receivers. Only some of them are capable of processing more than one frequency and more than one system. The developed receiver presented in this thesis is capable of processing multi-system and multi-frequency signals in order to enhance availability, accuracy, and integrity. The signal processing tasks have been implemented in software using C++ language. The developed receiver is capable of processing GPS L1 C/A and L2C signals, as well as Galileo E1B signals. The signal acquisition has been optimized to the parallel code-phase acquisition method. The necessity of using a zero padding algorithm has been investigated and tested. The signal acquisition has been tested using the available Galileo in-orbit validation element signals. The difficulties in tracking different GNSS signals have been analyzed. Therefore several tracking methods, using different settings, have been compared in performance. The influence of the ionosphere on the receiver position accuracy has been examined and different approaches for reducing the ionospheric errors are presented. The investigations focus on the use of ionosphere models, as well as on the use of dual-frequency observations. Two ionospheric models (Klobuchar and NeQuick) have been implemented and compared. Concerning the dual-frequency observations three different methods are presented and compared, analyzing their advantages and disadvantages. Referring to the combination of GPS and Galileo the issue of the different reference times has been discussed and two different methods have been compared. An innovative scheme for weighting the different observations has been elaborated and is presented. Recordings of real-world and simulated data are used to evaluate the performance of the receiver developed. This thesis describes an independent realization of a software-based receiver, able to handle measurements of two different frequencies of two different GNSS, which has been used to explore new fields of research, whose results are presented, and which provides a tool for further research work in the field of satellite navigation.

## 1.5 Related work and publications

The position, velocity, and time computation has been developed by the author of this thesis in the context of his employment at TeleConsult Austria GmbH (TCA) within the GAMMA (Assisted Galileo/GPS/EGNOS Mass Market Receiver) and GAMMA-A (Galileo Receiver for Mass Market Applications in the Automotive Area) projects (*www.gamma-project.info*). The research and development projects GAMMA as well as GAMMA-A have been managed by the European GNSS Agency (GSA) and have been co-funded within the 6th and 7th Framework Programme of the European Union. The extension regarding the GPS L2C capability has been performed by the author within the SoftGNSS 2 (Dual-frequency software-based GNSS receiver) project. The research and development project SoftGNSS 2 has been managed by the Austrian Research Promotion Agency (FFG) and

has been co-funded through the Austrian Federal Ministry for Transport, Innovation, and Technology (BMVIT). The part concerning signal processing was developed independently from the projects mentioned above, although investigations have been done within the SoftGNSS 2 project by the Signal Processing and Speech Communication Laboratory at Graz University of Technology in parallel. Parts of this thesis have been presented earlier at various conferences, meetings, and magazines in particular Berglez et al. (2009), Berglez et al. (2010), Overbeck et al. (2010), Rügamer et al. (2010), Huber et al. (2011), and Berglez (2008).

# Chapter 2

# GNSS signals and systems

## 2.1 Introduction

Within the last two decades the term GNSS was used as a synonym for the US navigation system with timing and ranging (NAVSTAR) Global Positioning System (GPS). The GPS concept dates back to the early 1970's and was developed for the military forces as an all-weather, space-based navigation system to allow the accurate determination of position, velocity, and time (Wooden 1985). The design – originally for military use only – was then changed to a dual-use system to serve both, military and civilian applications. Since that decision the number of civil users and applications increased constantly. For many applications, particular those demanding high position accuracy and integrity (alerting the user in case of malfunctioning), one system alone cannot provide the required level of performance. The GPS was not the only GNSS available, but the Russian counterpart GLONASS lacked over years an adequate number of satellites for reliable applications (Cojocaru et al. 2009).
The modernization of current GNSS (i.e., GPS and GLONASS) and also the development of new systems were mainly driven by new increased user requirements. Thus, in the 1990's, the Department of Defense (DoD) in cooperation with the Department of Transportation (DoT) started a GPS modernization process, denoted as GPS II and GPS III (McDonald 2002). In 1994, the European Union (EU) decided to launch a two-step program. The first step – denoted as European's contribution to GNSS-1 – is the European geostationary navigation overlay service (EGNOS) which enhances the accuracy and integrity of GPS within Europe. The second (GNSS-2) is the deployment of an independent GNSS, known as Galileo. Also the Russian government decided to re-activate its GLONASS program (Bartenev. et al. 2006) and to modernize it. Besides these, China currently develops and deploys a global navigation satellite system called Beidou/Compass. Other countries are planning activities in this field as well. In addition, these systems are enhanced by SBAS and GBAS.
From the technological point of view the modernized and new systems encompass new generations of satellites, new technologies in ground stations, new software routines but most importantly also new signals in various frequency bands. Thus, the improvements affect all levels of GNSS: the space segment, the user segment, and the ground segment. More details on the different systems and the modernization is given in Hofmann-Wellenhof et al. (2008) or Kaplan (2006). The modernization program of the existing GNSS, together with the deployment of new GNSS and regional systems, including SBAS, brings numerous systems and signal players into the field of navigation. Thus, a lot of radio navigation signals are currently, and will in the future too, be transmitted from space to earth. Since

this thesis focuses on the civil GPS and Galileo signals and their combination, only the signal structures of importance, including their main characteristics, are presented here; more thorough discussions of general GNSS operations and applications can be found in Hofmann-Wellenhof et al. (2008), Misra and Enge (2006), or Gleason and Gebre-Egziabher (2009). All systems in common is that they utilize broadcast signals, emitted from orbiting satellites and propagating through the atmosphere for time of arrival ranging. Satellite signal transmissions are based on highly accurate atomic frequency standards.

## 2.2 Characterization of GNSS signals

GNSS satellite signals are referred to as direct sequence spread spectrum (DSSS) signals. In telecommunications in general, spreading the spectrum of a signal is a technique where a signal is transmitted using a bandwidth larger than required to contain the payload (Ward et al. 2006a). Following Hofmann-Wellenhof et al. (2008), such a signal can be described using a three-layer model, as shown in Figure 2.1.

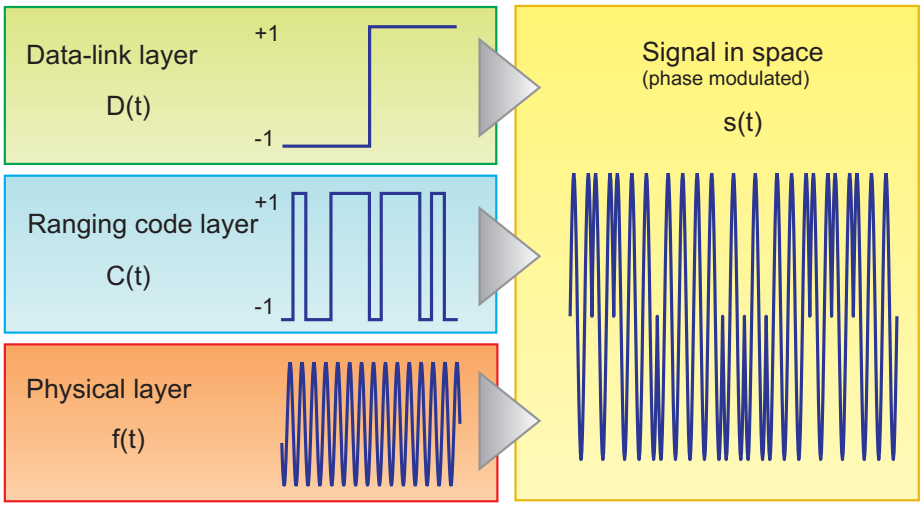



Figure 2.1: Composition of a GNSS signal

The signal is composed of a physical layer (the carrier wave), a ranging code layer, and a data message layer. The physical layer characterizes the physical properties of the transmitted signals (e.g., frequency, power, etc.). The ranging code layer consists of periodic noise-like binary sequences of rectangular pulses. These sequences $C[k] \in \{-1, +1\}$ are so-called pseudorandom noise (PRN) codes and have spectral properties similar to random binary sequences but are actually deterministic. The data-link layer, also denoted as navigation data, consists of a sequence of bits $D[m] \in \{-1, +1\}$. This bit sequence contains information about satellite orbits, satellite constellation status, correction data, and other necessary data. The PRN code as well as the navigation data are then modulated onto the physical layer. By choosing $D[m]$ and $C[k] \in \{-1, +1\}$ a binary phase-shift keying (BPSK) modulation is achieved. Note that the PRN code has a much higher symbol rate compared to the navigation data and thus spreads the signal across a wide bandwidth. Retrieving the information can be done by 'de-spreading' the spectrum, by correlating the incoming

signal with the same locally generated PRN sequence. This layer structure is inherent to all GNSS signals, but they use different types of ranging codes, data messages, and carrier frequencies.
Several reasons for using DSSS signals for satellite navigation exist (Kaplan 2006). For example, the high symbol rate of the PRN code enables precise ranging by the receiver using the good correlation properties. Further, DSSS signals provide a certain level of resistance to both unintentional and intentional interference, as well as a certain degree of multipath resistance. The use of different PRN sequences, transmitted on the same frequency, enables multiple satellites to transmit signals simultaneously. This method, transmitting multiple DSSS signals with different spreading sequences on a common carrier frequency, is called code division multiple access (CDMA). For the diversification of the satellite signals GPS and Galileo use CDMA, meaning that different codes of different satellites are transmitted on the same carrier frequency. GLONASS in contrast uses frequency division multiple access (FDMA). Thereby, the same code is transmitted on different frequencies, for signal separation. Since CDMA and FDMA require different signal processing techniques, the next generation of GLONASS satellite signal components will use CDMA techniques for compatibility reasons to other GNSS. This will also reduce the hardware complexity for combined GPS, Galileo, GLONASS receivers.

### 2.2.1 Signals and data

The emitted satellite signal $s(t)$ can be expressed mathematically as

$$s(t) = \sqrt{2P} \sum_{m=-\infty}^{\infty} \sum_{k=0}^{N_C-1} D[m]C[k] \operatorname{rect}_{T_C}(t - mT_D - kT_C) \cos(2\pi f t), \tag{2.1}$$

where $P$ denotes the signal power, $D[m]$ and $C[k]$ the data message sequence and the ranging code sequence respectively, $T_C$ and $T_D$ the chip duration and the data bit duration, $f$ is the (linear) frequency, and $t$ is the time parameter. The function $\operatorname{rect}_{T_C}(t)$ represents the rectangular pulse of duration $T_C$ and is defined as

$$\operatorname{rect}_{T_C}(t) = \begin{cases} 1 & \text{for } t \in [0,\ T_C] \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

For the sake of simplicity, the signal, defined in (2.1), will be represented in the following chapters as (Hofmann-Wellenhof et al. 2008)

$$s(t) = \sqrt{2P} D(t) C(t) \cos(2\pi f t). \tag{2.3}$$

In this case $C(t)$ is given by

$$C(t) = \sum_{k=0}^{N_C-1} C[k] \operatorname{rect}_{T_C}(t - kT_C), \tag{2.4}$$

where $N_C$ represents the PRN code length. Analogous $D(t)$ is given by

$$D(t) = \sum_{m=-\infty}^{\infty} D[m] \operatorname{rect}_{T_D}(t - mT_D). \tag{2.5}$$

The relationship between code chip duration $T_C$, the code length $N_C$, and the data bit duration $T_D$ is given by

$$T_D = N_P N_C T_C, \tag{2.6}$$

where $N_P$ defines the integer number of code epochs per data bit. The most common modulation used in GNSS is binary phase-shift keying (BPSK). Some of the upcoming new GNSS signals use binary offset carrier (BOC) modulation. The differences between BPSK and BOC as well as the advantages and disadvantages are discussed in Sect. 2.2.2. Commonly more than one PRN sequence and data message are transmitted on one carrier frequency signal by modulating the code sequences on the carrier using an in-phase $I$ and a quadrature-phase $Q$ component. The IQ-modulated (or quadrature modulated) signal can be written as

$$s(t) = \sqrt{2P_I} D_I(t) C_I(t) \cos(2\pi f t) - \sqrt{2P_Q} D_Q(t) C_Q(t) \sin(2\pi f t), \tag{2.7}$$

using two different power levels ($P_I$, $P_Q$), data messages ($D_I$, $D_Q$), and PRN codes ($C_I$, $C_Q$) and is known as quadrature phase-shift keying (QPSK). It is used for example in the case of GPS L1, combining C/A- and P-code. As mentioned in Sect. 2.2 the GNSS signal design is very important to meet the application requirements. Beside the correlation properties of the PRN code, the signal power distribution within the designated frequency band is important too. The two signal properties of most importance to describe these characteristics are the crosscorrelation function and the power spectral density (PSD). The crosscorrelation function is defined as

$$R(\tau) = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} s_1(t) s_2(t+\tau) dt. \tag{2.8}$$

It describes the degree of correspondence of two signals $s_1(t)$ and $s_2(t)$ as a function of time shift $\tau$ between them. A correlation coefficient $R(\tau) = 0$ represents perfect orthogonality of signals. In case of $s_1(t) = s_2(t)$, (2.8) represents the autocorrelation function (ACF). At zero lag, $\tau = 0$, the ACF has a maximum. The important correlation properties of a DSSS signal using a PRN binary code are (Hofmann-Wellenhof et al. 2008):

- maximum correlation at zero lag;
- uncorrelated at $\tau \neq 0$;
- uncorrelated with any other PRN sequence.

The orthogonality of the signals and good correlation properties are needed to isolate one satellite signal from the other and to minimize interference between the signals. Note that PRN codes are finite sequences and not continuous functions, thus, the crosscorrelation between two sequences $C_1$ and $C_2$, both with length $N$, is calculated by

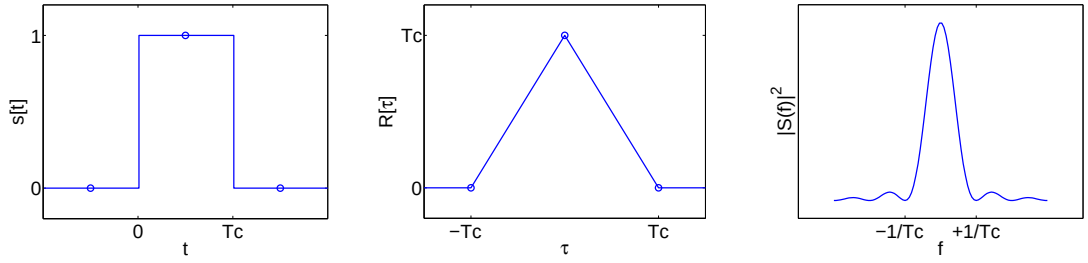
$$R[m] = \frac{1}{N} \sum_{n=0}^{N-1} C_1[n] C_2[n+m]. \tag{2.9}$$

In case of $n+m \geq N$ the sequence $C_2[n+m]$ is shifted according to $(n+m)$ modulo $N$. Note that in this equation the time shift in $C_2[n+m]$ is circular because the discrete operation

is periodic (Tsui 2005). The power spectral density $|S(f)|^2$ describes the distribution of power within the signal with respect to the frequency and is based on the Fourier transform $S(f)$ of the autocorrelation function:

$$S(f) = \int_{-\infty}^{\infty} R(\tau) e^{-2\pi i f \tau} d\tau. \tag{2.10}$$

Figure 2.2 shows the ACF and PSD for a simple rectangular function. Note that the smallest unit of a PRN code sequence is called chip.




(a) (b) (c)

Figure 2.2: Rectangular signal (a), autocorrelation function (b), and power spectral density function (c)

Special emphasis in the following sections is laid on the signals of GPS L1 C/A, GPS L2C, and Galileo E1B/E1C open service (OS), using the BPSK, BOC, and multiplexed binary offset carrier (MBOC) code modulation.

### 2.2.2 Binary offset carrier modulation

At the BOC signal modulation the PRN spreading code is modulated using a rectangular subcarrier derived from a sine or cosine phase. The subcarrier-function of code chips is defined as

$$\text{sign}(\cos(2\pi f_s t)), \tag{2.11}$$

with subcarrier frequency $f_s$ and the code chips starting at $t = 0$. Thus, by expanding (2.1) a satellite signal using a BOC modulation can be written as

$$s_{\text{BOC}}(t) = \sqrt{2P} \sum_{m=-\infty}^{\infty} \sum_{k=0}^{N_C-1} D[m]C[k] \, \text{rect}_{T_C}(t - mT_D - kT_C) \, \text{sign}(\cos(2\pi f_s t)) \cos(2\pi f t). \tag{2.12}$$

It is abbreviated using BOC($f_s, f_c$), where $f_s$ is the subcarrier frequency, and $f_c$ is the PRN code chipping rate, both multiples of the carrier frequency. The subcarrier frequency is chosen to have an integer number of half periods, $T_s$ (subchips) within a chip of the spreading sequence. The basic code chipping frequency, which is common to GPS and Galileo, is defined as $f_0 = 1.023$ MHz. Thus, $f_s$ and $f_c$ are defined as:

$$f_s = m f_0 = \frac{1}{2T_s}, \tag{2.13}$$

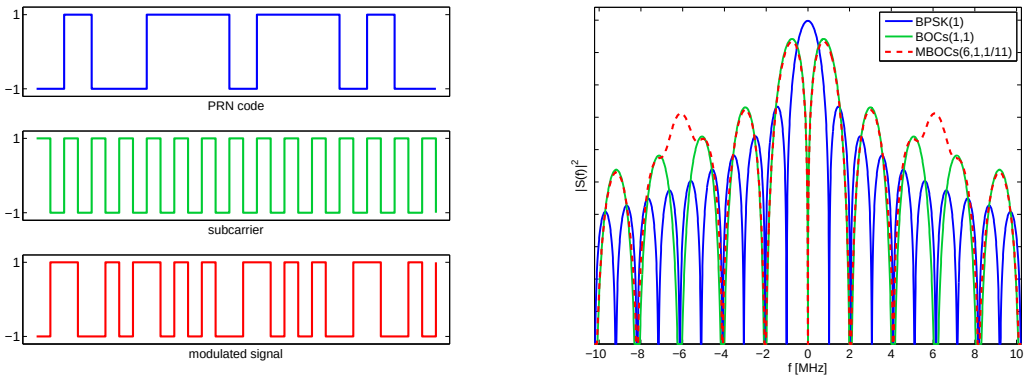
$$f_c = nf_0 = \frac{1}{T_c}. \tag{2.14}$$

The notation BOC$(n, m)$ represents a short version for a BOC modulation generated using a $m \times 1.023$ MHz square wave frequency for the subcarrier and a $n \times 1.023$ MHz PRN code chipping rate. Note that in the literature often the additional subscripts $s$ and $c$ are used to distinguish the sine-phased and the cosine-phased subcarrier. Analogously, the BPSK codes can be written as BPSK$(n)$.

The main purpose of the sub-modulation method is to shape the frequency spectrum in order to assign the signal energy to a dedicated frequency part, i.e, the signal power is shifted away from the carrier frequency in upper and lower side bands, with zero power at the center frequency. The center frequency having zero power is the primary benefit of BOC modulation (Gleason and Gebre-Egziabher 2009). This allows using the same frequency alongside existing phase-shift keying (PSK) signals. An additional advantage lies in the higher resistance to multipath and a small advantage in the code tracking accuracy (Gleason and Gebre-Egziabher 2009). However, signal acquisition and tracking are becoming more difficult in the presence of a subcarrier modulation due to appearance of additional side peaks in the autocorrelation function (Sect. 4.3). The proposed Galileo E1 open service (OS) signal (Sect. 2.4) is composed of two different BOC modulations using a certain ratio of power, called MBOC. In case of Galileo E1 OS, a BOC(1,1) and a BOC(6,1) are combined to a MBOC(6,1,1/11). Thereby, the signal power is $^{10}/_{11}$ for BOC(1,1) and $^{1}/_{11}$ for BOC(6,1). This method adds additional power to the side lobes and thus enhances the receiver's tracking properties. The power spectral density of the MBOC(6,1,1/11) signal is analytically given by:

$$|S(f)|^2 = \frac{10}{11}|S_{\mathrm{BOC}(1,1)}(f)|^2 + \frac{1}{11}|S_{\mathrm{BOC}(6,1)}(f)|^2. \tag{2.15}$$

For details on the different types of implementations please refer to, e.g., Hein et al. (2006). Figure 2.3b shows a PRN sequence modulated with a subcarrier. In the right part of Figure 2.3 the analytical PSDs of a BPSK(1), a BOC(1,1), and a MBOC(6,1,1/11) modulated signals are shown.




(a) (b)

Figure 2.3: Binary offset carrier modulation (a) and comparison of different power spectral densities (b)

## 2.3 GPS signals

The US satellite navigation system consists nominally of 24 operational medium earth orbit (MEO) satellites deployed in 6 evenly spaced planes with an inclination of 55°. The constellation was designed to provide global coverage with four to eight visible satellites above 15° elevation at any time of day. The satellites have an altitude of about 20 200 km above the earth's surface and an orbital period of one half of a sidereal day. The whole timing and signal generation is based on the fundamental frequency $f_0 = 10.23$ MHz. GPS uses three different carrier frequencies for satellite navigation, denoted as L1, L2, and L5 link. The frequencies are situated in the L-band of the electromagnetic spectrum and are multiples of the fundamental frequency. The multipliers as well as the resulting carrier frequencies and wavelengths are listed in Table 2.1.

Table 2.1: GPS frequency bands

| Link | Factor ($*f_0$) | Frequency [MHz] | Wavelength [cm] |
|---|---|---|---|
| L1 | 154 | 1 575.42 | 19.0 |
| L2 | 120 | 1 227.60 | 24.4 |
| L5 | 115 | 1 176.45 | 25.5 |

### 2.3.1 Civil signals

In its current configuration the L1 link consists of two components: the coarse/acquisition (C/A-) code and the precision (P-) code. The C/A-code is a public available PRN sequence for civilian use having 3 dB more power than the P-code and refers to the standard positioning service (SPS). The P-code in contrast refers to the precise positioning service (PPS) for military users. The P-code is encrypted using the W-code to the so-called P(Y)-code. The use of the P(Y)-code is restricted to authorized users only. The C/A-code has a length of 1 023 chips with a code frequency $f_c$ of 1.023 Mega chips per second (Mcps). The modernization of GPS brings a new set of signals for civilian use into the L1 link, denoted as L1C. Following European Commission (2010) it will be composed of a data channel ($L1C_D$) and a pilot channel ($L1C_P$). The L1C signal is not intended to replace the C/A-code but to provide superior performance while maintaining compatibility and interoperability with other signals in the same frequency band. In the past the L2 link was used for transmitting precision and additional military codes only. Up to now civil users utilize codeless or semi-codeless techniques in order to take benefit of the second frequency L2 (Hofmann-Wellenhof et al. 2008). According to Department of Defense (2008) the DoD plans to phase out semi-codeless phase relationships between L1 P(Y) and L2 P(Y) by December 31, 2020. Thus civil users are encouraged to utilize the next-generation civil signals. In the first GPS modernization step a second civil signal was added to the L2 link, referred to as L2C-code. This signal carries a chip-by-chip multiplexed civilian moderate length code (CM) with a code length of 10 230 chips and a longer length code (CL for

civilian long) with 767 250 chips. Muthuraman et al. (2007) state as a unique property of L2CM- and L2CL-codes that the number of ones and zeros are equal. The navigation data are modulated only on the L2CM-code at 50 bps (bits per second) and thus the L2CL-code serves as a pilot channel, which means that no data message is available. Pilot channels are used to increase the sensitivity of a receiver. As of June 1st, 2012, there were in total 9 GPS satellites transmitting the L2C signal. Since the start of the first Block IIF satellite in May 2010 (Langley 2009) GPS is transmitting civil navigation signals, referred to as L5C, on a third frequency (L5 link). It is designed to meet the requirements of safety-of-life applications. The L5C signal can be split into a L5I-code and a L5Q-code being in phase quadrature and using a QPSK(10) modulation. Although the first 2 satellites are already transmitting the new L5C signal, a full constellation will not be available before 2015. According to UNAVCO (*http://facility.unavco.org*) no launches of satellites capable of transmitting L1C are scheduled until at least 2015.
Table 2.2 provides an overview of the main characteristics of the available and planned civil signals transmitted by GPS satellites. Note that the parameters of the modernized GPS signals were taken from Hofmann-Wellenhof et al. (2008) and are subject to change.

Table 2.2: Civil GPS signals

| Link | PRN code | PRN code length [chip] | Code rate [Mcps] | Modulation | Bandwidth [MHz] | Data rate [sps/bps] |
|---|---|---|---|---|---|---|
| L1 | C/A | 1 023 | 1.023 | BPSK(1) | 2.046 | 50/50 |
| | $L1C_D$ | 10 230 | 1.023 | MBOC(6,1,1/11) | 4.092 | 100/50 |
| | $L1C_P$ | $10\,230 \cdot 1\,800$ | 1.023 | MBOC(6,1,1/11) | 4.092 | – |
| L2 | L2CM | 10 230 | 1.023 | BPSK(1) | 2.046 | 50/25 |
| | L2CL | 767 250 | 1.023 | BPSK(1) | 2.046 | – |
| L5 | L5I | $10\,230 \cdot 10$ | 10.23 | BPSK(10) | 20.46 | 100/50 |
| | L5Q | $10\,230 \cdot 20$ | 10.23 | BPSK(10) | 20.46 | – |

To generate the mentioned PRN codes, linear feedback shift registers (LFSRs) are used. More information on LFSR and on PRN code generation is provided in Chap. 4.

### 2.3.2 Navigation data structure

The modernized GPS uses two different structures for broadcasting the civil navigation data. The data bit rate, in general, is synchronous but slow compared to the PRN code epochs. The low data rate guarantees a low bit error ratio (BER) under weak signal conditions (Dierendonck 1996). The original data message (NAV) is modulo-2 added with 50 bps to the C/A- and P(Y)-codes on L1 frequency, and optional on the L2 P(Y)-code. The structure of the NAV message is organized in frames, divided into five subframes. Each subframe has a length of 300 bits which corresponds to a duration of six seconds. Each subframe starts with the telemetry word (TLM) containing an 8-bit synchronization pattern,

followed by the hand-over word (HOW). Beside others the HOW contains the subframe identification number and the time of week (TOW), sometimes denoted as Z-count. Each subframe ends with six parity bits for error detection. The parity algorithm (CRC) as well as a detailed description of the format and content can be found in Department of Defense (2011). The navigation data format of the new GPS signals, denoted as civil navigation message (C/NAV), is both more compact and more flexible than the NAV message. According to Department of Defense (2011), the C/NAV basic format is composed of 300-bit long messages. Thereby, each message contains a CRC parity block consisting of 24 bits covering the entire message. Nominally, according to Department of Defense (2011), the C/NAV message contains more accurate data compared to the NAV message. The whole bit stream is forward error correction encoded by a rate 1/2 convolutional code resulting in a 50 sps (symbols per second) rate. For the decoding a Viterbi decoding algorithm is used (Chap. 6). The benefit of this encoding is the more reliable recovery of the data bit stream. Parkinson and Spilker (1996) state that it is possible to decode the data stream, using a soft decision Viterbi decoding algorithm, with the same BER even at a 5 dB less signal-to-noise ratio (S/N).

## 2.4 Galileo signals

The European satellite navigation system Galileo foresees a 27 + 3 satellite constellation in three planes, having an inclination of 56° with respect to the equatorial plane (Falcone et al. 2006). Two experimental satellites have been launched in 2005 and 2007, named as GIOVE-A and GIOVE-B. The acronym GIOVE stands for Galileo in-orbit validation element. These satellites were the first to emit Galileo-like signals. On October 21$^{st}$, 2011 the first two Galileo in-orbit validation (IOV) satellites have been successfully launched. The third and fourth Galileo IOV satellite were launched on October 12, 2012. This marked the start of space segment construction for the European satellite navigation system (European Space Agency 2011). The first Galileo user position computation, using the four IOV satellites, is expected in spring 2013. The full operational capability (FOC) is scheduled for 2018. For compatibility reasons with other GNSS, the Galileo signals will be situated in the L-band and will use CDMA techniques. In total five carrier frequencies are foreseen: E1, E6, E5, E5a and E5b. The carrier frequencies, obtained from the fundamental frequency $f_0 = 10.23$MHz, are summarized in Table 2.3. Note that E1 corresponds to L1 in the GPS nomenclature and that E5a is equivalent to the GPS L5 band. The E5 band is typically denoted as E5a + E5b band, as it can be seen as the union of those two bands.

### 2.4.1 Open service signals

Galileo uses a service-oriented approach. It will provide ten different signal types making it possible to offer five different services: open service (OS), commercial service (CS), public regulated service (PRS), safety of life (SoL) and search and rescue (SAR). Note that according to European Commission (2011) the SoL service will not be available during the initial operational capability (IOC) phase and studies to redefine this service are currently conducted. Galileo, like GPS, also utilizes spread-spectrum signals to multiplex different

Table 2.3: Galileo frequency bands

| Link | Factor ($*f_0$) | Frequency [MHz] | Wavelength [cm] | Bandwidth [MHz] |
|---|---|---|---|---|
| E1 | 154 | 1 575.420 | 19.0 | 32.0 |
| E6 | 125 | 1 278.750 | 23.4 | 40.9 |
| E5 | 116.5 | 1 191.795 | 25.2 | 51.2 |
| E5a | 115 | 1 176.450 | 25.5 | 24.0 |
| E5b | 118 | 1 207.140 | 24.8 | 24.0 |

signals. The different signals can be split into two groups: signals containing navigation data (referred to as data channels), and signals without navigation data (called pilot channels). Data and pilot channels are situated in pairs in each band, except for E1A and E6A. Both, data and pilot signals, are transmitted at equal power. The purpose of the pilot channel is to improve the tracking performance under weak signal conditions by using long coherent integration time (Chap. 5). In order to provide better crosscorrelation properties, Galileo will use ranging codes of longer length compared to GPS. One method of generating longer ranging codes is to combine a faster medium length code, called primary code, with a slower smaller length secondary code. Galileo uses secondary codes of different length combined with the primary codes to form the final spreading codes.
Three data and three pilot channels, out of the ten signals are accessible to all Galileo users. The E1 link includes three navigation signals, denoted as E1A, E1B, and E1C. While E1A is used for the PRS and thus classified, E1B and E1C are intended for OS. The PRN codes of the data channel E1B and the pilot channel E1C are modulated by MBOC(6,1,1/11), as described in Sect. 2.2.2. This reduces the inter-system interference but maintains compatibility and interoperability. Both components have a primary code length of 4 096 chips. The E1B channel transports the navigation data, integrity information, and encrypted commercial data and has no secondary code. The pilot channel E1C in contrast is modulated by a secondary code of length 25 but carries no data. The primary PRN code sequences of E1B and E1C are designed to be memory codes. This means that they are not generated in the satellite or receiver by LFSR but are stored in memory. The codes are published in European Commission (2010). The E5 link uses an alternative binary offset carrier (AltBOC) in the specific case an AltBOC(15,10), modulation scheme. The AltBOC allows a constant envelop modulated signal and is derived from the BOC modulation. The AltBOC features a split spectrum with two main lobes. Thus, the E5 signal can be seen as two separate sidebands. They are denoted as E5a and E5b and are transmitting four OS signal components: two pairs of data and pilot channels. The data channels on E5a and E5b are commonly denoted as E5a-I and E5b-I, whereas the corresponding pilot channels are denoted as E5a-Q and E5b-Q. Following Canalda Pedrós (2009) the AltBOC modulated signal can be represented by two separate BPSK(10) modulations around the E5a and E5b center frequency. The E5a and the E5b channel emit unencrypted ranging codes and in case of the data channels, also navigation data. Thereby, E5a transmits the basic navigation data at a low data rate (50 sps) in

order to enable a more robust data demodulation. The E5b data stream uses a data rate of 250 sps and contains beside the basic navigation data, also integrity information and encrypted commercial data. E5b features apart from the OS also the CS and the potential SoL service. The E6 link is reserved for the commercial and the public regulated service. Table 2.4 summarizes the properties of the available OS signals.

Table 2.4: Galileo open service signals

| Link | PRN code | Channel | PRN code length [chip] | | Code rate [Mcps] | Modulation |
|---|---|---|---|---|---|---|
| | | | primary | secondary | | |
| E1 | E1B | data | 4 092 | 1 | 1.023 | MBOC(6,1,1/11) |
| | E1C | pilot | 4 092 | 25 | 1.023 | MBOC(6,1,1/11) |
| E5 | E5a-I | data | 10 230 | 20 | 10.23 | BPSK(10) |
| | E5a-Q | pilot | 10 230 | 100 | 10.23 | BPSK(10) |
| | E5b-I | data | 10 230 | 4 | 10.23 | BPSK(10) |
| | E5b-Q | pilot | 10 230 | 100 | 10.23 | BPSK(10) |

### 2.4.2 Navigation data structure

Galileo transmits four message types according to the European Commission (2010). The two message types accessible for civil users are the freely accessible navigation message (F/NAV) and the integrity navigation message (I/NAV), while the other two, which are encrypted, contain data for CS and PRS. As in the case of GPS, both civil message types contain all necessary navigation data. The I/NAV message provides integrity information in addition. Both messages are transmitted as a sequence of frames, composed of several subframes. The subframes are further divided into pages. The page is the basic format and can be transmitted in several time intervals, allowing a fast repetition of urgent data (e.g., integrity information) and medium and slow repetition rates for less important data. Each page consists of a synchronization pattern, data fields containing the data, and tail bits. The allocation of the open-access navigation messages, as well as their data rates are shown in Table 2.5.

Table 2.5: Galileo navigation message type allocation

| Message type | Services | Channel(s) | Data rate [sps/bps] |
|---|---|---|---|
| F/NAV | OS | E5a-I | 50/25 |
| I/NAV | OS/CS/SOL | E1B/E5b-I | 250/125 |

Since both E1B and E5b-I transmit the I/NAV message, the page ordering between them is changed to enable a faster data reception when using dual-frequency receivers. Both navigation message types use a three-step error correction encoding to reduce the bit error rate which occurs in general at higher bit rates. The data fields of each page are protected

by a 24 bit cyclic redundancy check parity block to detect the reception of corrupted data bits. The data fields, the CRC parity bits, and the tail bits are 1/2 convolutional encoded with an FEC code. The FEC method is used in order to have a better BER. Using half rate convolutional encoder results in a symbol rate which is twice the data rate. Finally, the $n \times k$ block interleaver takes $n \times k$ symbols and fills a matrix column by column having $k$ rows and $n$ columns. The symbols are then transmitted row by row. The F/NAV message uses a block interleaver of size $61 \times 8$, while the I/NAV message uses $30 \times 8$. The detailed message structure specifications of F/NAV and I/NAV, as well as the CRC, FEC and interleaving parameters are available in European Commission (2010).

# Chapter 3

# Software-based receiver architecture

## 3.1 Introduction

This chapter provides an overview of the software-defined radio concept. Note that in communications the term software-defined radio is commonly used, while in geodesy and navigation the terminology software-based receiver is used. Since the background of the author of this thesis is geodesy the term software-based receiver will be used in this thesis. This chapter shows an approach which is common to nearly all software receivers: placing the analog to digital conversion as close as possible to the antenna and processing as much as possible using software techniques in order to reduce the hardware elements to a minimum. Thus, the ideal case would be to digitize the GNSS signal right at the antenna. But as the Nyquist (Shannon) theorem must be fulfilled, this results in a sampling frequency of at least twice the analog bandwidth. Considering the GPS L1 frequency and assuming a 2 bit quantization per sample the following rough estimation can be done:

$$\begin{aligned} f_{\mathrm{GPS_{L1}}} &= 1575.42 \text{ MHz}, \\ f_{\mathrm{sampling}} &\geq 2 \cdot f_{\mathrm{GPS_{L1}}} \approx 3150 \text{ MHz}, \\ \text{Data rate} &\geq 2 \cdot f_{\mathrm{sampling}} = 6.3 \text{ GBit/s} \approx 800 \text{ MByte/s}. \end{aligned} \tag{3.1}$$

This leads to a data rate of about 800 MByte per second which is, currently, too high for energy efficient analog-to-digital converters (ADCs) and digital circuits. Thus, an intermediate frequency (IF) analog front-end is used to downconvert the incoming signal to an intermediate frequency in the MHz-range where it is converted into a digitized signal (binary bit stream). As shown in Figure 3.1 the high-level receiver architecture is split into a hardware and a software part.

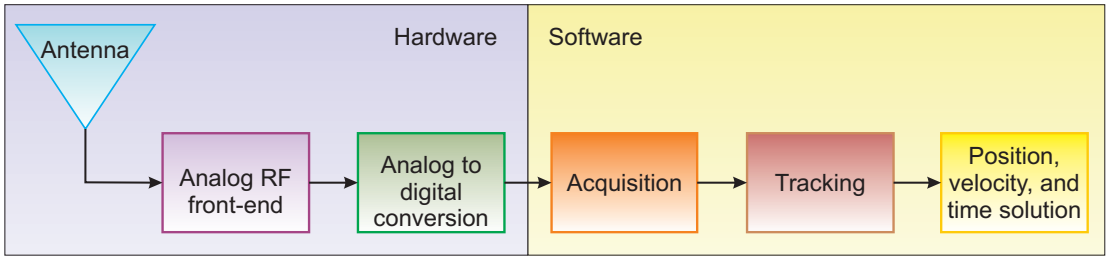



Figure 3.1: Generic software-based GNSS receiver architecture

The developed software-based receiver can be split into the DSP part and the position,

velocity, and time (PVT) part. The digital signal processing (DSP) covers the signal acquisition, signal tracking, and raw data generation, and is implemented in software. The computed raw data are then used within the position, velocity, and time (PVT) module to obtain a user position. The flexibility of this software-based implementation allows modifications and adaptations to new challenges by updating the software instead of replacing hardware parts. This saves hardware development costs. Currently, the most common receiver architectures are based on special hardware chips with the drawback that an upgrade to new specifications is difficult and results in a redesign of the hardware. Following Gleason and Gebre-Egziabher (2009) the primary innovation of a software receiver is the elimination of digital signal processing hardware, resulting in great simplification of the receiver design, including size, power consumption, cost, and flexibility. The disadvantage is that the digital signal processing parts, especially the correlators, are still needed. The correlation computation is, thus, moved to the central processing unit (CPU). This increases the processing load of the software. On the other hand, the software trend caused the engineers to have a closer look into signal processing and provides an optimum platform for developing and testing new algorithms.

## 3.2 GNSS front-end

The process of data acquisition, how the digital signal is gained, and what constraints arise thereby are important factors influencing the positioning performance. In general the whole process starts with the GNSS signal, propagating through space, which is then received by an antenna. The signal induces a voltage within the antenna which is extremely weak. Department of Defense (2011) states a guaranteed signal power $P$ of $-158.5$ dBW in case of GPS L1 C/A at the earth's surface. The thermal noise, following Hofmann-Wellenhof et al. (2008), is assumed to be white and Gaussian distributed. The expected power $N$ of the filtered noise can be computed as product of the Boltzmann constant $k = -228.6$ dBWK$^{-1}$Hz$^{-1}$, the absolute temperature $T$ in Kelvin, and the bandwidth $B_r$ processed by the receiver (Akos et al. 2000) and reads

$$N = kTB_r \text{ [W]}. \tag{3.2}$$

The signal-to-noise ratio (S/N) describes the performance by relating the average signal power $P$ to the noise power $N$. Taking the above mentioned signal power $P$ and the noise power $N$ for a typical signal bandwidth of 2 MHz, the S/N can be computed as follows

$$\text{S/N} = 10 \log_{10} \frac{P}{N} \approx -18 \text{ [dB]}. \tag{3.3}$$

In this case the S/N becomes negative. Thus, the signal is below the noise level and not detectable by a spectrum analyzer. This feature shows the need for appropriate signal processing to acquire and process the satellite signals. The induced voltage within the antenna is forwarded to the radio frequency (RF) front-end. A typical RF front-end contains a low-noise amplifier (LNA), filters to reject interfering signals, a mixer, and a local oscillator (LO) for the downconversion. Figure 3.2 provides a generic overview of the front-end operations.

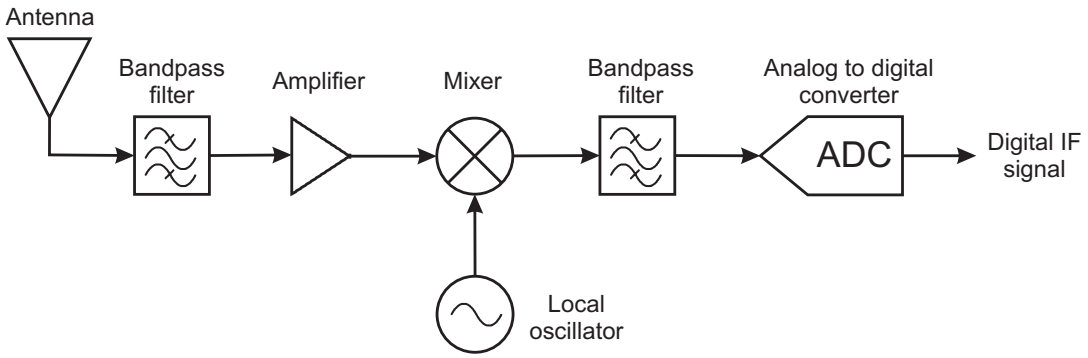



Figure 3.2: Schematic representation of a generic RF front-end

The front-end elements define the noise figure of the receiver and the observed signal power $P_r$ after the RF front-end. The relation between the received signal power to the noise per 1 Hz, denoted as carrier-to-noise-power-density ratio ($C/N_0$), reads as

$$C/N_0 = 10 \log_{10} \frac{P_r}{N_0} = 10 \log_{10} \frac{P_r}{kT}. \tag{3.4}$$

In case of using the same values as in the example above the $C/N_0$ is at about 45 dBHz. According to Hofmann-Wellenhof et al. (2008) a $C/N_0$ below 34 dBHz characterizes weak signals. The LNA amplifies the received signal and rejects out-of-band interference by using a bandpass filter. Afterwards a downconversion to IF is performed by a mixer driven by an LO. Typically LOs are based on quartz crystal oscillators. Due to their instability compared to atomic clocks a time drift is introduced. This drift in combination with the offset between the satellite clock and receiver clock has to be considered as an unknown parameter within the position estimation (Chap. 7). Starting from (2.3) the GPS L1 C/A signal for one satellite $s_{\mathrm{RF}}$ can be written as

$$s_{\mathrm{RF}}(t) = \sqrt{2P_{\mathrm{C/A}}}\, C_{\mathrm{C/A}}(t)\; D(t)\; \cos(\phi_0 + 2\pi(f_{\mathrm{L1}} - f_d)\; t) + e_{\mathrm{RF}}(t), \tag{3.5}$$

where $\phi_0$ denotes the initial received phase, $f_{\mathrm{L1}}$ the L1 carrier frequency, and $f_d$ the Doppler shift. The last term, $e_{\mathrm{RF}}$ represents the white Gaussian noise. The Doppler shift $f_d$ is a frequency shift compared to the nominal frequency, caused by the relative motion between transmitter and receiver (Chap. 4). Downconversion is a simple shift of the frequencies within the frequency spectrum. This can be done by multiplying the incoming signal centered on the frequency $f_{\mathrm{L1}}$ with a local generated pure harmonic signal $f_{\mathrm{LO}}$ (Hofmann-Wellenhof et al. 2008). The local oscillator is defined by the frequency $f_{\mathrm{LO}}$ as

$$\mathrm{LO}(t) = 2\cos(2\pi f_{\mathrm{LO}} t). \tag{3.6}$$

Following Ziedan (2006), the mixing results in a signal with some harmonics plus upper and lower sidebands, i.e., sidebands with center frequencies of $(f_{\mathrm{L1}} + f_{\mathrm{LO}})$ and $(f_{\mathrm{L1}} - f_{\mathrm{LO}})$. To remove the harmonics and the upper sideband a bandpass is used. The center frequency of the lower sideband

$$f_{\mathrm{IF}} = f_{\mathrm{L1}} - f_{\mathrm{LO}} \tag{3.7}$$

is called intermediate frequency (IF). The process of downconversion is shown in Figure 3.3.

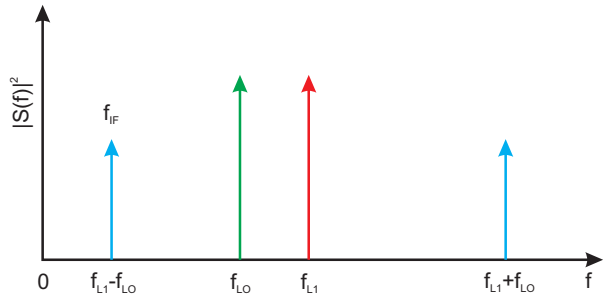



Figure 3.3: Frequency spectrum of the downconversion process

The resulting signal corresponds to

$$s_{IF}(t) = \sqrt{2P_{C/A}}\, C_{C/A}(t)\; D(t)\;\; \cos(\phi_0 + 2\pi(f_{IF} - f_d)\; t) + e_{IF}(t). \tag{3.8}$$

Note that not all receivers use the IF concept, but some manage the downconversion directly to zero or near zero frequency, i.e., down to baseband $f_{\mathrm{IF}} = 0$. After the downconversion the analog to digital (A/D) converter discretizes the incoming signal into a sequence of discrete samples and quantizes them by a predefined number of bit levels. An automatic gain control (AGC) is used to keep the signal level within an acceptable range. Thereby, the sampling frequency $f_s$ has to satisfy the Nyquist (Shannon) theorem in order to reconstruct the signal from discrete samples and to prevent aliasing effects. Thus, $f_s$ has to be greater than twice the maximum frequency of the IF signal. The quantization transforms the signal levels to the nearest quantization threshold values. Hofmann-Wellenhof et al. (2008) state that a one-bit quantization would be sufficient for signal processing, but higher quantization levels (e.g., 2-bit or 4-bit) show a better signal-to-noise ratio. The satellite signals after A/D conversion are discrete $s[k]$, thus the sampled received signal is expressed following Ziedan (2006) as

$$s_{\mathrm{IF}}[k] = A\; C_{C/A}[k]\; D[k]\;\; \cos(\phi_0 + 2\pi(f_{\mathrm{IF}} - f_d)\; kT) + e[k]. \tag{3.9}$$

Following Mongrédien et al. (2011) most front-ends feature a low-IF architecture with RF-bandwidth of approximately 2 to 4 MHz and a low resolution of the analog to digital converter of 1 to 3 bit. For narrow-band signals, like GPS L1 C/A and Galileo E1 BOC(1,1), this is sufficient but not for the new Galileo MBOC(6,1,1/11) or BPSK(10) signals. More information on RF front-end operations is provided in Samper et al. (2008) and Pany (2010).

## 3.3 Structure of the input data

Two different types of IF data sets have been used to verify the implemented algorithms and analyze the performance of the developed overall system. Since not all signals or systems, under investigation in this thesis, are currently fully available, a mixture of simulated and real-world data has been used. For verification purposes digital IF data sets generated by a

signal generator have been used. Real-world data sets have been recorded to demonstrate the performance and behavior under real conditions. The following data sources have been used:

- Signal generator
  - GIPSIE®
- RF front-end
  - SiGe Semiconductor GN3S v2 front-end
  - Fraunhofer IIS multiband front-end

In case of simulated data the GIPSIE® – intermediate frequency simulator (GIPSIE®–IFS), developed by TeleConsult Austria GmbH, was used to simulate GPS L1 C/A as well as Galileo E1B signals. The SiGe GN3S Sampler v2, manufactured by SiGe Semiconductor, has the capability to record L1/E1 signals, whereas the Fraunhofer Gesellschaft – Institute of Integrated Circuits (FhG-IIS) multiband front-end features L1/E1, L2, and L5/E5a signals. Both simulator and front-ends provide binary data files containing a digital signal. As output of the front-end either a digital signal stream for real-time processing or a digital signal file for post-processing is available. The software-based receiver discussed in this thesis uses binary data files containing digital IF signals in post-processing mode. Table 3.1 shows an overview of the frequency capabilities of the different data sources.

Table 3.1: Frequency band capabilities of the different RF front-ends

| | Frequency bands | |
|---|---|---|
| | GPS | Galileo |
| GIPSIE®–IFS | L1 | E1 |
| SiGe GN3S Sampler v2 | L1 | E1 |
| FhG-IIS front-end | L1, L2, L5 | E1, E5 |

Thereby, the amount of data is directly correlated with the sampling frequency and has a direct impact on the processing time. The topics data amount, processing and execution time is covered in Chap. 8. The important parameters for the signal processing are:

- Sampling frequency;
- Intermediate frequency;
- Quantization.

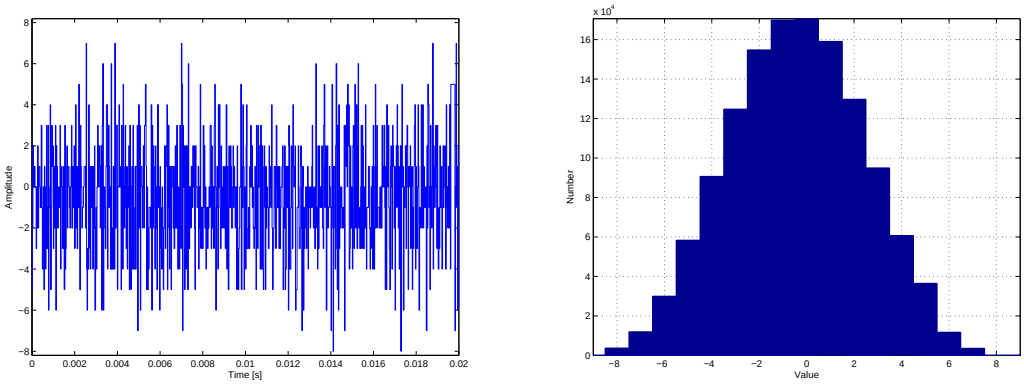
The specific parameters for the different data sources are given in Table 3.2.

Table 3.2: Front-end specifications (TeleConsult Austria GmbH 2010, SiGe Semiconductor 2011, Fraunhofer 2010)

| | GIPSIE®–IFS [1] | SiGe GN3S Sampler v2 | FhG-IIS front-end |
|---|---|---|---|
| Manufacturer | TCA | SiGe Semiconductor | FhG-IIS |
| Center frequency | 1 575.42 MHz | 1 575.42 MHz | 1 575.42 MHz<br>1 227.60 MHz<br>1 176.45 MHz |
| Sampling rate | 38.192 MHz | 8.1838 MHz | 40.96 MHz |
| IF bandwidth | 2.046 MHz | 4.4 MHz | 18 MHz |
| Quantization | 4 bit | 2 bit | 2 or 4 bit |
| IF frequency | 9.548 MHz | 38.400 kHz | 13 MHz |
| Connector | File | USB | 2 x USB |

[1] Sampling rate, IF frequency, IF bandwidth, and quantization are user adjustable; the given values are used in this thesis

Following Borre et al. (2007), the above mentioned parameters provide all the necessary information for the signal processing algorithms. Some other information, such as time, date, and approximate location of the data collection can speed up the acquisition, but are not required. In case of the SiGe GN3S Sampler v2 and the FhG-IIS front-end the digital signals were received at the USB port of the computer and then recorded into a file. As an example Figure 3.4 shows a time domain representation, as well as a histogram, of 0.02 seconds digital signal recorded by the FhG-IIS multiband front-end. While in the time

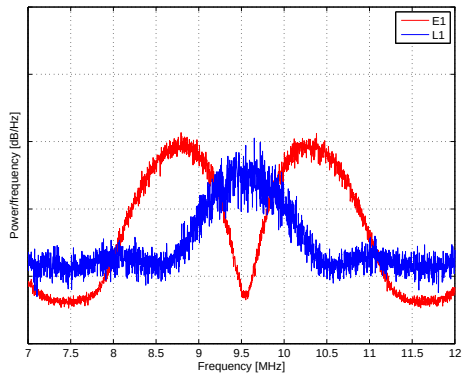



(a) (b)

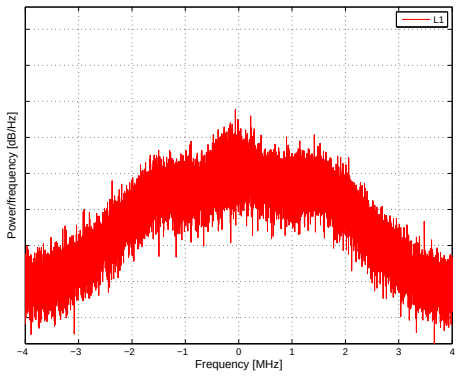
Figure 3.4: Time domain (a) and histogram (b) of 0.02 sec IF file recorded with the FhG-IIS front-end

domain no apparent structure, apart from the sampling frequency, is visible, the histogram shows that all the 16 levels of the 4-bit quantization are present. Figure 3.5 presents the power spectral densities of the selected digital IF input data. The sharp spectral lines in

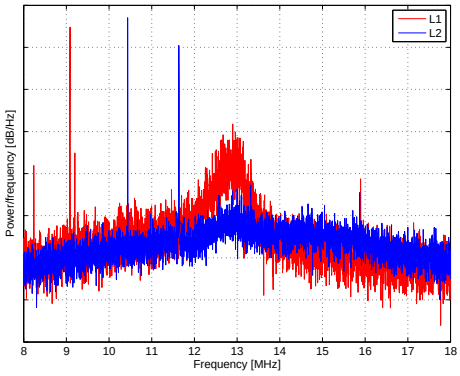
Figure 3.5c are harmonics of the internal local oscillator.




(a) GIPSIE®–IFS




(b) SiGe GN3S Sampler v2 front-end




(c) FhG-IIS front-end

Figure 3.5: Power spectral density of the different input signals

## 3.4 Receiver architecture

After the signal conditioning (downconversion and sampling) the software-based receiver executes two sequential signal processing operations: acquisition and tracking. Initially, the receiver has to search for visible satellites and to obtain rough estimates of the Doppler shift and the code phase of those satellites. This step is done by correlating the received signal with a locally generated PRN code and is referred to as acquisition (Chap. 4). After a successful acquisition the tracking module takes over. The basics of following the acquired satellite signal, denoted as tracking, are described in detail in Chap. 5.
One output of the tracking module delivers the navigation symbols, which are preprocessed and decoded to obtain the navigation data. The actual measurements are obtained in regular time intervals (Chap. 6). Out of the navigation data bits, the ephemerides data for the computation of the satellite positions, the ionospheric correction data and the system time information are extracted besides other parameters. The pseudorange computations are based on the time of transmission from the satellite and the time of arrival at the RF

front-end. The PVT module computes the receiver's position out of the raw observables and the decoded navigation data in several steps (Chap. 7). The digital signal processing tasks described above have to be executed for different satellites separately. Thus, a channelized structure is used for processing (Figure 3.6).

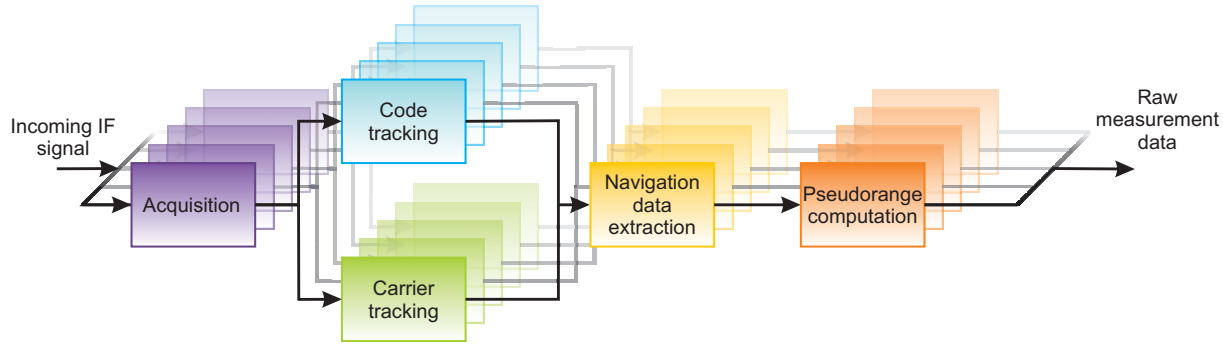



Figure 3.6: Receiver channel overview

An increased number of channels allows processing of more satellite signals. On the other hand, more channels require more processing capabilities. This approach is ideal for being implemented in software, since parallelization methods can be applied. The developed software-based receiver described in this thesis is based on this channelized structure. Currently only one channel is processed at a time and, thus, the developed receiver is only capable of operating in post-processing mode. Nevertheless, the development receiver comprises all the necessary functions and features in order to obtain a user position. The overall architecture can be split into two main parts, i.e., digital signal processing (DSP), position, velocity, and time (PVT) computation, as shown in Figure 3.7.

The first part contains all the necessary digital signal processing modules for acquiring, tracking, and demodulating GNSS satellite signals. In a first step several milliseconds of a digital signal which are stored in binary data files, either coming from the front-ends or the signal generator, are read into memory and forwarded to the acquisition module. As stated above, the main purpose of this module is to find visible satellite signals. Once a visible satellite is found, the acquisition parameters (Doppler frequency and initial code-phase) are forwarded to the tracking module. Based on the acquisition parameters the tracking module tries to keep track of the signal using code- and carrier-tracking loops. Hence, the loops read the necessary amount of IF signal data from the file. Then the output, namely the correlation results, are forwarded to the navigation data extraction module, where the starting points of the preambles or the synchronization bits are determined and consequently the navigation message is decoded and verified. The next step is to compute the relative ranges between the satellites and the receiver, denoted as pseudoranges, based on the tracking results.

Now the second part, the position, velocity, and time module takes over. Based on the raw observables and the navigation data, the raw observables are corrected using the broadcasted model parameters. In case of dual-frequency measurements the ionospheric error is reduced by forming ionosphere-free linear combinations (LCs) (Chap. 7). The satellite positions are computed out of the decoded broadcast ephemerides and the measurement time and together with the corrected observations the receiver position is determined using

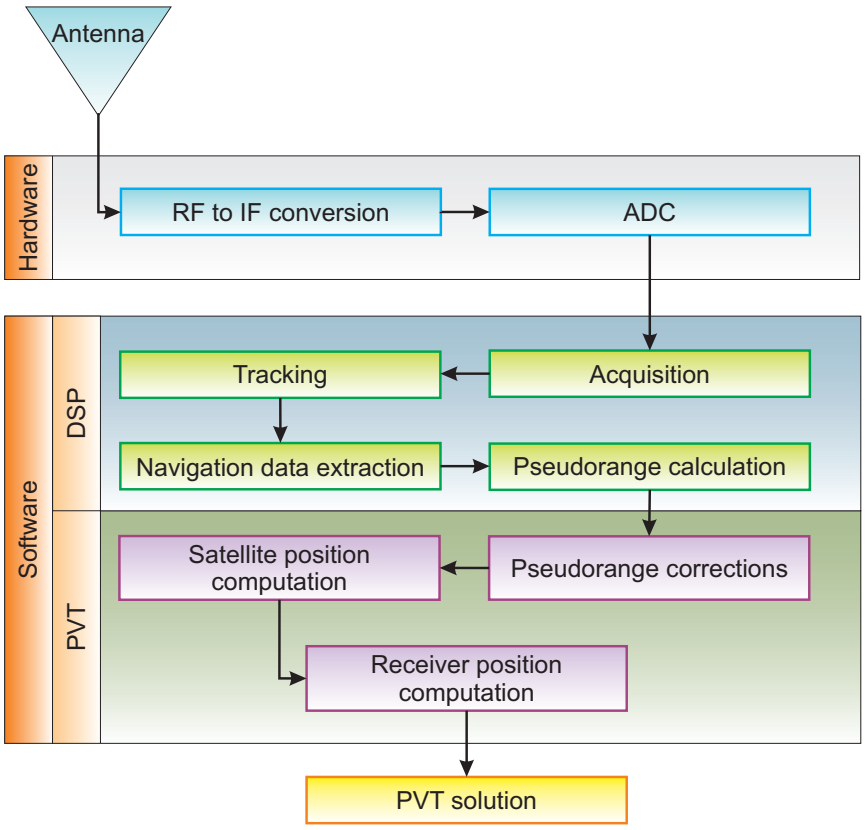



Figure 3.7: Software-based receiver high-level architecture

a least squares adjustment algorithm. Beside position, velocity, and time information the receiver provides accuracy and integrity measures to the user.
Both developed modules – the DSP module and the PVT module – represent a fully functional receiver, capable of processing GPS L1 C/A, GPS L2C, and Galileo E1B signals. Parts of the PVT module have been developed and extended during the author's employment at TeleConsult Austria GmbH within several research projects.
The initial PVT software version was used within the project GAMMA (Assisted Galileo/GPS/EGNOS mass market receiver) which was managed by the European GNSS Agency (GSA) as a project of the 2nd call, area 1B "User Segment – Technological Development" through funds of the EU's 6th Framework Programme (FP). The software was designed to run on an FPGA receiver developed by the Fraunhofer Gesellschaft – Institute of Integrated Circuits. This first version was capable of processing Galileo E1B, GPS L1 C/A, and SBAS signals. Additional features like acquisition aiding and assisted GNSS (AGNSS) have been implemented, too. The software as well as the whole receiver were tested using the GAMMA signal generator (Berglez et al. 2009). Within the follow-up project GAMMA-A (Galileo receiver for mass market applications in the automotive area), funded within the 7th Framework Programme of the EU, the module was further extended to process GPS L5 and Galileo E5 signals (Overbeck et al. 2010). The PVT software was further enhanced and used by TeleConsult Austria GmbH within two other European projects, called ASPHALT and GENEVA. The Galileo/EGNOS Enhanced Driver Assistance (GENEVA) project as well as the Advance Galileo Navigation System for Asphalt's Fleet Machines

(ASPHALT) project received fundings from the European Union Seventh Framework Programme and were managed by the European GNSS Agency (GSA). Within the SoftGNSS–2 project a software-based dual-frequency receiver was developed. SoftGNSS–2 was carried out under the responsibility of the Austrian Research Promotion Agency (FFG) and was funded through the Austrian Federal Ministry for Transport, Innovation, and Technology (BMVIT) (Berglez et al. 2010). While the DSP part in the SoftGNSS–2 project was accomplished by the project partner Signal Processing and Speech Communication Laboratory at Graz University of Technology, the PVT module was extended to GPS L2C. The L2C signal processing algorithms within this thesis have been developed independently from the SoftGNSS–2 project by the author of this thesis. Beside these projects the developed receiver is currently used at TeleConsult Austria GmbH within several other research and development projects.

In the following chapters a detailed description of the implemented algorithms for acquisition, tracking, and positioning is given. Thereby, special emphasis is put on the different signals. The implemented modules have been verified using simulated data using closed loop simulations. Closed loop simulations provide the ability of comparing the computed quantities with the simulated ones. These simulations have been accomplished using the GIPSIE® – satellite constellation simulator (GIPSIE®–SCS) as well as the GIPSIE®–IFS. The performance evaluation has been done using real-world data recorded by the RF front-ends.

# Chapter 4

# Signal acquisition

## 4.1 Introduction

The acquisition can be considered as a search process and it is the first stage within the DSP. The purpose of the acquisition is twofold: first of all it is used to decide whether a specific satellite signal is present or not; and secondly, it is used to determine rough estimates of the actual frequency (Doppler frequency shift) and the code phase of the incoming signal from a specific satellite signal. The relative motion between the satellite and the receiver causes a frequency shift, known as Doppler shift, in the received signal compared to the nominal frequency. The Doppler frequency shift is proportional to the relative radial velocity. In case of GNSS the maximum radial velocity is approximately 0.9 $^{km}/_{s}$, the resulting Doppler shift corresponds – assuming a carrier frequency of 1.5 GHz – to 4.7 kHz (Hofmann-Wellenhof et al. 2001). Depending on the receiver velocity the Doppler shift varies between ±5 kHz for a static receiver and ±10 kHz assuming very high receiver dynamics. The code phase denotes the code offset between the received signal and locally generated replica code within one code period. The knowledge of the code phase is essential because a perfectly aligned local replica of the PRN code is needed within the tracking module to de-spread the signal (remove the incoming code) and obtain the navigation data. Thus the acquisition task can be seen as a two dimensional search problem, finding the actual Doppler shift and the actual code phase. The Doppler dimension is associated with the replica carrier wave, while the code phase (or range) dimension corresponds to the replica code. After a brief introduction into PRN code replica generation and carrier replica generation at the beginning of this chapter, the implemented acquisition algorithm is described. Special focus is thereby laid on the differences within the acquisition process for different GNSS signals. The problem of the occurrence of destructive correlation when processing GNSS signals is addressed as well. At the end of this chapter selected acquisition results are presented and discussed.

### 4.1.1 PRN code generation

For acquisition, like for tracking, replicas of both the PRN code and the carrier of the desired satellite signal are required. In order to save processing time all possible PRN codes for the used signals and systems (GPS C/A, GPS L2C, and Galileo E1B) are generated in advance.

The pseudorandom noise codes are commonly generated using linear feedback shift register (LFSR). Note that, beside the PRN codes generated by LFSRs, also memory codes, which are stored in memory, exist. A LFSR consists of $n$ register cells, each representing one

chip. At each step the content of each cell is moved by one cell from left to right, the rightmost bit pops out of the register and represents the current code chip. The input to the leftmost cell is determined by the state of the previous register cells. The feedback chip is obtained by a characteristic polynomial, combining certain register cells, according to a defined linear generator polynomial using exclusive-or (XOR) operations. The chip sequence obtained by a LFSR is repeated after a maximum code length depending on the length of the register. Assuming a register having $n = 10$ cells will produce a maximum code length of $2^n - 1 = 1\,023$. Depending on the characteristic polynomials and the initial state of the register different sequences can be obtained. In case of GPS L1 C/A-code, two generator polynomials are combined to compute the PRN code sequence for one satellite (Department of Defense 2011).
As mentioned in Sect. 2.3.1 the L2C PRN sequence consists of two codes (L2CM and L2CL) which are multiplexed chip-by-chip. In case of chip-by-chip multiplexing one chip of the L2CM-code is transmitted followed by one chip of the L2CL-code. Both L2CM- and L2CL-codes are generated using a 27 cell long LFSR. This results in a code length of $2^{27} - 1 = 134\,217\,727$ chips. However the LFSR is short cycled to 10 230 and 762 750 chips in case of L2CM and L2CL, respectively. The distinct initial and final states of the LFSR for each space vehicle (SV) are given in Department of Defense (2011). Due to the multiplexing there are exactly 75 periods of L2CM-code within each L2CL-code period. According to Tran and Hegarty (2002) it is not practical to acquire the signal using an exact replica of the L2C-code, due to the data on L2CM-code and the lack of data on the L2CL-code. The L2CL-code can thus be used as a pilot channel enabling signal tracking under very weak signal conditions. Due to the chip-by-chip time-multiplexing the acquisition and thus the local code generation must be modified. Three different methods of L2C replica code generation have been suggested in Tran and Hegarty (2003):

1. Using a L2CM sequence, sampled at 511.5 kHz. Assuming a code search step of ½ chip leads to 20 460 search steps for the entire L2CM-code range. Dempster (2006) states two flaws of this method. First it flattens the correlation shape and thus the multipath resistance behavior is reduced. The second one is that additional L2CM-L2CL crosscorrelation noise is added by integrating during the non-useful L2CL chips and thus the signal-to-noise ratio is degraded.

2. Using a time-multiplexed sequence of L2CM and zeros, sampled at 1.023 MHz. Thereby the L2CL chips are replaced with 0s. This requires the local code to have a return-to-zero form (three states: +1, -1, and 0) (Muthuraman 2010). This doubles the number of code search steps, compared to the previous method, on the one hand, but results in a sharp correlation peak and reduces the crosscorrelation noise (Dempster 2006). This method can also be applied to the L2CL-code.

3. Using a replica L2CM sequence with a duration of 1.5 times the chip length. Like in the method above there are 40 920 search steps (assuming a step size of ½ chip) necessary. The correlation loss is smaller compared to the previous methods due to the 1.5 times longer duration of integration of this chip filter (Tran and Hegarty 2003).

The three different L2CM replica sequences which can be used to acquire the signal are shown in Figure 4.1.

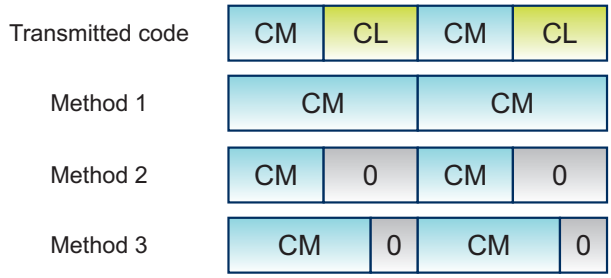
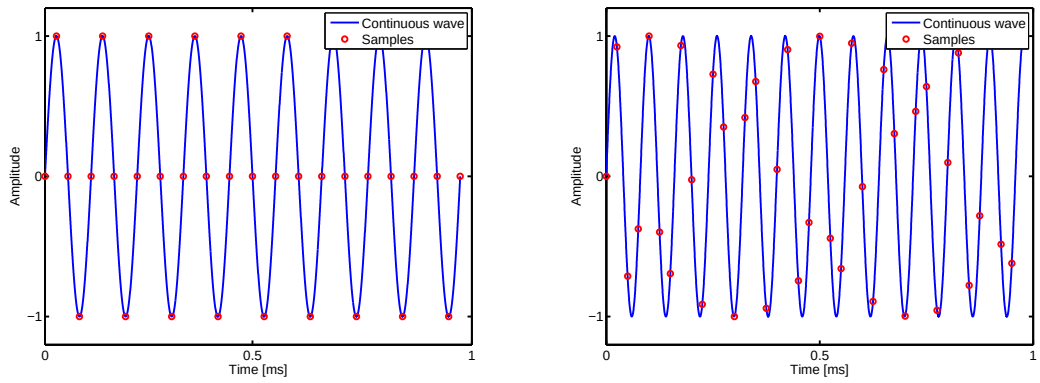



Figure 4.1: Comparison of the different L2C-code generation methods

In this work the zero-padded version (Method 2) of the L2CM-code is used for both acquisition and tracking. This approach enables an easy switch between L2CM- and L2CL-code in the future.
Galileo uses so-called long codes which are generated from primary and secondary codes of certain respective code lengths (Sect. 2.4). According to European Commission (2010) the primary spreading codes can be either truncated and combined maximum length sequences or randomly generated PRN sequences. The latter are stored in memory and therefore are often called memory codes. Both sequences can either be generated by a LFSR or are stored in the receiver's memory as well. The E1B and E1C primary codes are designed to be pseudorandom memory code sequences according to the hexadecimal representation provided in European Commission (2010). The E5 primary codes can be considered as memory stored binary sequences or can also be generated using linear feedback shift registers. Within the developed software all Galileo codes are considered to be memory codes and are loaded from memory. Note that Galileo in general, except E1B, uses secondary codes which have fixed sequences and are defined in European Commission (2010). They can be stored in memory as well.

### 4.1.2 Carrier generation

Two carrier replica signals with a phase difference of 90 degree, i.e., cosine and sine wave, have to be generated for the acquisition and tracking tasks. The frequency, which corresponds to the IF, must be adjustable due to the Doppler shift. Following Borre et al. (2007) the replica must have a length of one PRN code period (e.g., in case of GPS L1 C/A the length is 1 ms) and must be sampled with the sampling frequency depending on the used front-end. Due to the Doppler shift the generation needs to be flexible and, therefore, the discrete phase points have to be computed in advance. The argument of the sine and cosine function is obtained by multiplying the phase points with the IF, corrected by the Doppler shift. Within this implementation the C standard functions for sine and cosine have been used in a first step. For performance reasons the standard functions were replaced by approximations according to Press et al. (2007), providing an accuracy of $10^{-3}$ to $10^{-4}$. The provided accuracy is sufficient for the signal processing tasks. Figure 4.2 shows the first millisecond of sine waves generated at different sampling frequencies and IFs.




(a) (b)

Figure 4.2: Locally generated sine waves with different sampling frequencies ($f_s$) and IFs: (a) $f_s = 38.192$ MHz at IF $= 9.548$ MHz and (b) $f_s = 40.96$ MHz at IF $= 12.82$ MHz

### 4.1.3 Different acquisition methods

As mentioned in Sect. 4.1, beside the PRN number, dedicated to a specific space vehicle identification (SVID) number, the code phase (offset) $\delta\tau$ and the frequency offset (Doppler shift) $f_d$ have to be determined. Since the estimation of $\delta\tau$ and $f_d$ can only be achieved if the satellite signal of interest is present, the signal acquisition is commonly approached as a detection and estimation problem. The acquisition task is to provide rough estimates of the unknowns, whereas the refinement is performed during signal tracking. The acquisition process in general is based on a 2-D correlation function, called cross ambiguity function (CAF), which allows the satellite detection as well as the estimation of the unknown signal parameters ($\delta\tau$ and $f_d$) using the crosscorrelation properties of a specific PRN code sequence. The CAF represents an extension of the crosscorrelation function (CCF) defined in (2.8). The CAF can be formulated in the discrete time domain as follows:

$$Y(\tau; F_D) = \frac{1}{N} \sum_{n=0}^{N-1} r[n] c[n+\tau] e^{(-j2\pi F_D n)}, \tag{4.1}$$

where $r[n]$ represents the received signal, $c[n+\tau]$ denotes the local PRN code replica including the potential subcarrier and the secondary code. The parameters $\tau$ and $F_D$ denote the code phase (code delay) and the Doppler affected carrier frequency. In case of $\tau$ and $F_D$ matching the code delay and the Doppler shift of the received signal, the CAF shows a sharp peak. If no signal is present or if the code delay and Doppler shift are wrong no peak is visible. Noise and other impairments (e.g., multipath and interference) can degrade the sharp peak significantly, which can lead to a misdetection or to no detection at all. For detection the envelop of the CAF is considered only. The general scheme of an acquisition system can be divided into four functional blocks (Borio 2008):

- Computation of the cross ambiguity function;
- Computation of the CAF envelop and averaging;

- Detection and decision stage;
- Verification and fine frequency search (multi-trial and verification).

For the computation of the cross ambiguity function the received signal $r[n]$ is multiplied with two orthogonal sinusoidal waves (sine and cosine) at the frequency $F_D = (f_{IF} + f_d)T_s$. The two resulting signals can be written as:

$$\begin{aligned} Y_c(n; F_D) &= r[n] \cos(2\pi F_D n) \\ Y_s(n; F_D) &= -r[n] \sin(2\pi F_D n). \end{aligned} \tag{4.2}$$

Thus, the received signal is translated into baseband, removing the effect of the Doppler shift. These multiplications correspond to the complex modulation of (4.1) (Borio 2008). Thus the incoming signal is split into two branches (in-phase and quadrature). The normalized frequency

$$F_D = (f_{IF} + f_d)T_s = \frac{f_{IF} + f_d}{f_s} \tag{4.3}$$

is composed of the intermediate frequency $f_{IF}$ and the Doppler frequency $f_d$. In this case the IF frequency is known since it depends on the front-end architecture (Table 3.2). The Doppler shift, due to the relative motion between satellite and receiver, remains unknown. Thus $f_d$ is chosen from a finite set of values.

$$f_d = f_{d,\min} + l\Delta f \text{ with } l = 0, 1, \cdots, L-1. \tag{4.4}$$

Different Doppler frequencies (Doppler bins) have to be tested to determine the actual Doppler shift of the incoming signal. Typically a Doppler search range of $\pm 5$ kHz with a step width of $\Delta f = 250$ Hz is used. Borio (2008) states that the number of Doppler bins are determined are determined by the number of code periods used for the CAF computation. In a next step the two signals $Y_c(n; F_D)$ and $Y_s(n; F_D)$ are multiplied with the local code replica $c[n]$. Thereby $c[n]$ comprises the primary PRN code and potentially the subcarrier and secondary code. Assuming that the local replica is delayed by $\tau$ the multiplication result reads:

$$\begin{aligned} Y'_c(n; \tau; F_D) &= r[n] \cos(2\pi F_D n) c[n + \tau] \\ Y'_s(n; \tau; F_D) &= -r[n] \sin(2\pi F_D n) c[n + \tau]. \end{aligned} \tag{4.5}$$

Like the Doppler frequency, the code phase $\tau$ can vary depending on the distance between the satellite and the receiver. Thus different code phase (code phase bins) have to be tested. Typically delays are in the order of one chip are chosen from a predefined set

$$\tau = \tau_{\min} + h\Delta\tau \text{ with } h = 0, 1, \cdots, H-1. \tag{4.6}$$

By testing the different delays, starting with $\tau_{\min} = 0$, it is possible to estimate the delay of the received signal $r[n]$ by looking at the position of the correlation peak. Therefore the

signals $Y_c'(n;\tau;F_D)$ and $Y_s'(n;\tau;F_D)$ are then summed over time, leading to the in-phase and quadrature components $Y_I(\tau;F_D)$ and $Y_Q(\tau;F_D)$:

$$\begin{aligned} Y_I(\tau;F_D) &= \frac{1}{N}\sum_{n=0}^{N-1} Y_c'(n;\tau;F_D) \\ Y_Q(\tau;F_D) &= \frac{1}{N}\sum_{n=0}^{N-1} Y_s'(n;\tau;F_D). \end{aligned} \tag{4.7}$$

In (4.7), $N$ represents the number of samples used for evaluating the in-phase and quadrature components and defines the coherent integration time $T_C$.

$$T_c = NT_s \tag{4.8}$$

Usually $N$ is chosen as a multiple of the primary PRN code period. Note that the maximum code phase $H$ to be searched for can be different from $N$, since only a subset of all possible delays can be tested. Following Borio (2008), the two components of (4.7) represent the real and the imaginary parts of the CAF which reads

$$Y(\tau;F_D) = Y_I(\tau;F_D) + jY_Q(\tau;F_D). \tag{4.9}$$

Equation (4.9) is evaluated using different sets of Doppler frequencies (4.4) and code phases (4.6). This results in a two-dimensional grid, which is commonly referred to as the search space. As shown in Figure 4.3 the grid can be divided into cells, which are defined by a particular code phase $\tau$ and Doppler frequency offset $F_D$ .

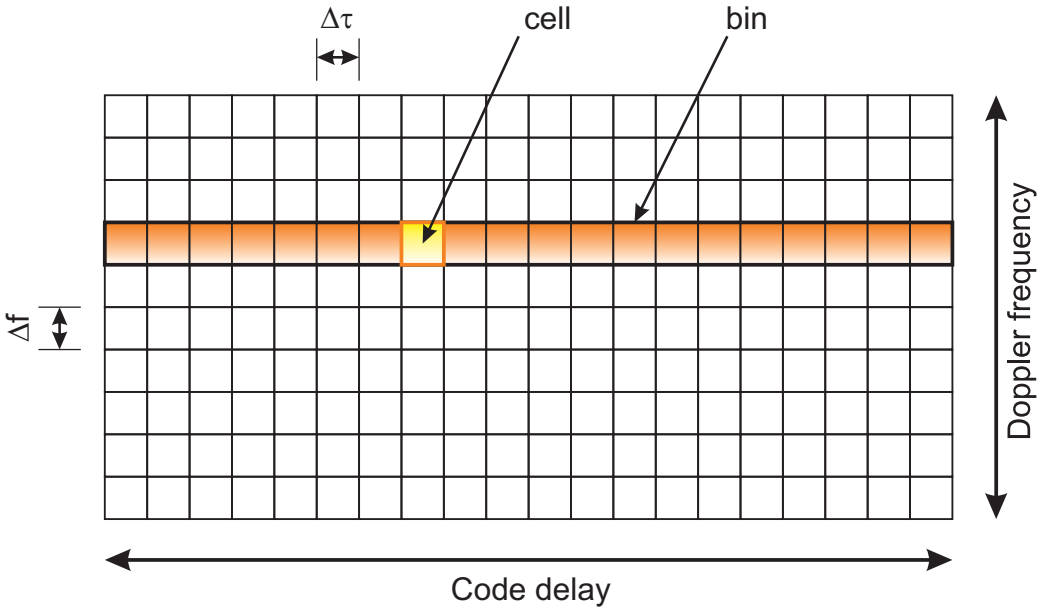



Figure 4.3: Two-dimensional search space

The size of the search space is variable and depends on the signal characteristics (e.g., code length), the type of acquisition performed (e.g., cold, warm, or hot start), and the

expected receiver dynamics (Doppler shift). Note that the number of cells influences the computational load of the acquisition task. The number of mathematical operations can be reduced by shrinking the search space. The correlation value of each cell is used as a test statistic, to decide whether the signal is present or not. Following Borre et al. (2007) three standard methods of evaluating the CAF exist:

- Serial search acquisition;
- Parallel frequency space search acquisition;
- Parallel code phase search acquisition.

The serial search method is one of the simplest and most frequently used acquisition methods (Misra and Enge 2006, Kaplan 2006, Borre et al. 2007). The algorithm performs two different steps: A frequency sweep over all different Doppler frequencies $f_d$ and a code phase sweep over all different code phases $\tau$. The algorithm steps through each cell and computes a correlation with the replica signal and the input signal.
Afterwards the dependence on the input signal phase has to be removed and noise reduction has to be applied. This can be achieved by considering the square absolute value (envelope) of the CAF (Borio 2008). The noise reduction (averaging) is performed before the envelope computation by the integration block. In case a value, of the computed envelope, exceeds a defined threshold the signal is declared present. Although this method is straightforward and implemented in most traditional receivers, the serial search is a very exhaustive method.
The parallel code phase search acquisition reduces the search to the frequency dimension by scanning the search space row by row (Borre et al. 2007). The parallel frequency space search acquisition in contrast scans the search space column by column and thus eliminates the necessity of searching all possible frequency bins. The two latter methods are based on the principle that a convolution of two signals in the time domain can be seen in frequency domain as the product of the Fourier transform of the two signals. The use of fast Fourier transform (FFT) algorithms can thus speed up the acquisition significantly. Note that a multiplication in frequency domain corresponds to a convolution in time domain. In the following the parallel code phase acquisition, which is implemented in the developed software-based receiver, will be discussed in more detail.

## 4.2 Parallel code phase acquisition

The advantage of this method is that by using a discrete Fourier transform (DFT) the correlation for an entire code phase dimension at a selected Doppler bin can be computed in a single step (Krumvieda et al. 2001). Compared to the serial method this method is faster and more efficient and therefore commonly used in software-based receivers (Charkhandeh 2007). The disadvantage of this method is that the convolved signal produces some errors in case the Doppler frequency of the replica signal is non-zero (Charkhandeh 2007).
The DFT, $\mathcal{F}\{.\}$, of a finite length sequences $x[n]$ with length $N$ is given following Oppenheim et al. (1999) and Tsui (2005) as follows:

$$X[k] = \mathcal{F}\{x[n]\} = \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}. \tag{4.10}$$

The inverse discrete Fourier transform (IDFT) $\mathcal{F}^{-1}\{.\}$ in contrast is given by

$$x[n] = \mathcal{F}^{-1}\{X[k]\} = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j2\pi kn/N}. \tag{4.11}$$

The circular crosscorrelation between two sequences $x[n]$ and $y[n]$ both having length $N$ is computed according to (2.9) as

$$z[n] = \frac{1}{N}\sum_{m=0}^{N-1} x[m]y[n+m]. \tag{4.12}$$

In case of $n+m \geq N$ the sequence $y[n+m]$ is shifted according to $(n+m)$ modulo $N$. In the following the scaling factor $1/N$ is omitted. Following Bertelsen et al. (2004) the discrete $N$-point Fourier transform of $z[n]$ is given by combining (4.10) and (4.12) and reads

$$\begin{aligned} Z[k] &= \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} x[m]y[m+n]e^{-j2\pi kn/N} \\ &= \sum_{m=0}^{N-1} x[m]e^{+j2\pi km/N}\sum_{n=0}^{N-1} y[m+n]e^{-j2\pi k[n+m]/N} \\ &= X^*[k]Y[k], \end{aligned} \tag{4.13}$$

where $X^*[k]$ denotes the complex conjugate of $X[k]$.
By applying (4.13) to $N$-points of the incoming signal the signal is converted into the frequency domain. This is achieved by multiplying the incoming signal with a locally generated carrier signal (sine and cosine) generating the I (in-phase) signal and the Q (quadrature) signal. The two resulting signals are then combined to form a complex input signal $q_l[n]$ to the DFT function. For each frequency bin $f_d$ the obtained sequence for the DFT reads:

$$q_l[n] = r[n]e^{(-j2\pi F_D n)}. \tag{4.14}$$

The replica code $c[n]$ is transformed into frequency domain using a DFT as well. The result is complex conjugated. Now it is possible to compute the crosscorrelation (cf. (4.1)) by multiplying the incoming signal's DFT $\mathcal{F}\{r[n]\}$ with the conjugate $\mathcal{F}\{c[n]\}^*$ of the reference signal's DFT. The result of this multiplication is then transformed into the time domain again by an IDFT

$$\tilde{Y}(\tau, F_D) = \mathcal{F}^{-1}(\mathcal{F}\{q_l[n]\}\mathcal{F}\{c[n]\}^*). \tag{4.15}$$

In a last step the acquisition has to remove the dependency on the input signal phase (Borio 2008). This is achieved by computing the square of the absolute value of (4.15). This

square of the absolute value $S(\tau; F_D)$ of the inverse Fourier transform output represents the correlation between the input signal and the replica PRN code. It is computed as follows:

$$S(\tau; F_D) = \tilde{Y}_I^2(\tau, F_D) + \tilde{Y}_Q^2(\tau, F_D). \tag{4.16}$$

The Doppler frequency bins are chosen to be 250 Hz apart and thus, the maximum frequency error is ±125 Hz. Since a frequency accuracy of 125 kHz would be too coarse for the tracking a refinement has to be done. This is achieved via a post-correlation FFT approach (Borre et al. 2007). This is usually done after the signal is declared present (Sect. 4.5). According to Bertelsen et al. (2004) the code phase can be determined with sampling precision, whereas the serial search acquisition method typically has a precision of ½ chip. Figure 4.4 shows a graphical representation of the parallel code phase search algorithm.

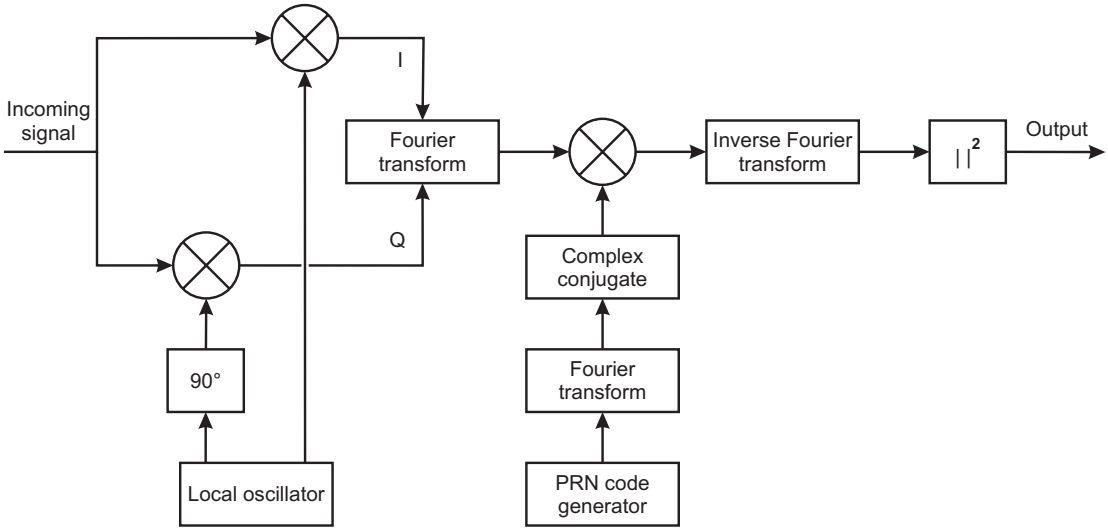



Figure 4.4: Parallel code phase search acquisition algorithm

The next task is to decide if the signal is present or absent, e.g., if there is a correlation between the incoming signal and the replica signal. Therefore the squared absolute values $S(\tau; F_D)$ are used within the detection and decision step to evaluate if the signal is present or not.

## 4.3 Detection and decision

The detection and decision process can be formulated as a statistical process because each cell either contains noise with the signal absent or noise with the signal present. The evaluated search space $S(\tau; F_D)$ represents a matrix of random cells $X_n$ with $n = 1, 2, \ldots, M = H \cdot L$ elements. The evaluation of these random cells is based upon hypothesis testing using a null hypothesis $H_0$ and an alternative hypothesis $H_1$. The distribution of the cells $X_n$ is given by (Borio 2008)

$$X_n | H_0 \sim f_{X_n}(x), \tag{4.17}$$

assuming the null hypothesis $H_0$, which corresponds to the signal not being present. The alternative hypothesis $H_1$ represents the logic negation of $H_0$ and, thus, corresponds to the signal being present. A detector is used to decide if the signal is present $D_1$ or not $D_0$. Following Mongrédien (2008) four possible results can occur as shown in Table 4.1.

Table 4.1: Possible results of hypothesis testing

| | Accept $H_0$ | Accept $H_1$ |
|---|---|---|
| $H_0$ is true | Correct decision<br>Probability: $1-\alpha$, $P_R$ | False alarm (Type I error)<br>Probability: $\alpha$, $P_{fa}$ |
| $H_1$ is true | Missed detection (Type II error)<br>Probability: $\beta$, $P_{MD}$ | Correct decision<br>Probability: $1-\beta$, $P_D$ |

The parameter $\alpha$ represents the level of significance of the test, whereas $\beta$ denotes the power of the test. Both, the null hypothesis as well as the alternative hypothesis have their own probability density functions (PDFs) (Kaplan 2006). The PDF in case of $H_0$ (noise with no signal present) is denoted as $P_N(z)$, whereas the PDF for noise with the signal present ($H_1$), is denoted as $P_S(z)$, both having a nonzero mean. The two important probabilities for the detection process are the detection probability $P_D$, and the false alarm probability $P_{fa}$. The false alarm probability on a single cell is given by

$$P_{fa}(\gamma) = \int_{\gamma}^{\infty} f_{X_n}(x)dx, \tag{4.18}$$

where $\gamma$ denotes the defined threshold. Following Borio (2008) the distribution of a random variable under the alternative hypothesis $H_1$ is given by

$$X_n|H_1 \sim f_A(x) \tag{4.19}$$

and the corresponding detection probability is given by

$$P_D(\gamma) = \int_{\gamma}^{\infty} f_A(x)dx. \tag{4.20}$$

The detection probability $P_D$ denotes the probability of a signal being detected correctly. If the test statistic is higher than the threshold but the code phase and/or Doppler frequency estimate is wrong, then the false alarm situation occurs. The probability of this case is denoted as false alarm probability, $P_{fa}$. The threshold is commonly based on an acceptable single trial probability of false alarm, $P_{fa}$ (Kaplan 2006). Figure 4.5 shows the probability density functions of $P_S$ and $P_N$ as well as the detection probability $P_D$ and the false alarm probability $P_{fa}$.

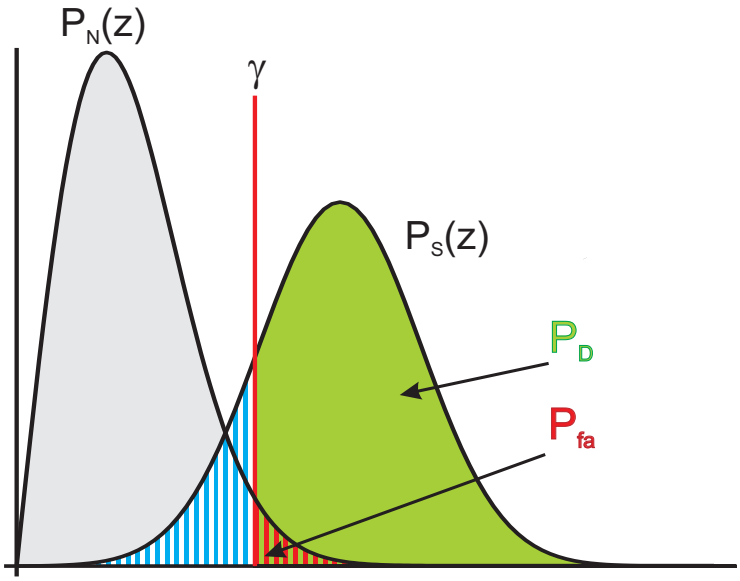



Figure 4.5: Probability density functions (Samad 2009)

The choice of the detection threshold influences the result significantly and must be done carefully. If $\gamma$ is set too low then the detection probability increases but so does the false alarm probability. If $\gamma$ is chosen too high the detection probability decreases. Thus, it is possible that the signal is not declared present although received by the antenna. The performance of this detector is evaluated by plotting the detection probability versus the false alarm probability and is called receiver operating characteristic (ROC) curve. As discussed in Borio (2008), the overall performance of the acquisition is not only defined by the test statistic used but also by the search strategy. Within the detection and decision stage each cell value is compared with the chosen threshold. If a cell value is detected above the chosen threshold $\gamma$ the signal is declared present. If all cell values are below the threshold no signal is detected. The decision to keep or discard the satellite PRN in the receiver channel is done by comparing the test statistic with a predefined threshold $\gamma$. The test statistic can be formed in different ways, e.g., as the value of the global maximum of the correlation output, as the ratio between the global maximum and the mean value of the correlation, or as the ratio between the global maximum and the next significant local maximum (Samad 2009, Jung 2004). More details on detection methods can be found in Borio (2008).
Within this development the correlation power peak ratio (CPPR) was used. The CPPR is defined as the ratio of the largest correlation peak to the second highest peak (Jung 2004). The test statistic for the comparison against the threshold is formulated by

$$10 \log_{10} \left( \frac{S_1(\tau, F_D)}{S_2(\tau, F_D)} \right) \geq \gamma, \tag{4.21}$$

where $S_1(\tau, F_D)$ denotes the highest correlation peak within the search space and $S_2(\tau, F_D)$ the second highest peak. The threshold is given in dB. The advantage of this approach is that, compared to the other methods, the detection threshold value is not based on the noise power and the distribution of the correlation power. Especially when there is

a change in receiver noise (e.g., different front-ends) a change of the threshold value is not necessary in case of using the CPPR method. Although it is possible to compute the receiver noise it requires additional system resources. Thus the CPPR detector was used within this implementation. Nevertheless the detection performance when taking the receiver noise into account would be improved. Details on the performance as well as the analytical formulation of the probability density functions can be found in Geiger et al. (2010) and Geiger et al. (2012). The detection threshold values were chosen empirically as shown in Table 4.2.

Table 4.2: Summary of the different detection thresholds used in the acquisition process

| Signal | Detection threshold $\gamma$ [dB] |
|---|---|
| GPS L1 C/A | 3.5 |
| GPS L2C | 2.5 |
| Galileo E1B | 3.5 |

Note that in case of GPS L1 C/A acquisition, Borre et al. (2007) and Jan and Sun (2010), use the same threshold value as the corresponding value in Table 4.2.
In case of BOC modulation the autocorrelation function has no longer a unique peak but contains side peaks. In Figure 4.6 the autocorrelation functions of a BPSK(1) and BOC(1,1) signal are shown. The BOC(1,1) correlation function shows one main peak at zero lag and two side peaks with half power of the main peak on each side.

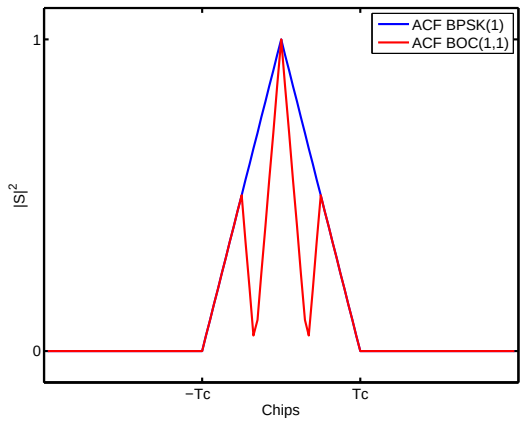



Figure 4.6: Autocorrelation function of a BPSK(1) and BOC(1,1) chip

Thus in case of the CPPR detector the second largest peak has to be searched more than 1-chip away from the main correlation peak. Note that this method might lead to misdetection in presence of multipath by acquiring the wrong peak. The problem of false peak tracking is addressed in Chap. 5 and a possible solution is presented. Several other techniques have been developed to change the shape of the autocorrelation function (i.e., removal of the side peaks) which improve the sensitivity and reliability of the acquisition (Yang et al. 2007).

## 4.4 Zero padding algorithm

Beside acquiring the wrong peak in case of BOC modulation another problem occurs when looking at the modulated data message. Since a data message is modulated onto the PRN sequence special focus must be put on the impact of changes in the sign of the data bit on the correlation function. Due to this possible change in sign, a destructive combination can occur over longer coherent integration times, leading to degraded acquisition performance or a total destruction of the correlation (Mongrédien 2008). In case of the GPS L1 C/A signal a data bit sign can change every 20 ms, thus the correlation can use 1 to 10 ms (i.e., 1 to 10 PRN code periods) of incoming signal without affecting the correlation. If a correlation is done over 10 ms and a change of the sign of a data bit occurs, then the next 10 ms must be used.
In case of Galileo E1B, for example, one PRN code has a duration of 4 ms and a change in the data bit sign can occur every 4 ms (Sect. 2.4). This increases the possibility of a destructive correlation. The same problem is encountered in case of secondary codes, because the sign of the secondary bit (like the data bit) can change each time the spreading code repeats. This situation occurs in case of GPS L2C, Galileo E1C, E5a-I, E5a-Q, E5b-I, and E5b-Q signals. To avoid a destructive correlation the zero padding technique is applied to these signals (Mongrédien 2008, Yang et al. 2004). In case of Galileo E1B, this method uses 8 ms of the incoming signal and one period of the local replica (4 ms) padded by 4 ms of zeros to compute the correlation. One or two correlation peaks will be generated depending on whether a change in the sign of the data bit or secondary bit occurs or not. According to Mongrédien (2008) this ensures that a full 4 ms correlation peak will be found in the first four milliseconds of the correlation result. Thus only the first half of the correlation result needs to be searched. The first peak will always be higher or of same amplitude as the second peak, because it corresponds to the correlation of an entire code period (Macchi 2010). Figure 4.7 shows an illustration of the zero padding technique.

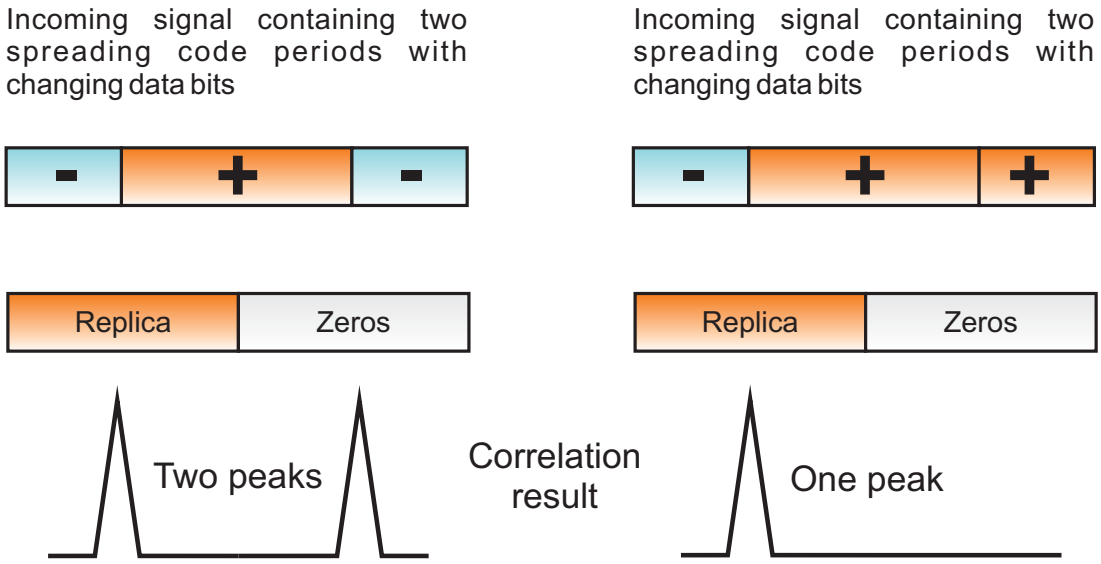
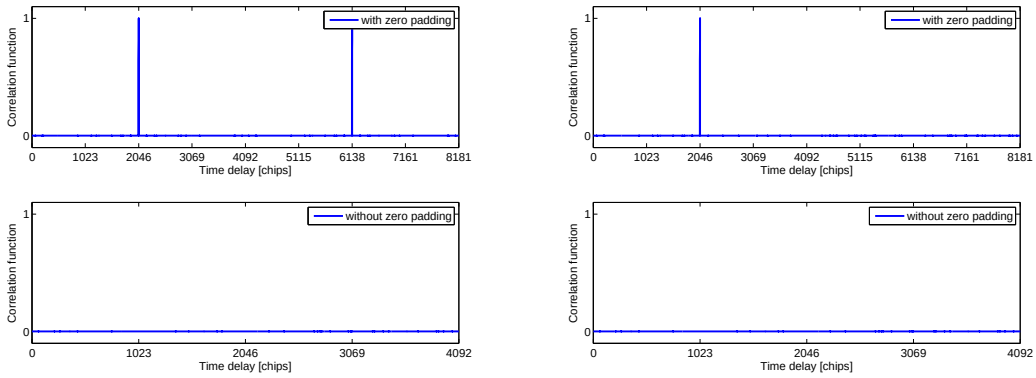



Figure 4.7: Schematic view of the zero padding technique

The left part shows an entire spreading code (red) in the middle of the incoming signal which is correlated with the local replica appended with zeros. This generates a first peak in the first half of output. Because the sign of the data bit of the incoming signal is the same on the left and the right (blue), the two portions of spreading code add during (cyclic) correlation and a second peak of similar size is produced in the squared absolute value domain (i.e., positive peaks only). In the right part of Figure 4.7 the peak in the first half of output is generated in the same way. Due to the fact that the sign of the data bit is not the same at the beginning and at the end the correlation will be destructive. If the left and right part are of the same size no peak will be visible, otherwise the second peak will be small compared to the first. Figure 4.8 shows an example of the zero padding technique. In the lower plots in Figure 4.8, 4 ms of incoming signal, generated by the GIPSIE®–IFS, have been correlated with 4 ms of E1B replica code. The data bit transition occurs right in the middle of the incoming signal and leads to a destructive result. By applying the zero padding technique (taking 8 ms of input signal and 4 ms of replica appended with 4 ms zeros), depending on the signs of the data bits, either two or one peak occur. In the upper plot of Figure 4.8a the signs of the data bits at the beginning and at the end are the same as described above, two correlation peaks are visible. In the upper plot of Figure 4.8b the signs of the data bits at the beginning and at the end are different, leading to destruction and thus only one peak is visible.




(a) (b)

Figure 4.8: Comparison of different data bit transitions and their influence on the correlation result.

Figure 4.9 shows the result in case the data bit transition does not occur right in the middle of the incoming 4 ms signal. In this case both methods show a correlation peak.

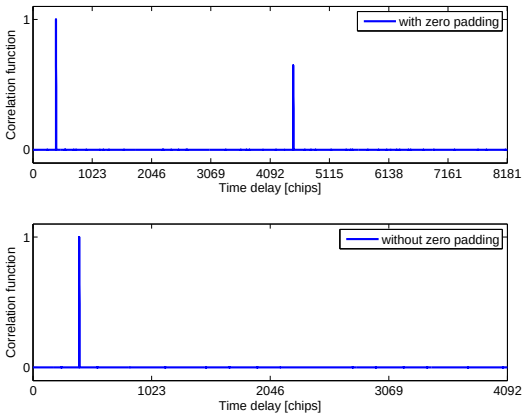



Figure 4.9: Comparison of zero padding method for a data bit transition at 400 chips

More details on the zero padding technique can be found in Yang et al. (2004), Mongrédien (2008), and Macchi (2010).

## 4.5 Fine frequency estimation

The acquisition process described above provides only a coarse estimate of the Doppler frequency. The accuracy of the coarse estimation depends on the used frequency step width. For tracking the Doppler frequency must be known within a few Hertz due to the limited bandwidth of the tracking loop (Chap. 5). For a fine estimation of the Doppler frequency the integration period has to be increased. Akopian et al. (2002) proposed a method to obtain a fine resolution of the Doppler frequency based on the coarse estimate using the acquisition technique described previously. The coarse values of the Doppler frequency and the code phase are used to remove the code. The resulting signal samples are squared to remove the navigation data and are then integrated. Finally a FFT is performed to get a fine estimate of the Doppler frequency. In this case the number of signal samples (integration time) defines the fine resolution

$$\Delta f_{\text{fine}} = \frac{f_s}{N}. \tag{4.22}$$

Assuming one C/A-code period (1 ms) and a sampling frequency $f_s = 40.96$ MHz leads to $N = 40\,960$ number of samples. The FFT needs a length of a power of two where the remaining (unused) samples are filled with zeros, thus, 65 536 samples are needed in total. This leads to a fine resolution of $\sim$ 312 Hz. In case 10 C/A-code periods are used the same calculation leads to a fine resolution of $\sim$ 39 Hz. The implemented algorithm uses 10 spreading code periods in case of GPS L1 C/A to determine the fine resolution of the Doppler frequency. Due to the code length of the L2C spreading code and the limitation in memory only 2 spreading code periods have been used, enabling a fine resolution of $\sim$ 10 Hz. For Galileo 5 E1B spreading code periods have been used, leading to a fine resolution of the Doppler frequency of $\sim$ 20 Hz.

## 4.6 Acquisition results

The tasks described above are repeated for all possible satellite PRNs of the desired system and frequency. Thus, in case of GPS 32 possible PRNs, in case of Galileo 30 possible PRNs have to be tested in order to determine the visible satellites. The acquisition module was tested and verified using different IF data sets. For verification purposes the GIPSIE® simulator as well as recorded real-world data have been used. The simulator is capable of simulating GPS L1 C/A and Galileo E1 signals, while the recordings contain GPS L1 C/A, L2C and GIOVE-A signals. In case of verifying the acquisition module the computed Doppler frequency and code phase are compared with the values simulated by GIPSIE®. For more information on the simulated scenarios and recorded data sets refer to Chap. 8. The simulated combined GPS and Galileo scenario contains 8 GPS satellites and 8 Galileo satellites. The acquisition was performed for both GPS L1 C/A and Galileo E1B. The results of the performed GPS L1 C/A acquisition are summarized in Table 4.3. For all simulated GPS satellites the correlation power peak ratio values are above the predefined threshold and thus the signal is declared present. The comparison of simulated and estimated Doppler values shows a good accordance in the range of ±40 Hz, as expected. The code phase could be estimated within sub-chip accuracy compared to the simulated values.

Table 4.3: GPS L1 C/A signal acquisition results compared with the simulated values by GIPSIE®–IFS

| PRN | CPPR | Doppler [Hz] | | Code phase [chips] | |
|---|---|---|---|---|---|
| | | estimated | simulated | estimated | simulated |
| 1 | 6.82 | +1721 | +1740 | 433.18 | 433.20 |
| 4 | 7.43 | +1089 | +1110 | 206.37 | 206.40 |
| 11 | 7.82 | +93 | +64 | 455.60 | 455.54 |
| 13 | 9.36 | +1138 | +1190 | 449.35 | 449.31 |
| 14 | 5.69 | +2231 | +2236 | 231.30 | 231.35 |
| 22 | 5.35 | −2828 | −2857 | 762.68 | 762.73 |
| 23 | 8.77 | −77 | −53 | 263.11 | 263.09 |
| 30 | 4.50 | +3475 | +3505 | 929.03 | 929.11 |

Figure 4.10 shows the acquisition search space in case of a successful acquisition of GPS PRN number 14. The acquisition peak is clearly visible and above the threshold. The corresponding code phase bin and Doppler bin are projected into the XZ and YZ plane of the plot. Note that the Doppler search space was increased to ±10 kHz instead of ±3.5 kHz for testing purposes.

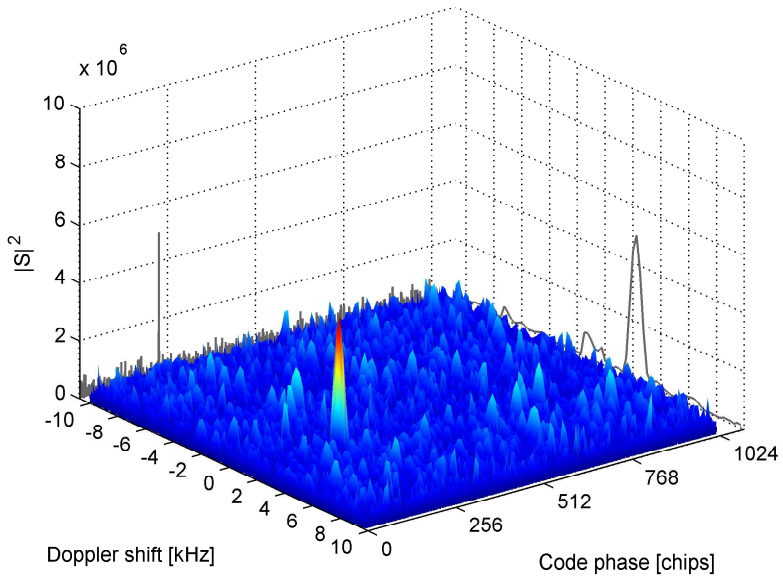



Figure 4.10: Acquisition result in case of acquiring GPS L1 C/A signal of PRN 14

In case of searching for a PRN which is not present in the incoming signal, no acquisition peak is visible. Figure 4.11 shows the acquisition result of satellite PRN number 3 which is not present in the incoming signal. In this case no distinct peak is visible and the CPPR value of 1.3 is far below the threshold of 3.5.

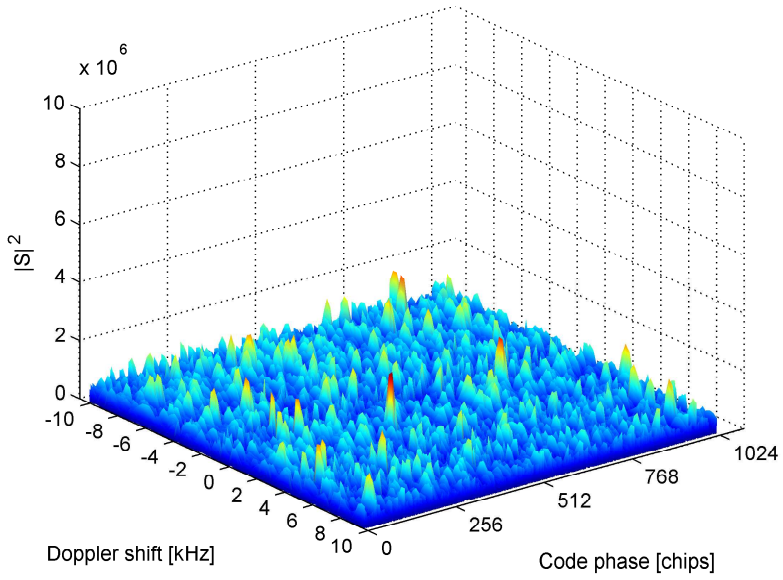
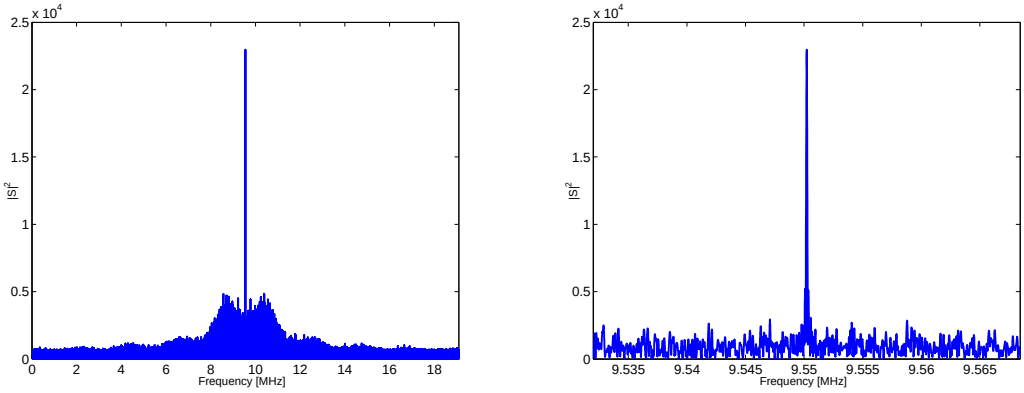



Figure 4.11: Acquisition result in case of acquiring GPS L1 C/A signal of PRN 3

The fine resolution of the Doppler frequency was accomplished using 10 spreading code periods. The frequency spectrum in Figure 4.12a shows a peak close to the center frequency.

The zoomed area in Figure 4.12b reveals the actual frequency.




(a) (b)

Figure 4.12: Result of the fine frequency search for GPS satellite PRN 14; (a) shows the whole spectrum, while (b) shows the correlation peak in detail

In case of Galileo the software is able to acquire the E1B signal as well as the E1C signal. Since for the position computation the navigation message is needed, the acquisition is accomplished using the E1B spreading code. Due to the possible occurrence of data bit transitions 8 ms of incoming signal are used to apply the zero padding technique for the proper acquisition. The acquisition result is based on the first 4 ms. For the fine resolution five spreading code epochs, allowing a fine resolution of $\sim$ 20 Hz, have been used. Table 4.4 shows a comparison of the computed acquisition parameters with the simulated values. The Doppler values match the expected accuracy and the code phase estimate is, like in the previous case, within sub-chip accuracy. Note that for an easier distinction of the PRNs between Galileo and GPS, the Galileo PRNs, originally numerated from 1 to 92, are numbered from 62 to 153 in this thesis.

Table 4.4: Galileo E1B signal acquisition results compared with the simulated values by GIPSIE®–IFS

| PRN | CPPR | Doppler [Hz] | | Code phase [chips] | |
|---|---|---|---|---|---|
| | | estimated | simulated | estimated | simulated |
| 64 | 5.69 | +2491 | +2472 | 1481.14 | 1481.13 |
| 65 | 5.99 | −291 | −309 | 2286.74 | 2286.73 |
| 66 | 6.05 | −2181 | −2163 | 1326.33 | 1326.33 |
| 71 | 5.59 | +2321 | +2341 | 3285.64 | 3285.63 |
| 72 | 5.47 | +597 | +614 | 1511.68 | 1511.65 |
| 73 | 5.44 | −1598 | −1616 | 1266.00 | 1266.01 |
| 80 | 5.26 | +2023 | +2009 | 1580.25 | 1580.27 |
| 88 | 5.21 | +2768 | +2759 | 3719.96 | 3719.97 |

Figure 4.13 shows only a part of the search space in case of a successful acquisition. The

correlation peak is clearly visible. Two side peaks appear due to the BOC modulation (Sect. 2.2.2). In this case the side peaks have to be excluded when searching for the second largest peak.

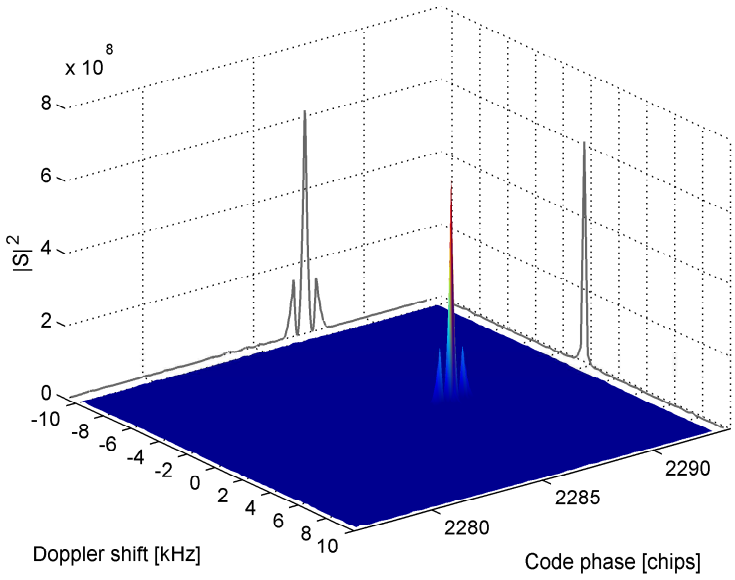
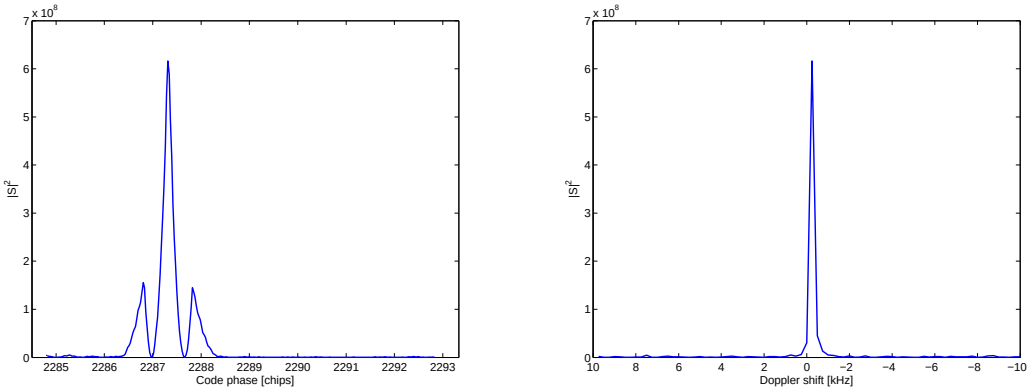



Figure 4.13: Acquisition result in case of acquiring Galileo E1B signal of PRN 65

The code phase bin and the Doppler bin, of the acquisition of Galileo PRN 65, containing the maximum are shown in Figure 4.14. The two side peaks appear within one chip on each side of the main peak. The fine resolution of the actual frequency is shown in Figure 4.14b.

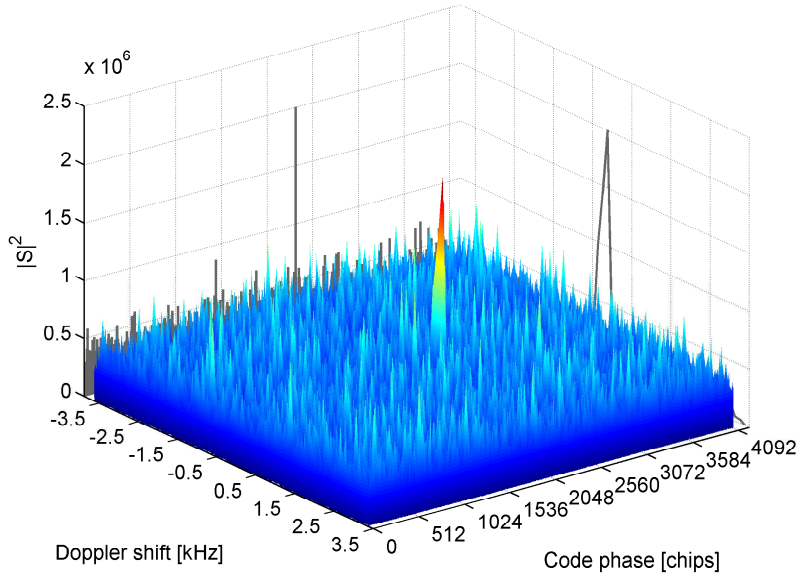



(a) (b)

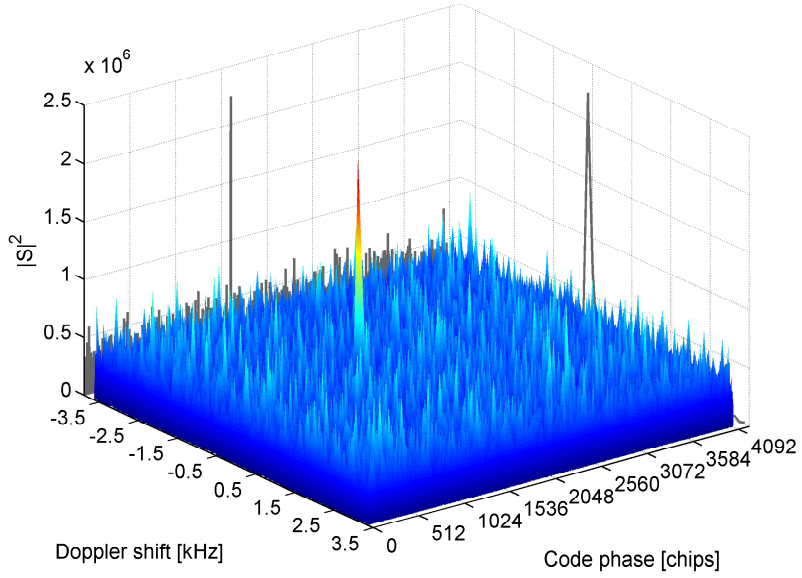
Figure 4.14: Visualization of the code phase bin (a) and Doppler bin (b)

In 2005 the first Galileo In Orbit Validation Element (GIOVE), named GIOVE-A, was launched. Two years later, in 2007, a second test satellite, called GIOVE-B, was launched. These two satellites are transmitting a Galileo-like signal. A data set was recorded on May

5th 2010 at 4 a.m. using the SiGe GN3S-v2 sampler front-end containing both GIOVE-A and GIOVE-B signals. The acquisition was performed using the GIOVE-A and GIOVE-B E1B spreading codes (available in European Space Agency 2008). Figure 4.15 shows the successful acquisition of the GIOVE-A and GIOVE-B satellite. In both cases the correlation peak is clearly visible and above the threshold.




(a)




(b)

Figure 4.15: Results of a successful GIOVE-A (a) and GIOVE-B (b) E1B signal acquisition

This proofs that the acquisition does not only work for simulated Galileo E1 signals but also with real-world data.
In a next step real-world data are used to test the acquisition module in case of GPS L2C signals. This task was accomplished using the Fraunhofer multi-band front-end. In the literature several algorithms exist for a combined acquisition of L1 and L2C. In this study the acquisition of different frequencies is treated independently. The acquisition of the L2C signal is computed using the zero-padded version of the L2CM code generated and sampled at $f_s$ and 20 ms $\times$ $f_s$ samples from the recorded IF data. When the L2CM-code acquisition had been successful, then the tracking process was started. Tracking is also performed on the L2CM-code only. When L2CL-code tracking is intended the L2CL-code has to be acquired as well. Muthuraman (2010) mentions that the L2CL-code search is restricted to only 75 different segments of the L2CL-code, due to the L2CM-L2CL-code synchronization. Table 4.5 shows a comparison of the GPS L1 C/A and L2C acquisition results as well as a the differences between the corresponding Doppler values $\Delta D_{f_{L1}f_{L2}}$ computed from 4.23.

Table 4.5: Comparison of the GPS L1 C/A and L2C acquisition results using real-world data

| PRN | CPPR | | Doppler [Hz] | | $\Delta D_{f_{L1}f_{L2}}$ [Hz] |
|---|---|---|---|---|---|
| | L1 | L2 | L1 | L2 | |
| 5 | 4.65 | 4.41 | −2656 | −2065 | −5.2 |
| 12 | 10.53 | 5.76 | −1289 | −996 | −8.4 |
| 21 | 5.40 | — | +1094 | — | — |
| 25 | 4.25 | 11.62 | +3007 | +2344 | +0.1 |
| 29 | 6.35 | 11.98 | −2607 | −2031 | −0.5 |
| 30 | 5.21 | — | -2002 | — | — |
| 31 | 9.38 | — | +986 | — | — |

As shown, only 4 out of 7 satellites are currently transmitting the L2C signal. Figure 4.16 shows the acquisition search space and the existing correlation peak in case of successful L2C signal acquisition of GPS PRN 12.

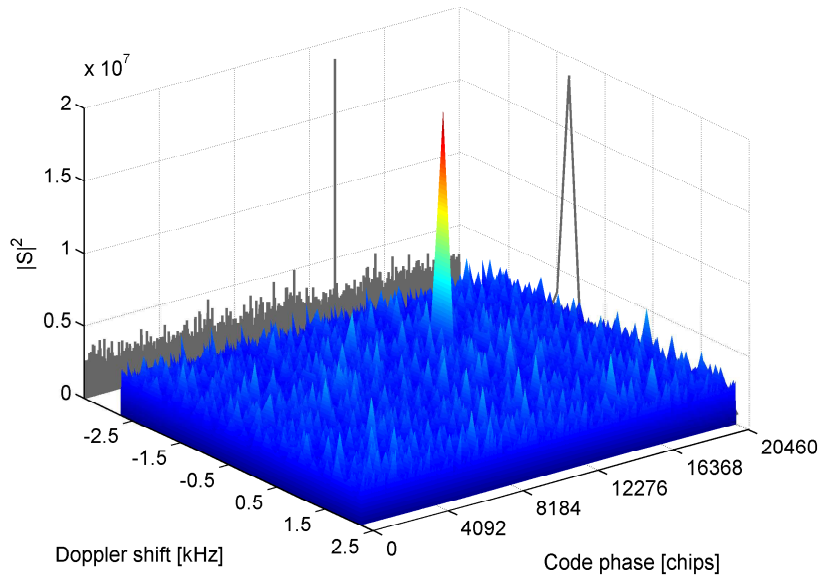



Figure 4.16: Successful L2C signal acquisition of GPS satellite PRN 12

The Doppler frequency estimates of L1 and L2C can be compared using the following expression

$$\Delta D_{f_{L1}f_{L2}} = D_{f_{L1}} \frac{f_{L2}}{f_{L1}} - D_{f_{L2}}. \tag{4.23}$$

In the ideal case, without interference or multipath present, the differences between the corresponding Doppler values, $\Delta D_{f_{L1}f_{L2}}$, must be equal zero. The remaining differences are mainly due to the resolution of the fine frequency estimation. Since the computational load for the L2C signal acquisition is high (especially the fine resolution estimation) compared to the L1 C/A signal acquisition, the relationship between the two frequencies is exploited for a faster L2C signal acquisition. In this case a Doppler frequency estimate for the L2C signal is computed by (4.23) using the estimated Doppler for L1 C/A. Thus, it is possible to shrink the Doppler search space significantly to one or two frequency bins respectively. The simulated scenarios, as well as the real-world recordings proofed that the implemented acquisition module is able to acquire GPS L1 C/A and L2C signals, as well as Galileo E1B signals. Additionally it was possible to acquire both GIOVE satellites.

# Chapter 5

# Carrier and code tracking

After a successful acquisition of a satellite signal, the next task is to refine the coarse estimates of the frequency and the code phase, keep track of the signal, and demodulate the navigation data. Due to the relative motion between satellite and receiver the unknown parameters (the carrier Doppler shift, the carrier phase, the PRN code Doppler shift, and the code phase) of the incoming signal are functions of time. Therefore it is important to track these parameters over time to end up with accurate position estimates. Whereas the acquisition performs a search of the parameters in a wide range (search space), the tracking module focuses its operation to the neighborhood of the previous estimates. This is done by using so-called tracking loops. The purpose of the tracking loops is to 'lock on' the corresponding parameter by continuously adjusting the local replicas to match the received signal. The time-delay (code phase) is typically tracked by a delay locked loop (DLL). A DLL performs, generally speaking, the same correlation operation as has been done in acquisition, but with reduced complexity due to the reduced parameter space. Commonly phase locked loops (PLLs) and frequency locked loops (FLLs) are used to track phase and Doppler shift. Within conventional receivers dedicated hardware structures are used to fulfill this task. These structures are also used in software-based receivers (Borre et al. 2007, Tsui 2004). Within this development standard implementations of tracking loops (i.e., Costas loop) have been coded in software. Different tracking strategies can be found in the literature, whereas the recent strategies are targeting the combined tracking of multiple signals to cope with difficult signal conditions. Extended Kalman filter strategies or vector tracking loops with inertial sensor aiding have been developed to enhance the tracking capabilities. After a short introduction into tracking loops, the basic demodulation scheme for obtaining the navigation data is presented in this chapter. Thereby, the functionality and the implementation of a DLL and of a PLL are discussed in more detail. The difficulties of BOC tracking are highlighted and a common solution is presented. At the end of this chapter selected results are presented.

## 5.1 Introduction

As mentioned above, one task of the tracking module is to obtain the navigation data modulated onto the carrier wave and PRN code. This task is commonly referred to as de-spreading. Figure 5.1 shows a basic demodulation scheme, which is used to demodulate the navigation message. Following this scheme the input signal is multiplied with a carrier replica, taking the actual Doppler frequency shift into account. This is done to transform the IF signal into baseband. Afterwards the remaining signal is multiplied with a code

replica, accounting for the actual code phase. The integration (correlation) is performed by the integrate and dump (I&D) function. The output of this correlation, after thresholding, represents the navigation data bits. Note that for an easier understanding the term bits is used when referring to the navigation data. Additional a distinction between bits and symbols in case of FEC coding has to be done (Sect. 6.2). To fulfill this task the tracking

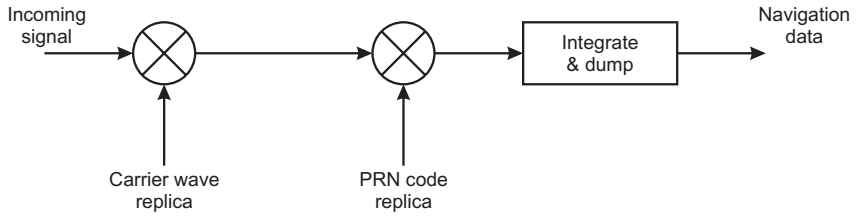



Figure 5.1: Basic demodulation scheme

module has to generate two replicas, one for the carrier and one for the code (Sect. 4.1.1 and Sect. 4.1.2). The following paragraphs describe the demodulation scheme for a signal only with one single satellite in order to reduce the complexity of the equations and to enhance the readability. In reality there exist signal contributions from all visible satellites. Considering a received GPS L1 signal from satellite $k$, after filtering and downconversion, the signal reads

$$s^k(t) = \sqrt{2P_C}C^k(t)D^k(t)\cos(\omega_{\mathrm{IF}}t) + \sqrt{2P_{P_{L1}}}P^k(t)D^k(t)\sin(\omega_{\mathrm{IF}}t), \tag{5.1}$$

where $P_C$ denotes the power for the C/A-code sequence $C^k(t)$ and $P_{P_{L1}}$ represents the power of the P(Y)-code sequence $P^k(t)$, both downconverted to the intermediate frequency $\omega_{\mathrm{IF}}$. The navigation data are represented by $D^k(t)$. Due to the narrow bandpass filter around the center frequency, the P(Y) code is distorted after the A/D conversion. Thus, the second term of (5.1) can be described as noise $e(n)$ which cannot be demodulated (Borre et al. 2007). The reformulated discrete signal can be written as

$$s^k[n] = \sqrt{2P_C}C^k[n]D^k[n]\cos(\omega_{\mathrm{IF}}n) + e[n], \tag{5.2}$$

where $n$ represents the time index of the sampled signal and is given in steps of $1/f_s$. In a next step, in order to obtain the navigation data, the signal has to be converted to baseband. This is done by removing the carrier by multiplying the input signal with a carrier wave replica. The replica wave has to have the same frequency and phase as the incoming signal. In this case the multiplication of both reads

$$\begin{aligned} s^k[n]\cos(\omega_{\mathrm{IF}}n) &= C^k[n]D^k[n]\cos(\omega_{\mathrm{IF}}n)\cos(\omega_{\mathrm{IF}}n) \\ &= \frac{1}{2}C^k[n]D^k[n] + \frac{1}{2}.\cos(2\omega_{\mathrm{IF}}n)C^k[n]D^k[n]. \end{aligned} \tag{5.3}$$

The first term represents the navigation message multiplied with the PRN code sequence. The second term contains the carrier with twice the intermediate frequency and can be removed by applying a low pass filter (Borre et al. 2007). After filtering the remaining signal reads

$$\frac{1}{2}C^k[n]D^k[n]. \tag{5.4}$$

The removal of the code sequence $C^k[n]$ from the signal can be achieved by correlating the signal with a local replica $C^k_{\text{local}}[n]$ of the code. The output of the correlation is given by

$$\sum_{n=0}^{N-1} C^k[n] D^k[n] C^k_{\text{local}}[n] = N D^k, \tag{5.5}$$

prevailing $C^k[n] = C^k_{\text{local}}[n]$. The term $ND^k$ contains the navigation bits multiplied by the number of points in the signal $N$.
The described technique shows only the demodulation for a signal with one single satellite. In reality the received signal is a sum of the signals from all visible satellites. Following Haykin and Veen (2005) and Borre et al. (2007) this does not change the presented method but results in larger noise terms.

## 5.2 Generic tracking loop

Both, the carrier tracking and the code tracking use an analytic linear phase lock loop model. It can be used to predict performance of tracking. The basic concept of a tracking loop is to generate a replica reference signal $x_{\text{ref}}[n]$, that matches an incoming signal $x[n]$ over time. Thereby, the incoming signal is a function of the variable under consideration. Depending on the different loops, FLL, PLL, or DLL, the variable corresponds to the Doppler frequency, the carrier phase, or the code phase, respectively. A block diagram of a generic tracking loop in the time-domain is shown in Figure 5.2. A digital input

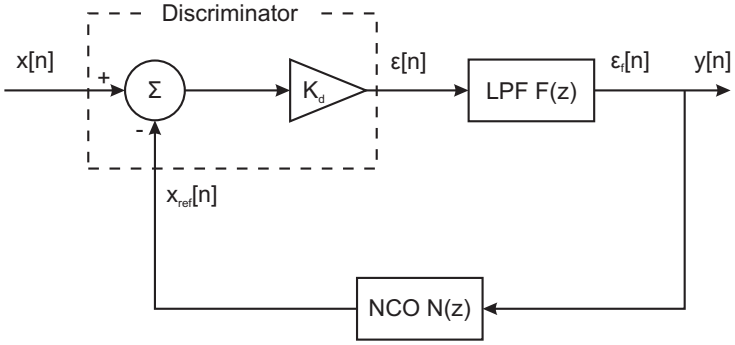



Figure 5.2: Generic tracking loop

signal $x[n]$ is fed into a discriminator which produces an error value $\epsilon[n]$. In this case the discriminator uses a simple addition with an associated gain $K_d$. Note that this is only a simple representation of a discriminator, whereas actual GNSS discriminators are more complex as shown later. The output of the discriminator is then low-pass filtered to reduce the noise. In a next step the resulting filtered value $\epsilon_f[n]$ is used by a numerically controlled oscillator (NCO) to produce a new replica signal for the next iteration of the loop. The error can thus be considered as the difference between the predicted and actual variable under consideration. The goal is to keep the output of the loop $y[n]$ as close to zero as possible. In this case the replica signal is equal to the input signal. As visualized in Figure 5.2 a generic tracking loop consists of several functional blocks:

- Loop discriminator block: Generates, based on the I&D output, the residual error of the estimated code phase, Doppler frequency, or carrier phase;
- Integrate and dump (I&D) block: The purpose of this block is to reduce the noise of the input signal by low-pass filtering (integrate) and to provide an output in predefined intervals (dump);
- Loop filter: The loop filter $F(z)$ reduces the noise to produce an accurate estimate of the incoming signal;
- Numerically controlled oscillator: Generates a local replica of carrier or code signal based on the loop filter output.

The order of the loop filter and the noise bandwidth determine the response to signal dynamics. Several different orders can be used. The most common are the first-order filter using a bandwidth of 0.5 Hz or the standard second-order loop filter. Figure 5.3 shows a first-order loop filter.

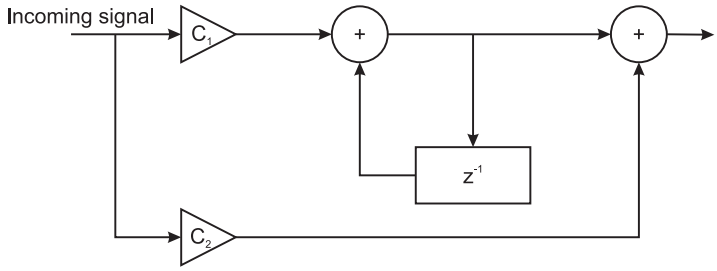



Figure 5.3: First-order loop filter (Borre et al. 2007)

To understand the behavior of such a loop it is important to understand the relation between input and output. This relation is described by the bilinear transfer function (Tsui 2005). Following Borre et al. (2007) the transfer functions for the digital filter $F(z)$ and the NCO $N(z)$ are given by

$$F(z) = \frac{(C_1 + C_2) - C_1 z^{-1}}{1 - z^{-1}}, \tag{5.6}$$

$$N(z) = \frac{K_0 z^{-1}}{1 - z^{-1}}, \tag{5.7}$$

where $K_0$ denotes the NCO gain. The goal is to find appropriate values for the coefficients $C_1$ and $C_2$ in the first-order filter transfer function. Therefore the transfer function of the digital loop

$$H(z) = \frac{K_d F(z) N(z)}{1 + K_d F(z) N(z)} \tag{5.8}$$

is used and compared with the analog transfer function (Tsui 2005). In (5.8) $K_d$ denotes the discriminator gain. Following Chung et al. (1993) the coefficients can be expressed by

$$C_1 = \frac{1}{K_0 K_d} \frac{8\zeta\omega_n T_s}{4 + 4\zeta\omega_n T_s + (\omega T_s)^2} \tag{5.9}$$

$$C_2 = \frac{1}{K_0 K_d} \frac{4(\omega_n T_s)^2}{4 + 4\zeta\omega_n T_s + (\omega T_s)^2}. \tag{5.10}$$

The order of the loop is determined by the denominator of the transfer function $H(z)$ which is a second-order function (Tsui 2005). Thus, the implemented tracking loops are second-order loops. In (5.9) and (5.10) the term $K_0 K_d$ denotes the loop gain, $\zeta$ the damping ratio, $\omega_n$ the natural frequency, and $T_s$ is the sampling time. The natural frequency in turn can be expressed as a function of the damping ratio and the noise bandwidth $B_L$ in the loop (Parkinson and Spilker 1996)

$$\omega_n = \frac{8\zeta B_L}{4\zeta^2 + 1}. \tag{5.11}$$

In general, the accuracy of a tracking loop depends on its ability to follow the errors (e.g., thermal noise, multipath, interference and user dynamics) (Macchi 2010). The damping ratio and noise bandwidth control the behavior of the loop. The settling time of the filter as well as the overshoot of the filter are defined by the damping ratio. Thus, the choice of the damping ratio is a compromise between overshoot and settling time. Following Tsui (2005), the damping ratio is typically chosen $\zeta = 0.7$, which is close to the optimum of $\zeta = 0.707$ (Gardner 2005). This results in a reasonable fast converging filter having not too much overshoot. The amount of noise in the filter is controlled by the noise bandwidth. Like the damping ratio this parameter has an influence on the settling time. The initial frequency, output by the acquisition, is usually off by some Hz (Sect. 4.5). The tracking loop is then trying to lock onto the correct frequency by minimizing the error value. In literature this process is often referred to as pull-in phase. If a large noise bandwidth is chosen the tracking loop will lock onto the real frequency but will generate a higher noise in the locked state. In contrast a smaller noise bandwidth will reduce the noise at the locked state but it will take longer until the right frequency is found. Even a false lock is possible in case of being not fast enough, due to a phase shift, to lock on the correct frequency. Typically a noise bandwidth is chosen of 20 Hz for land applications (e.g., pedestrian, automotive, leisure) (Borre et al. 2007). More details on derivations and on generic tracking loop design can be found in Borre et al. (2007), Parkinson and Spilker (1996), and Chung et al. (1993).

As discussed in Sect. 5.1 two local signal replicas - one for the carrier and one for the code - are needed to demodulate the navigation bits. Since the frequency and the code phase change permanently, two feedback loops are needed for generating the two replicas. The feedback loop for the carrier replica is denoted as the carrier tracking loop, and the feedback loop steering the code replica is referred to as code tracking loop.

## 5.3 Carrier tracking

The carrier tracking loop refines the carrier frequency and phase and keeps track of them. This is commonly done by using a phase locked loop, or a frequency locked loop. Most receivers start with frequency tracking using an FLL. After having achieved a lock onto the frequency the carrier tracking is switched to phase tracking using a PLL. Following Ward et al. (2006b) this is done because a FLL is less sensitive to dynamic stress and more robust under weak signal conditions (Kaplan 2006), while phase tracking loops are less noisy and, thus, more accurate. In order to enable precise pseudorange, integrated Doppler, and carrier phase measurements, a GNSS receiver must track the phase of the incoming signal carrier (Gleason and Gebre-Egziabher 2009). While the FLL removes the carrier frequency from the incoming signal, the PLL removes both the carrier frequency and phase. Within the PLL the numerically controlled oscillator generates a sinusoidal signal with a turnable frequency and phase. The incoming signal, still containing the spreading symbols, is first multiplied by a code replica from the DLL (Sect. 5.4). This wipes off the spreading symbols. The resulting signal is then multiplied by the locally generated carrier allowing the removal of the signal carrier. The signal passed to the integrate and dump block contains only a residual error of frequency and phase since the carrier wipe off is expected to follow closely the phase variations due to the change in signal propagation time between satellite and receiver (Julien 2005). To obtain the remaining phase error a phase discriminator is used. The estimated phase error is then filtered by a low-pass filter. The filter removes as much noise as possible from the discriminator output. The output of the filter is then used to control the NCO of the carrier generation. The drawback of using ordinary PLLs is that they are sensitive to 180° phase shifts (Tsui 2005). Such phase transitions are caused by the navigation data or secondary code chips. Those affect the polarity of the signal, which also implies a change in phase. Thus, a PLL which is used in GNSS signal processing has to be designed to be insensitive to 180° phase shifts in order to maintain the tracking sensitivity. Typically a Costas phase locked loop, as shown in Figure 5.4, is used for this task.

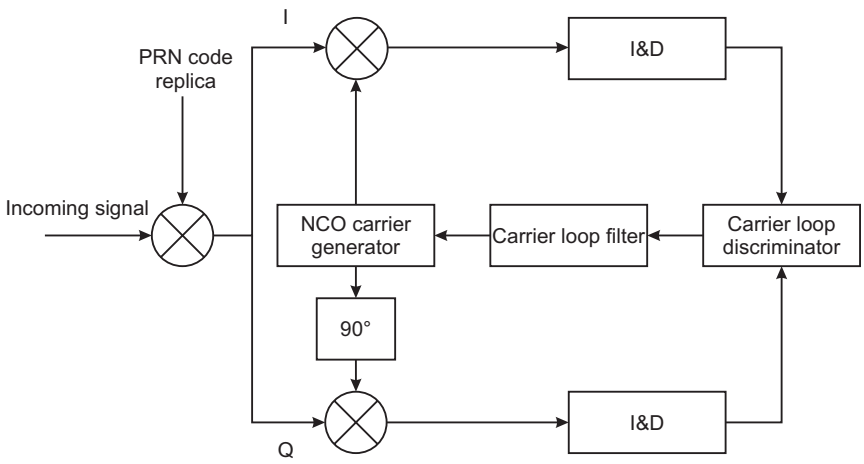



Figure 5.4: Costas tracking loop

After multiplication by the code replica the signal is split into two branches, compared with the generic tracking loop design (Figure 5.4). The upper branch, denoted as in-phase branch, represents the product between the input signal and the local carrier wave. The second (lower) branch contains the multiplication with a 90° phase-shifted carrier wave and is denoted as quadrature branch. Assuming a perfectly aligned carrier replica the multiplication of the in-phase ($I$) and quadrature ($Q$) branch reads (Borre et al. 2007)

$$\begin{aligned} D^k[k]\cos(\omega_{\mathrm{IF}}n)\cos(\omega_{\mathrm{IF}}n+\phi) &= \frac{1}{2}D^k[k]\cos(\phi)+\frac{1}{2}D^k[k]\cos(2\omega_{\mathrm{IF}}n+\phi) \\ D^k[k]\cos(\omega_{\mathrm{IF}}n)\sin(\omega_{\mathrm{IF}}n+\phi) &= \frac{1}{2}D^k[k]\sin(\phi)+\frac{1}{2}D^k[k]\sin(2\omega_{\mathrm{IF}}n+\phi). \end{aligned} \tag{5.12}$$

The phase difference between the phase of the incoming signal and the phase of the replica signal is denoted as $\phi$. After low-pass filtering the terms containing twice the intermediate frequency are filtered and, thus, the remaining two signals are

$$\begin{aligned} I^k &= \frac{1}{2}D^k[k]\cos(\phi) \\ Q^k &= \frac{1}{2}D^k[k]\sin(\phi). \end{aligned} \tag{5.13}$$

After the integration, dumping, and filtering the discriminator of a Costas loop uses the in-phase and quadrature-phase correlation values ($I^k$ and $Q^k$) to obtain the phase error. Different discriminator functions can be found in the literature. One of them can be formed by dividing $Q^k$ by $I^k$ and reads

$$\frac{Q^k}{I^k} = \frac{\frac{1}{2}D^k[k]\sin(\phi)}{\frac{1}{2}D^k[k]\cos(\phi)} = \tan(\phi). \tag{5.14}$$

The phase error or phase difference can thus be obtained by

$$\phi = \arctan\left(\frac{Q^k}{I^k}\right). \tag{5.15}$$

Possible discriminator functions are listed in Table 5.1.

Table 5.1: Different PLL discriminator functions

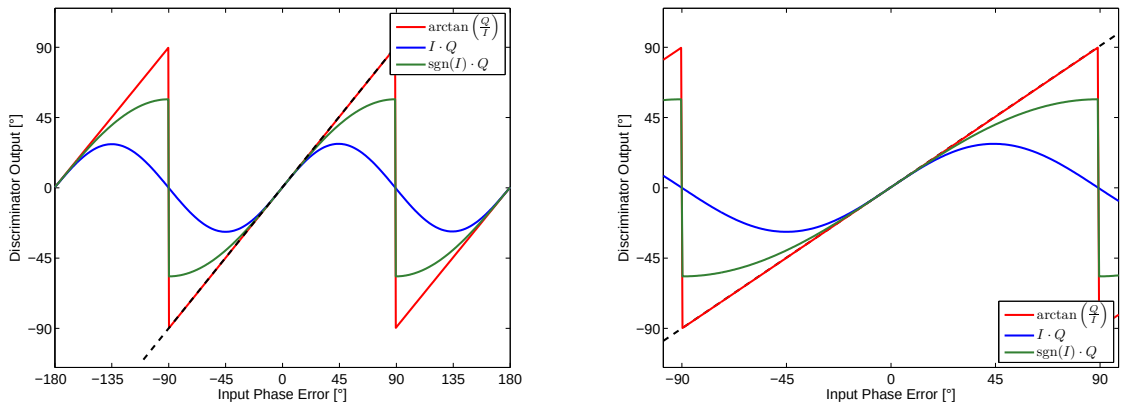
| Discriminator | Characteristics |
|---|---|
| $\mathrm{sgn}(I^k)Q^k$ | Output is proportional to $\sin(\phi)$ |
| $I^kQ^k$ | Output is proportional to $\sin(2\phi)$ |
| $\arctan\left(\frac{Q^k}{I^k}\right)$ | Output is the phase error |

The in-phase correlation values $I^k$ represent the navigation data bits. A preprocessing of the raw correlation values has to be done before the decoding of the parameters (Chap. 6).

The goal of phase tracking is to keep all the power in the in-phase component, which allows the decoding of the bits of the navigation message. Following Julien (2005) two main figures of merit can be used to characterize a discriminators inherent tracking capabilities:

- Stability region;
- Linear tracking region.

The stability region defines the region surrounding the zero phase error where a certain phase error input results in a discriminator output having the same sign as the input error. Thus, for a certain input error the discriminator will produce an output which leads to a phase error converging towards zero. The linear tracking region defines the region around the zero phase error where a phase error input results in a discriminator output which is equal to the input error, in this region the discriminator reacts almost perfectly. Figure 5.5 shows the response of different PLL discriminator functions depending on the true phase error. The black dotted line in the plot represents the linear tracking region. For true phase errors of 0° and 180° the output of all functions is zero due to the insensitivity to 180° phase shifts of the Costas loop.




(a) (b)

Figure 5.5: Different PLL discriminator functions (a) and zoomed area around the linear tracking region (b)

The maximum coherent integration time is limited to the duration of the data bit. For the GPS L1 C/A signal and the Galileo E1 data channel the coherent integration is limited to 20 ms and 4 ms respectively, due to a possible change of a navigation bit. Within this development the arctan discriminator as formulated in (5.14) is used. It is the most precise of the Costas discriminators, but also the most time-consuming one (Borre et al. 2007). These PLL characteristics are illustrated in Figure 5.6, where the phasor $A$ (the vector sum of $I^k$ and $Q^k$), tends to remain aligned with the I-axis and switches by 180° during each data bit reversal. The constellation diagram, sometimes also denoted as phasor diagram, will be used later to visualize the PLL tracking performance. Further details regarding PLLs can be found in Dierendonck (1996), Kaplan (2006), and Borre et al. (2007).

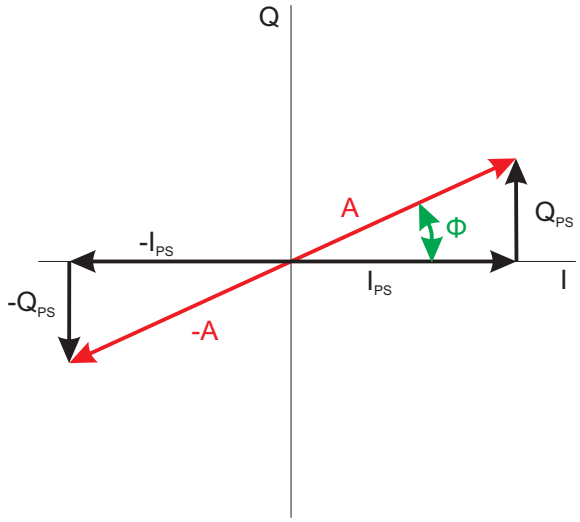



Figure 5.6: I and Q constellation diagram

## 5.4 Code tracking

The code tracking is commonly implemented using a DLL. The main purpose of the DLL is the refinement of the code phase estimate and tracking its changes by generating a local replica of the PRN code. The DLL tries to keep the code replica aligned over time with the signal received from the satellite. The code tracking uses the same principle as the phase tracking of the carrier. The main difference between the PLL and the DLL is the implementation of the discriminator function. While the PLL uses two branches (in-phase and quadrature-phase carrier replica) the code tracking loops use several branches (several multipliers, filters, and envelope detectors) to build the discriminator function. The DLL tracking loop starts with removing the carrier frequency by multiplying the incoming signal with the locally generated in-phase and quadrature-phase carrier replica, generated by the PLL. In this case it can be assumed that the remaining unknown is only the code phase (Giordanengo 2009). In a next step the remaining signal is multiplied separately by three code replicas. The three code replicas ($C_E$,$C_P$, and $C_L$) are nominally generated with a spacing of a fraction of a chip $\delta$ by the code generator NCO. The resulting signals are then passed to the I&D blocks, creating the in-phase and quadrature early, prompt and late ($I_E$, $Q_E$, $I_P$, $Q_P$, $I_L$, and $Q_L$) correlator outputs. The correlation output, assuming

no external distortions and no frequency uncertainty, can be written as

$$\begin{aligned} I_E &= \sqrt{\frac{A}{2}}\, R(\tau - \hat{\tau} - \delta)\cos(\Delta\Phi) \\ I_P &= \sqrt{\frac{A}{2}}\, R(\tau - \hat{\tau})\cos(\Delta\Phi) \\ I_L &= \sqrt{\frac{A}{2}}\, R(\tau - \hat{\tau} + \delta)\cos(\Delta\Phi) \\ Q_E &= \sqrt{\frac{A}{2}}\, R(\tau - \hat{\tau} - \delta)\sin(\Delta\Phi) \\ Q_P &= \sqrt{\frac{A}{2}}\, R(\tau - \hat{\tau})\sin(\Delta\Phi) \\ Q_L &= \sqrt{\frac{A}{2}}\, R(\tau - \hat{\tau} + \delta)\sin(\Delta\Phi), \end{aligned} \tag{5.16}$$

where $A$ denotes the signal power, $\delta$ is the correlation spacing (or chip spacing), $\tau - \hat{\tau}$ is the estimated code phase error, and $\Delta\Phi$ is the estimated phase error. These correlator results are then passed to the discriminator function and the loop filter to estimate the remaining code phase error (difference) between the incoming signal and the local replicas. In a last step these tracking errors are used in the NCO to update the code replicas. Using the in-phase and quadrature branches makes the DLL independent of the phase on the local carrier wave (Borre et al. 2007).

Figure 5.7 shows a basic example of how early, prompt, and late correlation values change depending on the relative code phase between incoming signal and replica code. For an easier understanding only one chip of a BPSK modulation and an ideal noise free scenario using a correlator spacing of $\delta = 0.5$ chips is shown. In this case the autocorrelation function has triangular shape (Sect. 2.2.1). If the replica codes are properly aligned, as shown in Figure 5.7b, the highest peak occurs at the prompt replica. The early and late replicas are equal in amplitude and the estimated code phase error is zero. If the replica codes are misaligned, then the early and late correlation values are unequal. Assuming that the replica codes are shifted by 1/2-chip early (Figure 5.7a), the maximum correlation value appears at the late replica. The early replica shows zero correlation and the prompt value is between the early and late replicas. The values are proportional to the amount of the code phase error. Thus, the code phase must be decreased, in this case, in a next step to get the proper alignment. Having equal correlation values for the early and prompt, and a lower value for the late, indicates that the code phase must be increased. In case of Figure 5.7c the replicas must be shifted by 1/4-chip. The code discriminator observes the difference in direction (early or late) and amplitude of the early and late correlation values and, thus, maintains the highest correlation peak at the prompt correlator.

Table 5.2 summarizes common discriminator functions and their main characteristics which are used for providing feedback to the code NCO. The choice of the discriminator is typically made depending on the signal environment and the computational capabilities of the receiver. Following Kaplan (2006) normalization removes the amplitude sensitivity and, thus, improves the receiver performance under rapidly changing S/N conditions but increases the computational load.

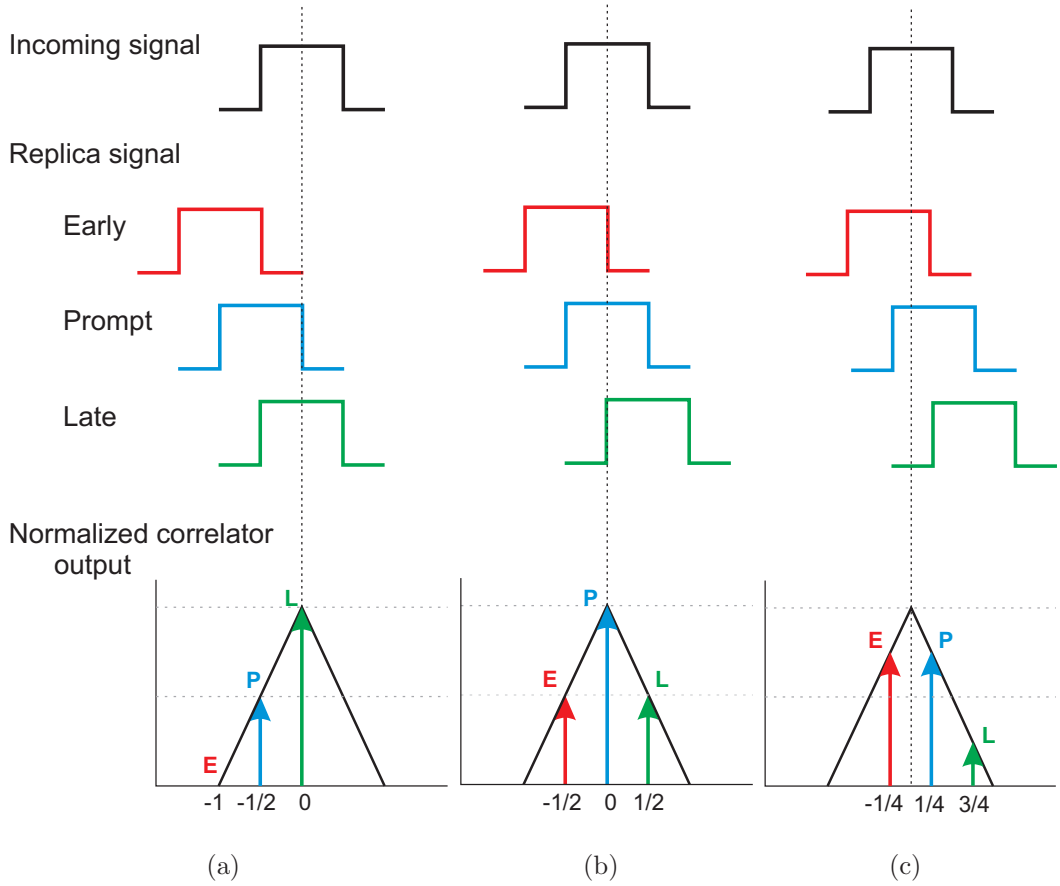



Figure 5.7: Code correlation principle: (a) replica code 1/2-chip early, (b) replica code aligned, and (c) replica code 1/4-chip late

Table 5.2: Various types of DLL discriminators

| Discriminator | Characteristics |
|---|---|
| $I_E - I_L$ | Early minus late; simple discriminator; does not require the Q branch, but a good carrier tracking |
| $(I_E^2 + Q_E^2) - (I_L^2 + Q_L^2)$ | Early minus late power; moderate computational load; same error performance as above |
| $\frac{(I_E^2 + Q_E^2) - (I_L^2 + Q_L^2)}{(I_E^2 + Q_E^2) + (I_L^2 + Q_L^2)}$ | Normalized early minus late power; moderate computational load; |
| $I_P(I_E - I_L) + Q_P(Q_E - Q_L)$ | Dot product power; uses all three correlators; low computational load; nearly true error output |

Figure 5.8 shows a comparison of different discriminator functions. The plots assume one chip spacing between the early and late correlators. Furthermore an ideal signal (no noise) is assumed.

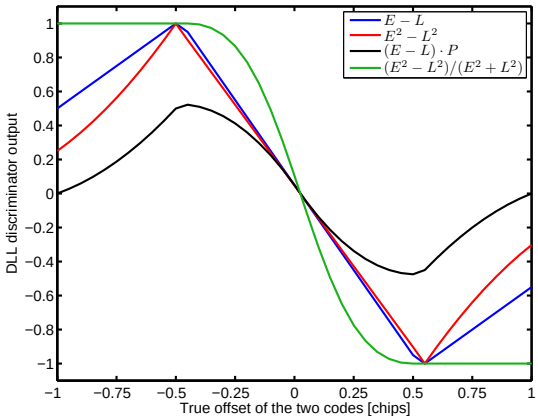



Figure 5.8: Comparison of different DLL discriminator functions ($\delta = 0.5$)

For the software-based receiver implementation, the normalized early minus late power discriminator function was selected. Following Borre et al. (2007) this discriminator is independent of the performance of the PLL since it uses both the in-phase and the quadrature branch. Another advantage, due to the normalization, is that the discriminator can be used under different signal strengths and under different S/Ns and, thus, different front-ends are possible. The correlation spacing $\delta$ between the early, prompt, and late codes determines the noise bandwidth in the delay locked loop. By increasing the correlator spacing the dynamics and the robustness of the loop can be increased. A smaller spacing will make the DLL more precise. For all BPSK signals a chip spacing of half a chip is commonly used.

Following Macchi (2010), the DLL is more affected by noise and multipath, compared to the FLL or PLL, while interference affects all types of loops. The length and chipping rate of the spreading code has a major influence on the impact of these errors on tracking. For example a higher chipping rate and a longer spreading code period reduce the impact of these errors. Thus, the modernized GNSS signals use higher chipping rates and longer spreading codes.

Figure 5.9 illustrates a general structure of the implemented tracking module, including the code tracking (blue) and the carrier tracking (red). The DLL part uses in total three different replicas, namely the prompt $P$, early $E$ and late $L$, to estimate the code phase error. Whereas the PLL part uses two different carrier replicas, denoted as in-phase $I$ and quadrature-phase $Q$, to estimate the phase error. Note that a common timing base is required for the two loops in order to work together in a stable way. Usually the LO is used for this task.

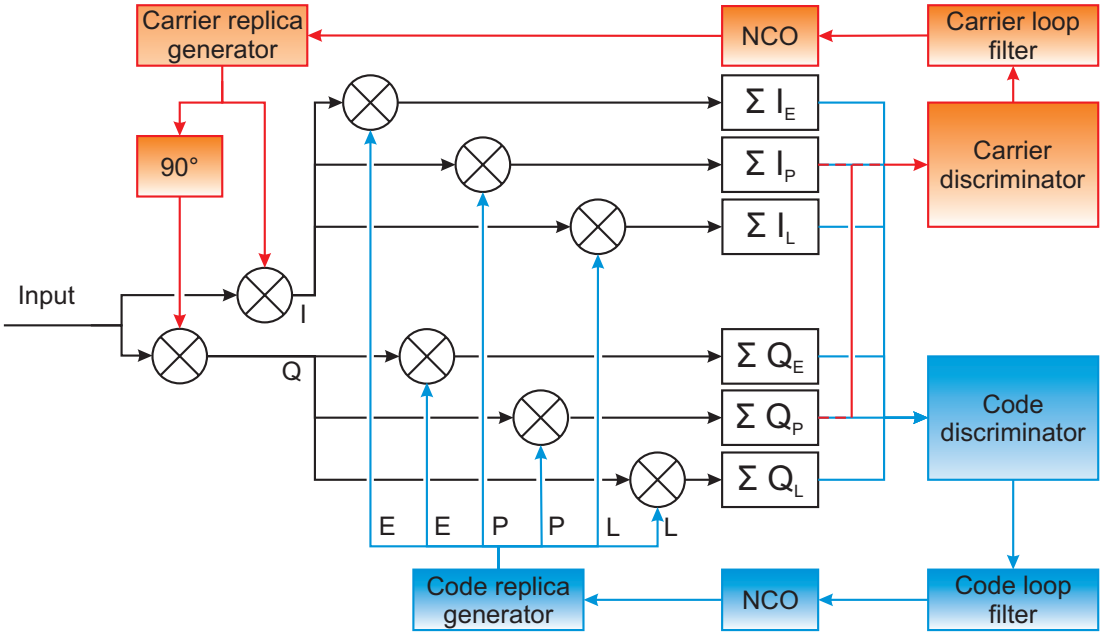



Figure 5.9: Implemented signal tracking module in case of BPSK signals

## 5.5 Tracking of binary offset carrier signals

In case of a BOC modulation the autocorrelation function has several peaks, as described in Sect. 4.3. The number of peaks depends on the order of the BOC modulation. The main peak is sharper, compared to the one of BPSK modulation and, thus, performs better in the presence of multipath (Macchi 2010). The conventional DLL discriminators for BPSK signals show multiple zero crossings, which can produce "ambiguous" or false lock positions under non-ideal conditions. False locks degrade the position performance significantly. Figure 5.10 shows the multiple zero crossings when tracking a BOC modulated signal using the previously mentioned DLL discriminators.

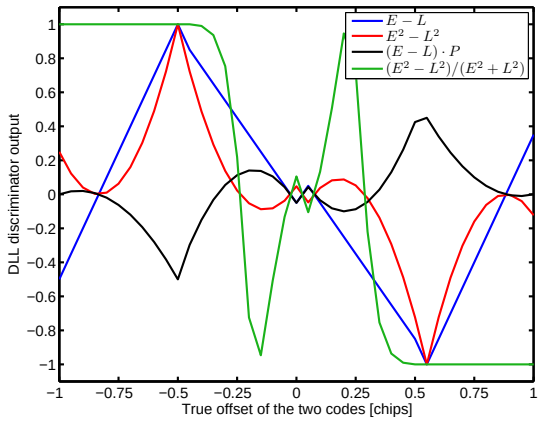



Figure 5.10: Comparison of different discriminator functions in case a BOC signal is tracked ($\delta = 0.5$)

Several code tracking techniques have been proposed in the literature with the task to keep track on the main peak which indicates the existence of the main signal. The proposed tracking algorithms can be divided into the following classes:

- BOC tracking using a single sideband (SSB);
- BOC tracking with the bump-jumping (BJ) algorithm;
- BOC tracking using the dual estimator (DE);
- BOC tracking with multiple-gate discriminators (MGD).

The SSB tracking method treats each BOC sideband as a separate PSK signal. Each sideband tracking creates an unambiguous PSK correlation peak which can be used by a standard PSK discriminator. This method provides a robust solution but degrades the tracking performance and is only suitable for a low precision receiver (Blunt 2007).
The bump-jumping algorithm compares the amplitude of the correlation peak which is currently tracked with the amplitude of the adjacent correlation peak to assure that the correct correlation peak is being tracked. Therefore two additional replica codes, denoted as very early (VE) and very late (VL), are needed (Gleason and Gebre-Egziabher 2009). The drawback of this method is the time it takes to reach the correct code phase after a slip in tracking. The dual estimator tracking method is implemented as a three loop configuration. The first tracks the code phase of the received signal, the second tracks the subcarrier phase and the third loop tracks the carrier frequency and/or phase (Gleason and Gebre-Egziabher 2009).
The multiple gate delay approach, proposed by Bello and Fante (2005), synthesizes an unambiguous discriminator function by combining multiple correlator channels. The multiple gate delay (MGD) technique is composed of $k$ early and $k$ late correlators. The early and late correlators are combined using different weighting coefficients $a_k$ in order to form the shape of the discriminator. The multiple gate delay approach was used within this implementation due to the easy possibility of the implementation of this method in software. The drawback of this method is that the computational load is increased significantly compared to the other methods. Following Blunt (2007) the correlations can be written as follows

$$\begin{aligned} I_{E_k} &= \sqrt{\frac{A}{2}}\, R\left(\tau - \hat{\tau} - \left(k - \frac{1}{2}\right)\delta\right)\cos(\Delta\Phi) \\ I_{L_k} &= \sqrt{\frac{A}{2}}\, R\left(\tau - \hat{\tau} - \left(k + \frac{1}{2}\right)\delta\right)\cos(\Delta\Phi) \\ Q_{E_k} &= \sqrt{\frac{A}{2}}\, R\left(\tau - \hat{\tau} - \left(k - \frac{1}{2}\right)\delta\right)\sin(\Delta\Phi) \\ Q_{L_k} &= \sqrt{\frac{A}{2}}\, R\left(\tau - \hat{\tau} - \left(k + \frac{1}{2}\right)\delta\right)\sin(\Delta\Phi). \end{aligned} \tag{5.17}$$

In (5.17) the variable $\delta$ denotes the early-to-late spacing of the standard DLL algorithm, and $k$ is an integer running from 1 to $N$. Following Gleason and Gebre-Egziabher (2009)

an unambiguous BOC discriminator can be obtained by using the normalized early late power discriminator (Table 5.2). The discriminator function can be formulated as

$$D_{\mathrm{MGD}(N)}(\tau) = \frac{\sum_{k=1}^{N} a_k \left[R^2\left(\tau - \hat{\tau} - \left(k - \frac{1}{2}\right)\delta\right) - R^2\left(\tau - \hat{\tau} - \left(k + \frac{1}{2}\right)\delta\right)\right]}{\sum_{k=1}^{N} a_k \left[R^2\left(\tau - \hat{\tau} - \left(k - \frac{1}{2}\right)\delta\right) + R^2\left(\tau - \hat{\tau} - \left(k + \frac{1}{2}\right)\delta\right)\right]}. \tag{5.18}$$

Depending on the number of correlators $N$ and the correlator spacing $\delta$ different weighting coefficients $a_k$ can be used. In the literature (Blunt 2007, Bello and Fante 2005, Skournetou and Lohan 2007, Hurskainen et al. 2008) different combinations of $N$ and $a_k$ are proposed. Two main classes of MGD discriminators are distinguished: smooth and bumpy. A smooth discriminator provides a monotonic function with a single shallow zero crossing, while a bumpy discriminator synthesizes a single steep zero crossing with many undulations across the discriminator characteristic (Gleason and Gebre-Egziabher 2009). Within the literature (Bello and Fante 2005, Hurskainen et al. 2008) several optimizations regarding the correlator spacing and the coefficients have been proposed in order to enhance the multipath resistance. Thereby, unequal correlator spacing is taken into account as well. Within this thesis three different MGD approaches, as shown in Table 5.3 have been investigated.

Table 5.3: Different multiple gate delay discriminator parameters

| Type | $N$ | Correlator spacing $\delta$ | Coefficients $a_k$ |
|---|---|---|---|
| Smooth | 4 | 0.525 | [1.000 1.250 1.500 1.750] |
| Bumpy | 4 | 0.200 | [1.000 1.125 1.250 1.375] |
| Unequal spacing | 3 | [0.72 0.44 0.30] | [0.25 0.27 0.51] |

More information on different MGD implementations can be found in Skournetou and Lohan (2007). Figure 5.11 shows a comparison of the MGD discriminator functions mentioned in Table 5.3. The smooth discriminator shows a shallow zero crossing, while the bumpy one has a steeper one. The unequal spacing MGD shows a larger linear region compared to the other two.

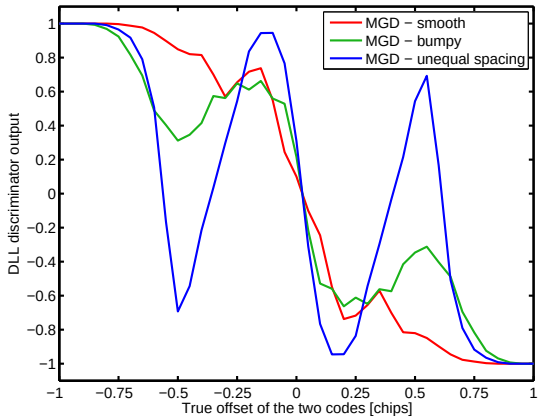



Figure 5.11: Comparison of the different MGD discriminator functions

Following Macchi (2010) the MGD approach can be used for different BOC modulation types, including composite binary offset carrier (CBOC). Within this thesis only a BOC(1,1) modulation is considered, but tracking a CBOC modulated signal would enhance the resistance to multipath. Within Sect. 5.6 a comparison of the different MGD discriminators is provided. The pilot channel tracking loop follows a similar structure to that of the general tracking architecture shown before, but an additional step is required to remove the secondary codes. For the sake of simplicity the secondary code was omitted within this thesis. More information on secondary code tracking can be found in Mongrédien (2008). For BOC tracking the overall tracking loop architecture is shown in Figure 5.12. It must be mentioned that for a better understanding only one additional correlator pair is visualized. The following section shows selected investigations and results of the implemented tracking module.

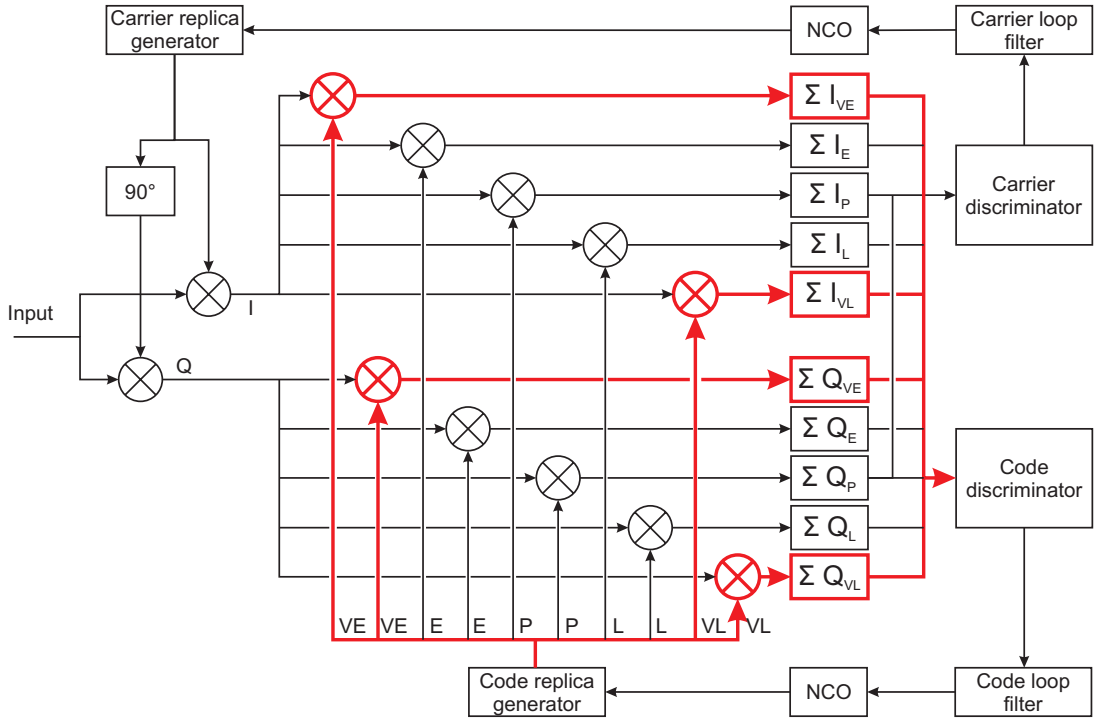
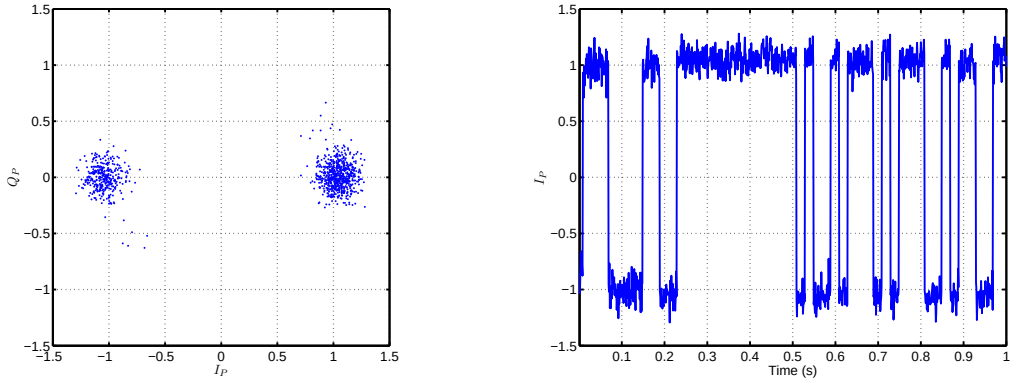



Figure 5.12: Implemented MGD tracking module in case of tracking BOC signals

## 5.6 Tracking results

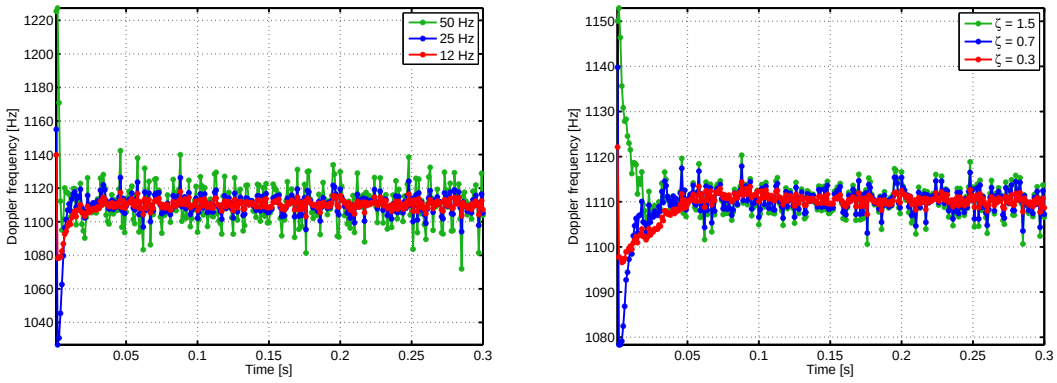
The tracking procedure described above is repeated for all visible satellite PRNs in the receiver channels. The tracking module was tested and verified using different IF data sets. For verification purposes the GIPSIE®–IFS as well as recorded real-world data have been used. More information on the simulated scenarios and recorded data sets are provided in Chap. 8.
Based on the acquisition results of the simulated GPS/Galileo scenario (Sect. 4.6), the tracking module was tested. Each IF data set contains 60 seconds of signal. Processing 60 s of IF data represents a good compromise between a reasonable computation time (Sect. 8.4) and an adequate amount of results for investigation. All data sets were processed over the full time, but the plots in this section only show the first second for a better visibility. The tracking was performed for both GPS L1 C/A and Galileo E1B. In case of the Galileo E1B signal different discriminator types have been tested. Figure 5.13a shows a constellation diagram of the first second of a GPS L1 C/A signal. The diagram represents the phase error and the noise between the phase of the incoming signal and the phase of the local replica wave. The two spots - one on the left side and one on the right side of the Q-axis - indicate the 180° phase shifts due to the data bit reversals. The averaged position of the phasor remains aligned with the I-axis, thus the local replica phase is aligned with the phase of the incoming signal. The blunders (Figure 5.13a) occur at the initial tracking phase representing the settling time of the loop. This effect is also visible in Figure 5.13b




(a) (b)

Figure 5.13: Normalized I and Q constellation diagram (a) and in-phase prompt correlator output (b) for a GPS L1 C/A signal generated by GIPSIE®–IFS

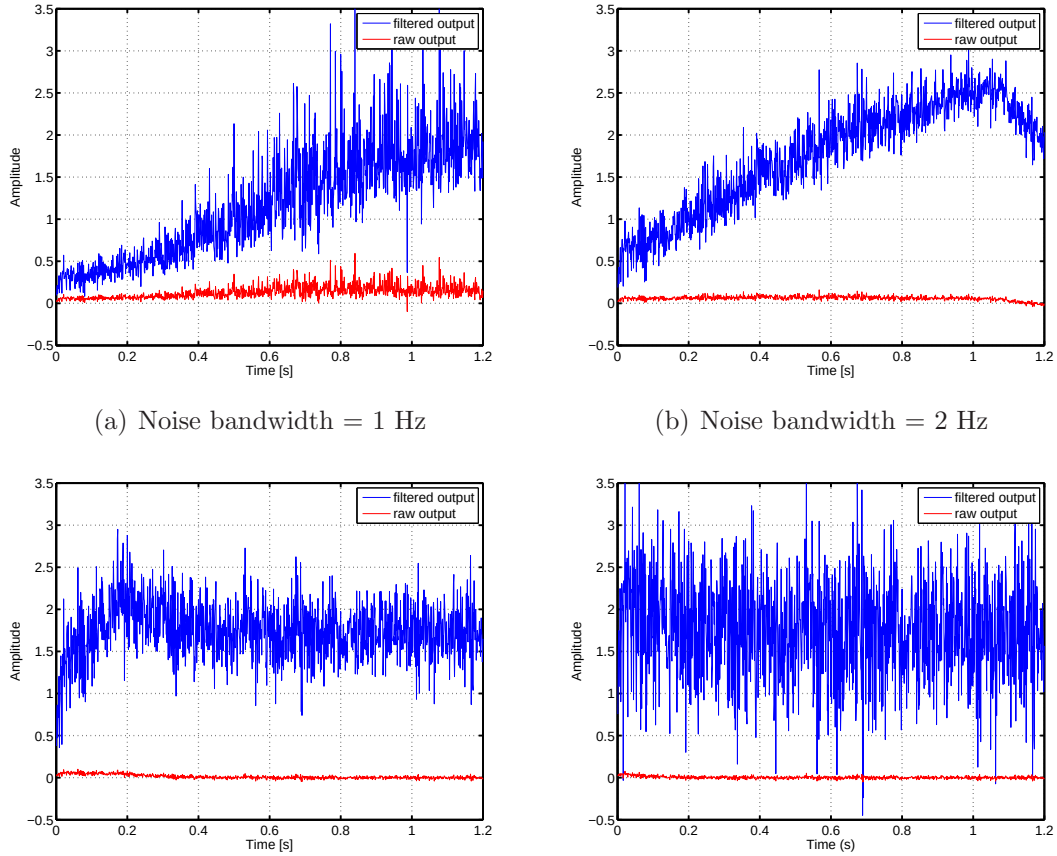
at the very beginning of the time series. Figure 5.13b shows the first second of the in-phase prompt correlator output when tracking a GPS L1 C/A signal. The jumps correspond to the change in the sign of the data bit. In this case the positive and negative values already indicate the navigation data bits. Note that a preprocessing, described in Chap. 6, has to be done before the navigation data can be used. As discussed in Sect. 5.2 the noise bandwidth and the damping ratio control the behavior of the loop. Figure 5.14a shows the estimated Doppler frequency for a GPS L1 C/A signal in case of using a damping ratio $\zeta = 0.7$ and different noise bandwidths (12 Hz, 25 Hz, and 50 Hz).




(a) (b)

Figure 5.14: Comparison of the PLL discriminator output for different noise bandwidths (a) and damping ratios (b) using a GPS L1 C/A signal generated by GIPSIE®–IFS

In all cases the estimated Doppler frequency for, e.g., GPS PRN 4, matches the simulated one (Table 4.3). But it is visible that a larger noise bandwidth results in a larger settling time, while a smaller noise bandwidth might lead to a wrong discriminator output.

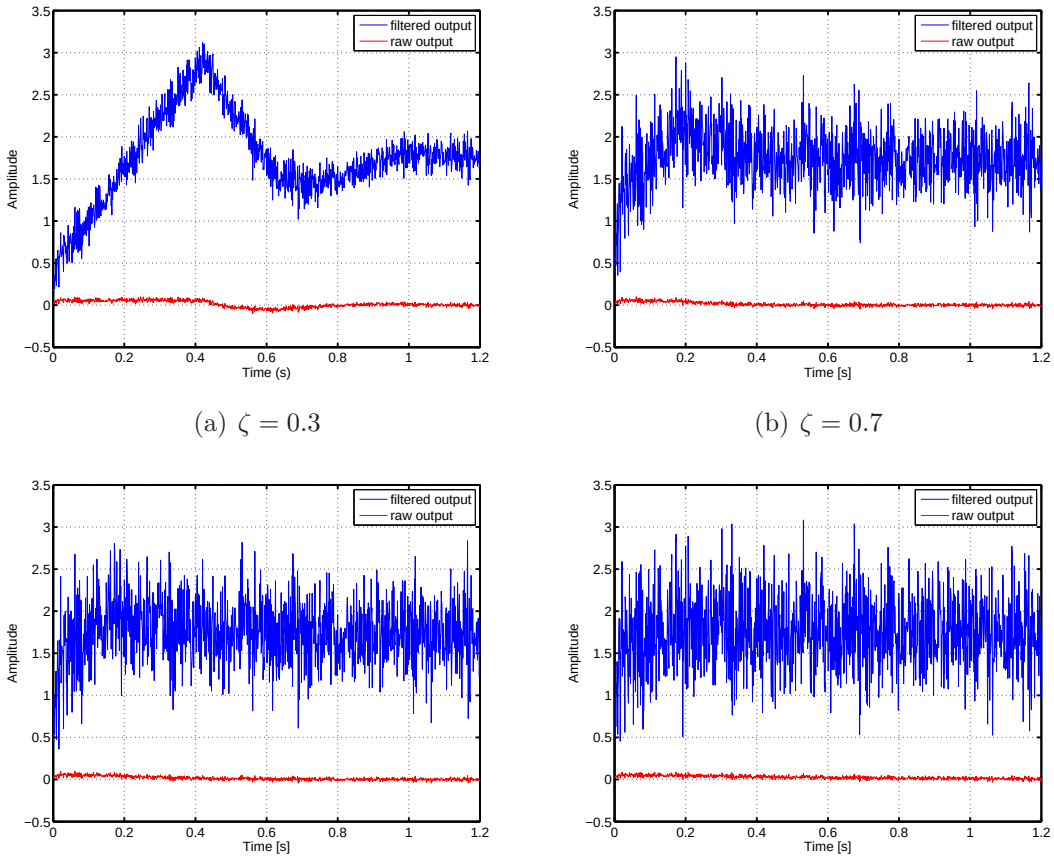



(a) Noise bandwidth = 1 Hz (b) Noise bandwidth = 2 Hz

(c) Noise bandwidth = 4 Hz (d) Noise bandwidth = 8 Hz

Figure 5.15: Comparison of the filtered and raw DLL discriminator output in case of using different noise bandwidths when processing a GPS L1 C/A signal

The Doppler frequency estimations of the PLL using a noise bandwidth of 12 Hz and different damping ratios ($\zeta = 0.3$, 0.7, and 1.5) for GPS PRN 4 are visualized in Figure 5.14b. In case of using a small damping ratio a larger overshoot occurs. A higher damping ratio enables a fast settling but can create false locks. The same behavior can be observed within the DLL.
Figure 5.15 shows the amplitude of the filtered and raw GPS DLL discriminator output when using different noise bandwidths (1 Hz, 2 Hz, 4 Hz, and 8 Hz) and a damping ratio of $\zeta = 0.7$. The red curve shows the raw output of the discriminator, while the blue curve shows the filtered input for the NCO. The amount of noise in the filter is dependent on the noise bandwidth. It also controls the settling time. The filtered output settles after $\approx 0.3$ s using a small noise bandwidth of 4 Hz (Figure 5.15c). When using a smaller noise bandwidth (e.g., 2 Hz) the noise within the filtered output is reduced but the settling time is increased. If the noise bandwidth is chosen too small (i.e., 1 Hz), as shown in Figure 5.15a, then the loop is not able to achieve a lock. A higher noise bandwidth of e.g., 8 Hz, as shown in Figure 5.15d, leads to a shorter settling time but allows more noise in the filtered discriminator output. As shown in Figure 5.16, the damping ratio influences the settling time and the amount of overshoot.




(a) $\zeta = 0.3$

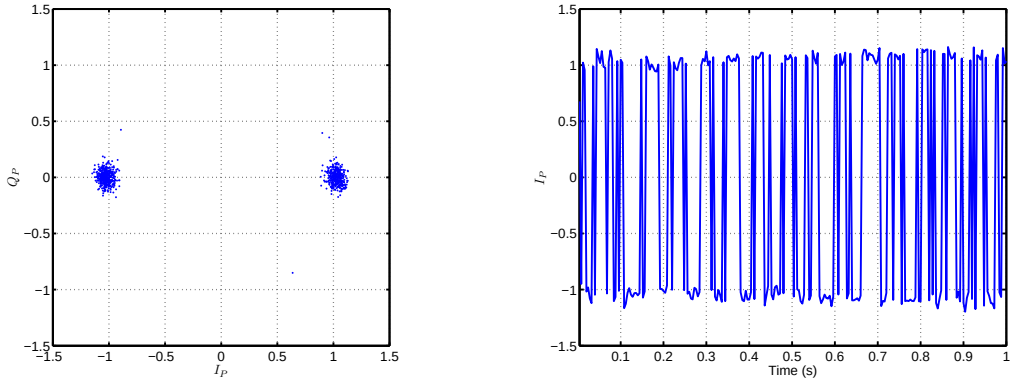
(b) $\zeta = 0.7$

(c) $\zeta = 0.9$

(d) $\zeta = 1.5$

Figure 5.16: Comparison of the filtered and raw DLL discriminator output in case of using different damping ratios when processing a GPS L1 C/A signal

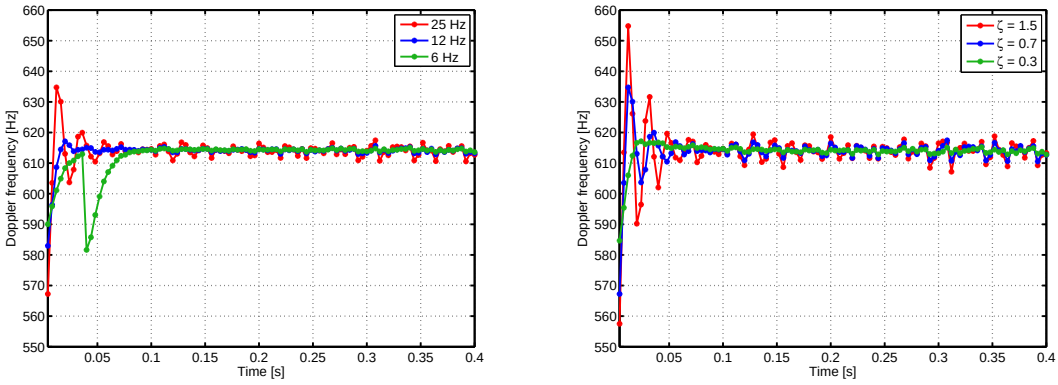
An underdamped system ($0 < \zeta < 1$) shows an overshoot and a swing before settling down. In contrast, if the system is overdamped ($\zeta > 1$) no swing occurs but the noise is increased. By comparing the Figures 5.16a and 5.16d the differences of an underdamped and an overdamped system are visible. A damping ratio of $\zeta = 0.7$ (Figure 5.16b) shows a good compromise between settling time and oscillation. Thus, for processing all GPS L1 C/A signals a noise bandwidth of 4 Hz and a damping ratio of 0.7 have been used.
In a next step the tracking of Galileo E1B signals is evaluated. The normalized I and Q constellation diagram of the first second of tracking is depicted in Figure 5.17a. Compared to the constellation diagram of the GPS signal, the magnitude of the I and Q values are larger. This is due to the longer integration interval of 4 ms in case of Galileo E1B, compared to 1 ms in case of GPS L1 C/A. The 'outliers' occur due to the initial tracking phase, when the loop is refining the coarse values. Figure 5.17b shows the in-phase prompt correlator output, representing the demodulated navigation message symbols. Compared to the corresponding GPS L1 C/A correlator output, the higher symbol rate of the Galileo E1B navigation message is noticeable. Note that every 4 ms a change in the navigation symbol can occur. Again a preprocessing is necessary before the navigation message parameters can be used.




(a) (b)

Figure 5.17: Normalized I and Q constellation diagram (a) and in-phase prompt correlator output (b) for a Galileo E1B signal generated by GIPSIE®

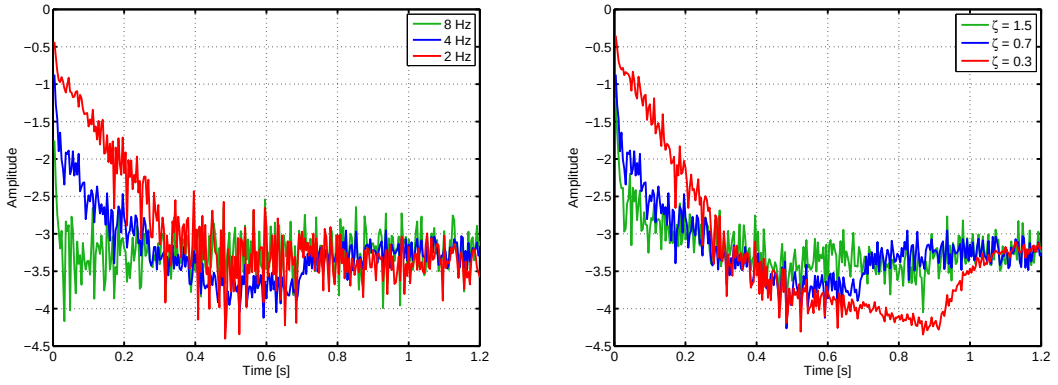
Figures 5.18a and 5.18b show the estimated Doppler frequency of the PLL when using different noise bandwidths and damping ratios in case of tracking Galileo PRN 72. For a better visualization only the first 0.4 s are shown.
As discussed previously a larger noise bandwidth results in a larger overshoot, as in the case of using a noise bandwidth of 25 Hz. In all cases the PLL estimates the correct Doppler frequency (Table 4.4). In case of using a smaller noise bandwidth of 6 Hz the PLL shows a wrong Doppler frequency estimate within the first milliseconds of tracking. When using a damping ratio of $\zeta = 1.5$ a large overshoot in the magnitude of 40 Hz is visible. In this case it takes much longer for the PLL to settle onto the correct Doppler frequency value. The settling time and the overshoot is reduced in case of using a smaller damping ratio (e.g., $\zeta = 0.3$). A damping ratio of 0.7 represents a good compromise between overshoot and damping ratio. As visualized, the overshoot and stability depends on the noise bandwidth




(a) (b)

Figure 5.18: Comparison of the PLL discriminator output for different noise bandwidth (a) and damping ratios (b) using a Galileo E1B signal generated by GIPSIE®

and damping ratio. Figure 5.19a shows the amplitude of the filtered DLL discriminator output in case of tracking Galileo PRN 72 and using different noise bandwidths. A higher noise bandwidth (e.g., 8 Hz) increases the noise of the filtered output but enables a faster settling. When using a lower noise bandwidth of 2 Hz the noise is reduced but the settling time is increased. The filtered DLL discriminator outputs for different damping ratios are presented in Figure 5.19b. Also in this case a higher damping ratio results in a faster settling time but with increased noise. While a lower damping ratio (e.g., $\zeta = 0.3$) produces a larger overshoot.




(a) (b)

Figure 5.19: Comparison of the DLL discriminator output for different noise bandwidth (a) and damping ratios (b) using a Galileo E1B signal generated by GIPSIE®

Thus, a compromise between settling time and the amount of noise must be made. Within this implementation a damping ratio of $\zeta = 0.7$ and a noise bandwidth of 4 Hz are used for the tracking of Galileo E1B signals. As discussed in Sect. 5.5, tracking a BOC signal requires a more sophisticated DLL. For tracking BOC modulated signals the MGD approach has been used. The different MGD correlator types (i.e., smooth, bumpy, and

unequal-spacing) are compared in Figure 5.20. The red line in each plot shows the prompt correlator output, while the other colors represent the early and late correlators according to the values given in Table 5.3.

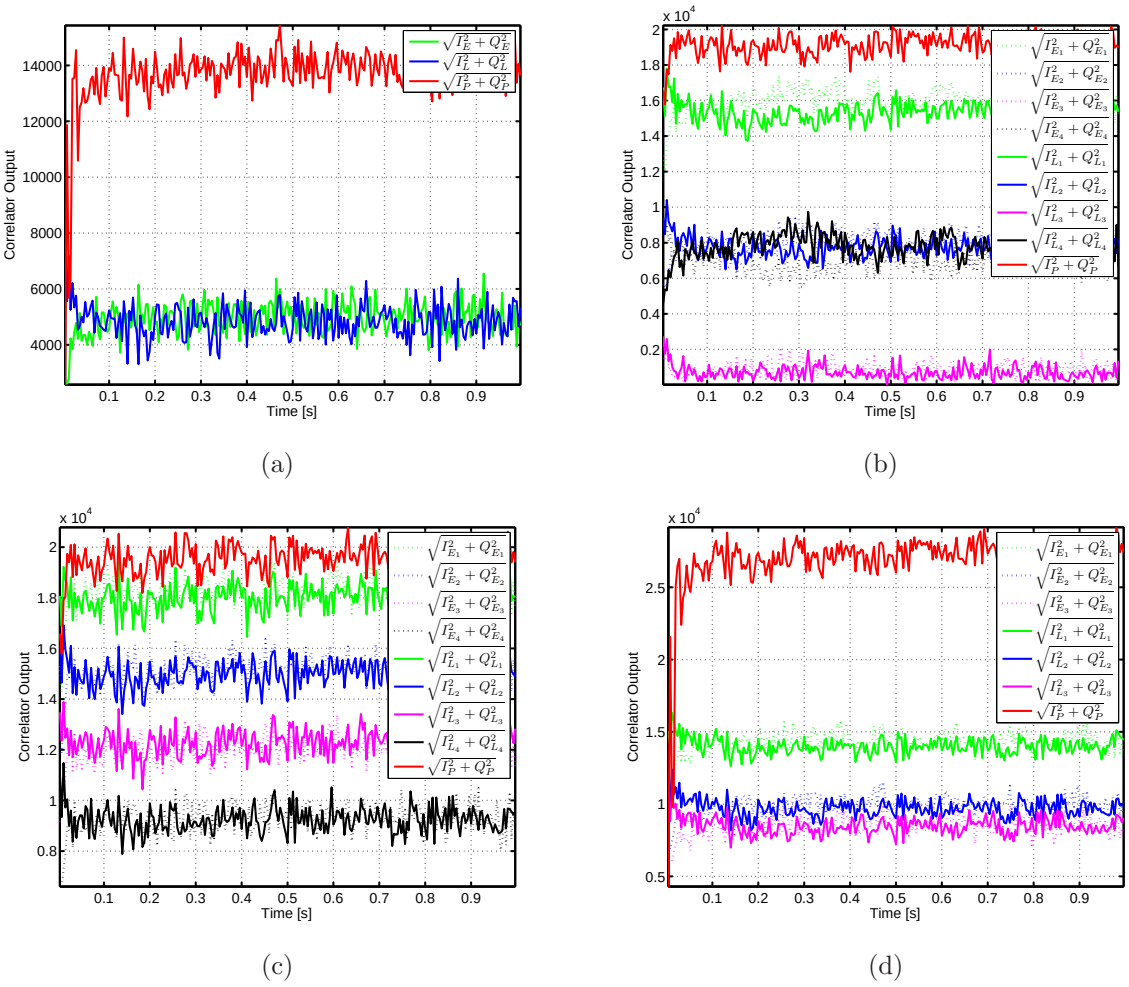



Figure 5.20: Comparison of different multiple gate delay discriminators: (a) standard discriminator, (b) smooth, (c) bumpy, and (d) unequal-spacing when tracking a Galileo E1B signal generated by GIPSIE®–IFS

According to theory the prompt correlation output must have the highest magnitude. If another correlator shows a higher correlation value a false lock has occurred. This case is visualized in Figure 5.21. The standard discriminator, as used in case of GPS L1 C/A signal tracking, shows a higher correlation value for the late correlator. Although starting with the same magnitude with the prompt correlator the DLL locks onto the wrong peak (side peak of the BOC ACF). This would cause a significant error within the pseudorange measurements.

For tracking BOC modulated signals, the unequal-spacing discriminator function has been used. The unequal-spacing discriminator function has the advantage that only 6 replicas instead of 8 are needed and, thus, the processing time is reduced. The evaluation of the simulated IF data proofs that the tracking module is able to process BPSK as well as BOC

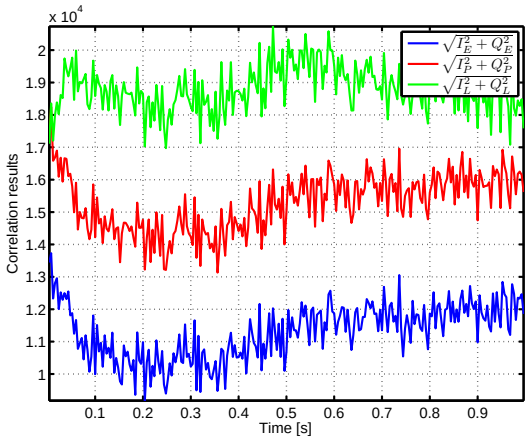
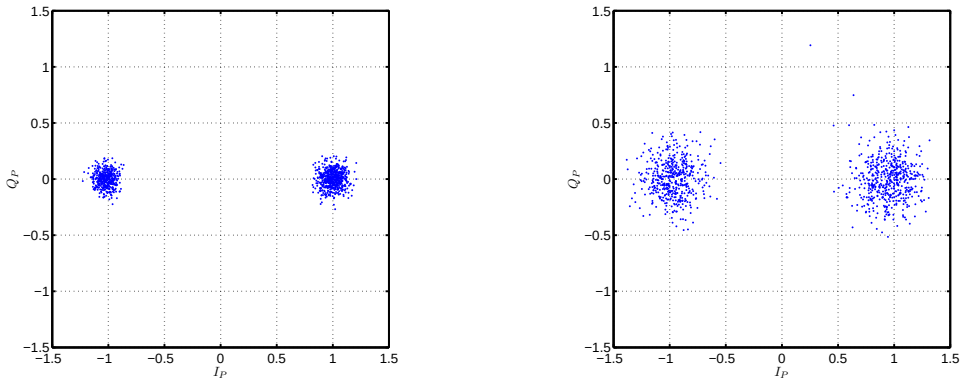



Figure 5.21: Wrong tracking result in case of using a standard DLL discriminator function using a Galileo E1B signal generated by GIPSIE®

modulated signals. The tracking of BOC modulated signals requires a special treatment to avoid the tracking of side peaks.
In a next step real-world data are used to test the tracking module in case of GPS L1 C/A and GPS L2C signals using the Fraunhofer multi-band front-end. The tracking of GPS L1 C/A and GPS L2C is performed independently as described in the previous section. For tracking L2C signals the L2CM-code has been used (Sect. 4.1.1). Figure 5.22 shows a comparison of the normalized constellation diagram in case of L1 C/A and L2C signal tracking. In case of tracking the GPS L1 C/A signal the constellation diagram shows the same behavior as in the case of the simulated data. The L2C constellation diagram reveals that the correlation values show a larger noise level compared to the L1 tracking. This is due to the fact, that the L2CM-code is used for the tracking only. The zeros within the PRN code replica cause a degradation of the correlation result.

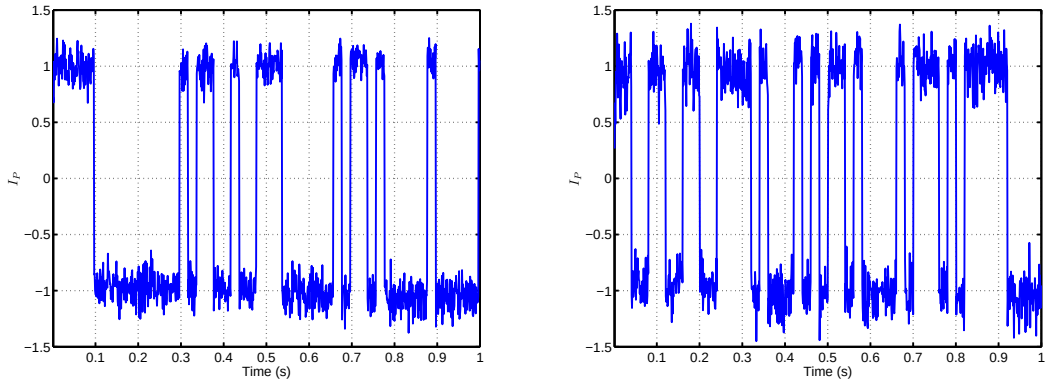



(a) (b)

Figure 5.22: Normalized I and Q constellation diagram of (a) GPS L1 C/A and (b) GPS L2C signal

Nevertheless in both cases the averaged position of the phasor remains aligned with the

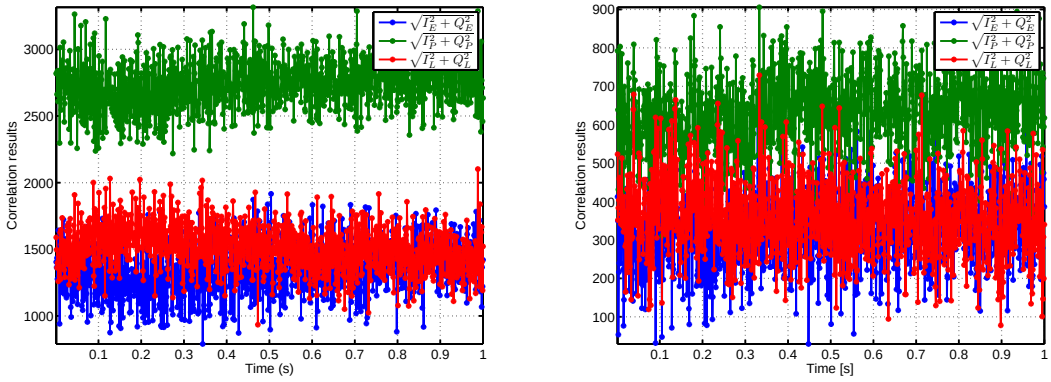
I-axis, thus the local replica phase is aligned with the phase of the incoming signal. Figure 5.23 shows a comparison of the first second of the L1 C/A and the L2C in-phase prompt correlation output.




(a) (b)

Figure 5.23: In-phase prompt correlator output (a) GPS L1 C/A and (b) GPS L2C signal

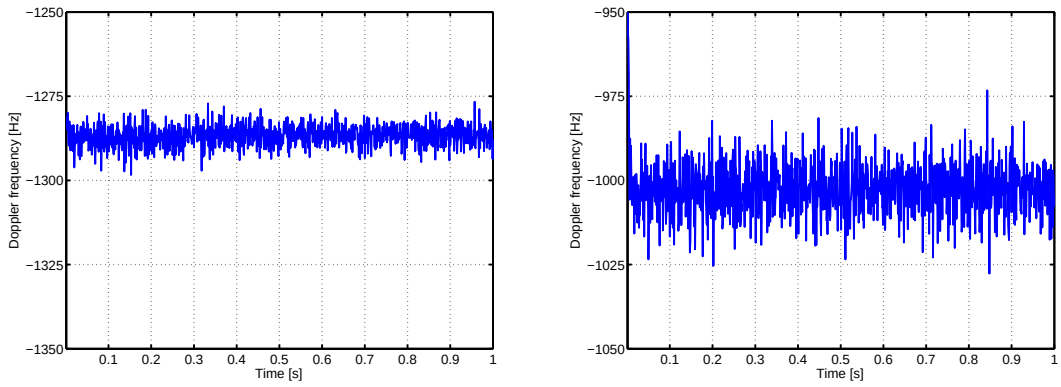
In both plots the 180° phase shifts, due to the navigation data bits, are clearly visible. Figure 5.24a shows the first second of the early, prompt, and late correlation results in case of tracking a GPS L1 C/A signal. The prompt correlation values always show the highest amplitude. The same can be observed when looking at the early, prompt, and late correlation results in case of tracking a GPS L2C signal (see Figure 5.24b). The difference between the two results is that the noise in case of L2C tracking is higher.




(a) (b)

Figure 5.24: Early, prompt, and late correlation results in case of tracking (a) GPS L1 C/A and (b) GPS L2C signals

Figure 5.25 shows a comparison of the estimated Doppler frequency for GPS PRN 12 in case of tracking one second of L1 C/A and L2C signals. The estimated L2C Doppler frequency shows a higher noise compared to the L1 frequency.




(a) (b)

Figure 5.25: PLL discriminator output for (a) GPS L1 C/A and (b) GPS L2C signals

When comparing the Doppler frequency estimates of L1 and L2 the expression of (4.23) can be used. In the ideal case the differences between the corresponding Doppler values, $\Delta D_{f_{L1}f_{L2}}$, must be equal zero. As shown in Figure 5.26, the $\Delta D_{f_{L1}f_{L2}}$ values have a mean almost zero and a standard deviation of 10 Hz can be observed.

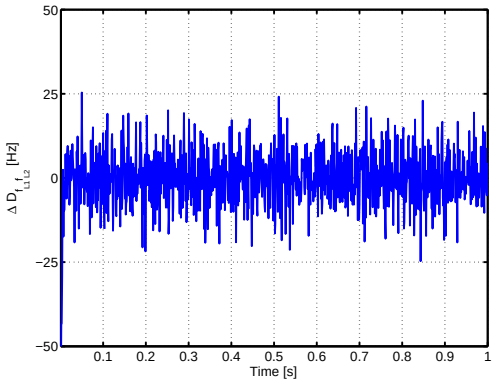



Figure 5.26: Visualization of the delta Doppler values $\Delta D_{f_{L1}f_{L2}}$

Both the simulated scenarios as well as the real-world recordings proofed that the implemented tracking module is able to process GPS L1 C/A and L2C signals, as well as Galileo E1B signals. The impact of different noise bandwidths and damping ratios was investigated. The MGD approach for tracking BOC modulated signal was tested and verified. Note that the tracking module operates in post-processing mode and is currently able to handle one channel at a time but can be expanded for multiple channels for future applications. In a next step the output of the tracking loops has to be pre-processed in order to obtain the navigation message data and to compute the pseudorange measurements.

# Chapter 6

# Data preprocessing

Within this chapter the necessary steps for obtaining the raw measurements (i.e., pseudoranges) and navigation data out of the tracking loop outputs are discussed. The decoding procedures for different navigation message formats are shortly highlighted. The pseudorange computation algorithms, as well as the navigation message decoding procedures are validated by comparing the computed values with the simulated ones.

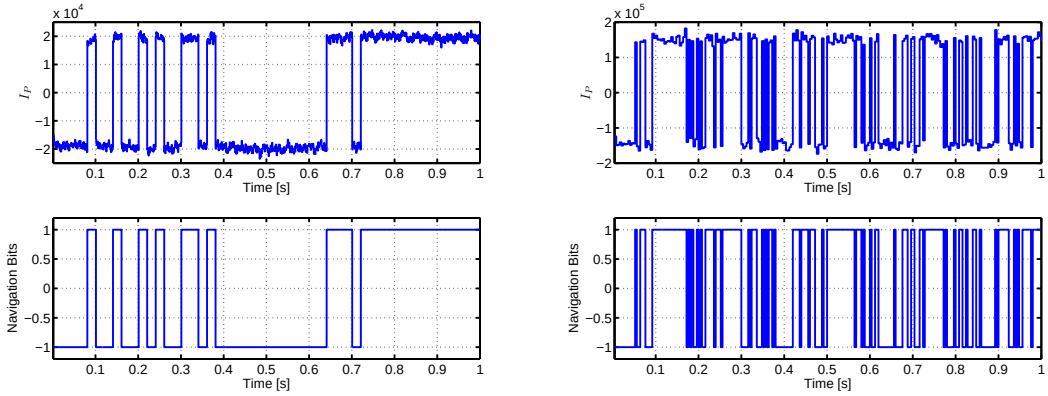
## 6.1 Introduction

A receiver position estimation is only possible if both, pseudorange measurements and ephemerides data, for at least four tracked satellites are available. Since the output of the tracking loops are code and carrier phases it is necessary to convert them into pseudorange, Doppler, or phase measurements which can be used within the receiver position computation (Kaplan 2006). Therefore several steps are necessary in order to obtain the raw measurements. Additionally the navigation messages have to be decoded to obtain ephemerides and almanac data as well as time information. As described in Chap. 2 different systems and services use different data messages. Common to all is that at least one subframe or page exists which transmits ephemerides and time information (Hofmann-Wellenhof et al. 2008). This enables an autonomous computation of the satellite position information which in turn is needed for the receiver position computation. The decoding of the navigation data involves in general three operations:

- Symbol synchronization;
- Frame synchronization;
- Navigation data recovery.

As described in Sect. 5.3 the PLL removes the code and carrier components from the incoming signal. The remaining phase changes between 0 and 180 degrees correspond to the data symbol transitions. Thus, the data symbols can be retrieved by analyzing the changes in sign of the in-phase prompt correlation values (e.g., $\mathrm{sgn}(I_P)$). The task of the symbol synchronization is to find the symbol or bit boundaries within the correlation values. In case of GPS L1 C/A, using a coherent integration time of 1 ms, a demodulated navigation bit has a length of 20 consecutive $I_P$ values. Thus, the first bit transition has to be located. Since the length of a bit is known it is possible to find all the other bit transitions. The left plot of Figure 6.1 shows the bit detection in case of GPS L1 C/A NAV bit stream. In this case 20 correlation values have to be replaced by only one bit. In

case of Galileo E1B, as shown in Figure 6.1b, due to 4 ms coherent integration time, each $I_P$ value symbolizes already one Galileo navigation symbol. The five times higher data rate of the Galileo I/NAV message, compared to the GPS NAV message, is clearly visible.




(a) (b)

Figure 6.1: Symbol detection in case of a GPS L1 C/A NAV bit stream (a) and a Galileo E1B I/NAV symbol stream (b)

Since the navigation message structures are based on frames (subframes or pages) the boundaries of those frames have to be located in a next step. The frame synchronization can be achieved by correlating the incoming stream of bit values with a well-known data bit sequence called preamble or synchronization pattern. This unique sequence is transmitted at the beginning of each subframe or page. The binary pattern in case of the Galileo I/NAV message is given by 0101100000 and for the Galileo F/NAV 101101110000, respectively. The preamble bits in case of GPS are defined as 10001011 for both the NAV and C/NAV messages. After having found the first frame boundary it has to be verified that another boundary occurs exactly after one whole message length. The knowledge of the frame boundaries is important for two reasons. First for the decoding of the navigation data which is done at subframe or page level and for obtaining the time of transmission, which is transmitted within the navigation message and is related to the 'leading edge' of the subframe or page.
An important task when decoding the navigation data is the detection and correction of navigation data errors. Three different techniques, dealing with bit/symbol errors, are used within GNSS:

- Cyclic redundancy check (CRC);
- Forward error correction (FEC);
- Interleaving.

The CRC is used to detect bit errors, whereas the FEC can be used to detect and correct the errors. For mitigating the loss or corruption of a series of bits block interleaving is used within the Galileo navigation message structure. Note that not all navigation messages are protected by FEC or block interleaving (Chap. 2).

## 6.2 Forward error correction coding and Viterbi decoding algorithm

In a typical digital communications system, a stream of bits (information) is transferred from one point to another through a communication channel and is, therefore, susceptible to bit errors or noise. Forward error correction techniques allow to better exploit the channel capacity by carefully adding redundant information to the data being transmitted through the channel. In FEC systems, the transmitted data are encoded in such a way that the receiver can detect as well as correct errors. Therefore, the convolutional encoder inserts redundant information bits into the data stream. Thus, it is possible for the decoder to reduce and correct errors which are introduced during signal transmission. The transmitter sends a data sequence comprising $k$ bits of data to an encoder. The encoder uses a convolutional coding scheme. As mentioned above the GPS C/NAV message as well as the Galileo navigation messages are protected against transmission errors by using a forward error correction coding based on a convolutional code. The navigation bit stream in all cases is FEC encoded by a rate $1/2$ convolution code. The convolutional encoding has a constraint length of 7 according to the specifications. The used coding scheme for GPS and Galileo is depicted in Figure 6.2.

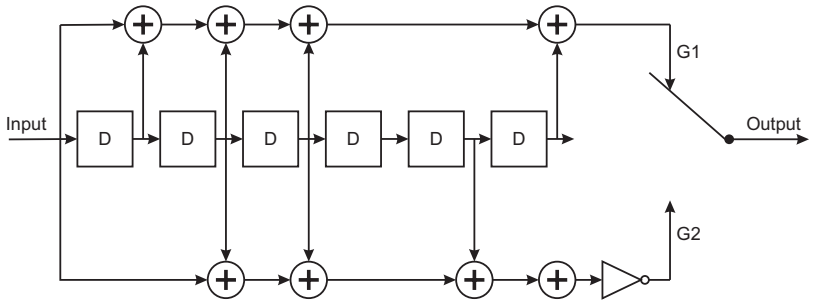



Figure 6.2: Convolutional coding scheme

The encoder takes the input data bits and generates an output symbol stream. The output symbol stream has twice the length of the input bit stream and is computed based on the given binary logic. Note that the encoded bits are referred to as symbols to avoid misunderstanding. Using this scheme makes the decoding slightly more complex compared to the GPS navigation message (NAV). For the decoding of the data stream a Viterbi decoder using the Viterbi algorithm is commonly used. The Viterbi algorithm estimates, based on a given set of observations, the most likely sequence of data. This is achieved by dynamic programming, which is an optimal recursive search process (Ziedan 2006). The Viterbi decoder takes the input symbols and performs the decoding. The main purpose of the decoder is to select the sequence (code word) with the minimum distance between the received symbol and the code word (Lou 1995). The Viterbi algorithm can use either hard or soft decision techniques. The hard decisions method is based on the Hamming distance, while the soft decision technique uses the Euclidean distance as a metric. The advantage of the hard decision method is, that it is simple to implement and does not involve probabilities. The implementation in this thesis uses a soft decision technique

to decode the symbols (Berglez 2010). Since FEC and Viterbi decoding are commonly used within communications, refer to Viterbi (1995), Ziedan (2006), Ipatov (2005), or Lou (1995) for further information.

## 6.3 Navigation data extraction

After the frame synchronization and, if required, de-interleaving and Viterbi decoding, the cyclic redundancy check has to be validated before the data recovery can start. The CRC protects the data bits against bit errors and is used within all navigation message formats. The CRC is computed using a generator polynomial $G(x)$. The bit field, which is protected by the CRC, is therefore written as a polynomial. This polynomial is then divided by the generator polynomial using a modulo-2 arithmetic. The resulting remainder represents the CRC value. This value is then compared with the transmitted one. If the computed and the received values match, then the CRC is passed. The number of CRC bits depends on the generator polynomial which is defined in the corresponding interface control documents (Department of Defense 2011, European Commission 2010). More information on CRC can be found in Kaplan (2006).

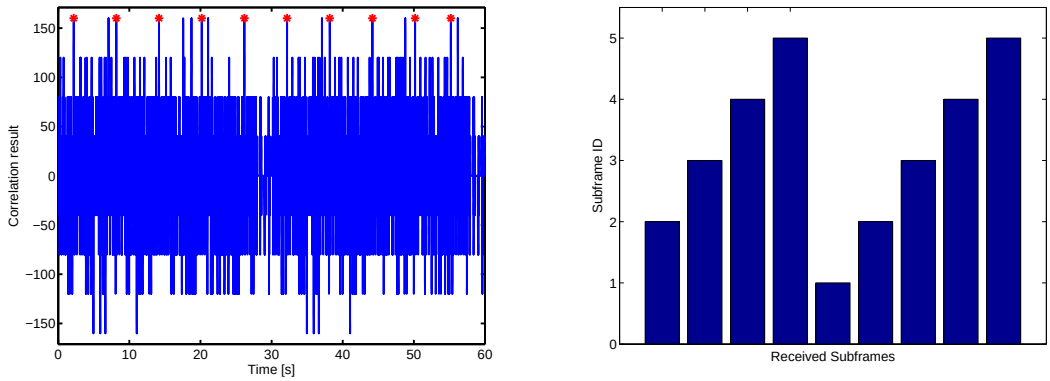
### 6.3.1 GPS navigation message

To retrieve the ephemerides and other data within the GPS NAV or C/NAV symbol stream the following tasks have to be accomplished:

- In case of C/NAV message decode the symbols into data bits using the Viterbi decoding algorithm;
- Detect and verify the preamble;
- Verify the CRC;
- Extract the information from the data bits.

Beside the decoding of the ephemerides data also the extraction of the time of signal transmission – time when the received subframe was transmitted from the satellite – is important. Therefore each subframe of the NAV and C/NAV message contains the so-called Z-count. The Z-count represents a truncated version of GPS time of week at the time of signal transmission. The Z-count is increased in 6 s (NAV) respectively 12 s (C/NAV) steps corresponding to the time between two consecutive subframes. The Z-count is referred to the leading edge of the following subframe. The Z-count is extracted from the navigation message, converted into the time of week (TOW), and stored within the receiver. A detailed description of the GPS navigation data extraction and the timing parameters is given in Department of Defense (2011).
Figure 6.3a shows the subframe synchronization and the occurrence of the preamble bits within the GPS L1 NAV message stream. As visible the preambles occur in regular intervals (300 bits). The obtained subframe ID sequence is shown in Figure 6.3b. The sequence shows two frames of the navigation message. According to the specification each frame

consists of five subframes which are transmitted in the order 1 to 5. Thereby, the content of the subframes 4 and 5 changes in each frame. The CRC was passed in this case as well. The decoded ephemerides values are stored in an internal format for further processing.




(a) (b)

Figure 6.3: Subframe synchronization showing the occurrence of the preamble bits (a) and received subframe numbers (b) in case of GPS L1 NAV message

In case of the C/NAV message Figure 6.4 shows the occurrence (red dots) of the preambles which are transmitted in regular intervals (every 12 s).

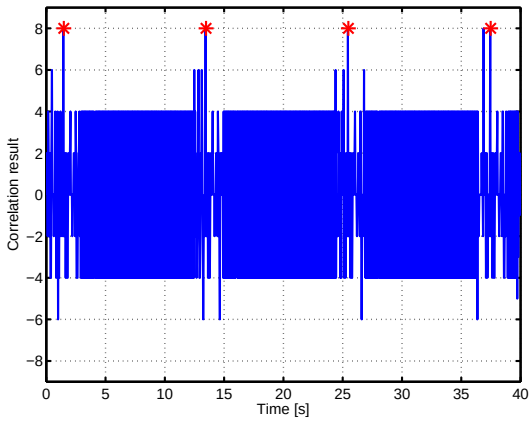



Figure 6.4: Occurrence of preambles within the C/NAV data bit stream

All received C/NAV messages contain message type 0, which corresponds to the dummy message type specified in Department of Defense (2011). Figure 6.5 shows the dummy message subframe. While the preamble, the TOW, and the CRC bits contain valid data, the data field itself is composed of alternating 1's and 0's. Also the message type field is zero. Thus, no navigation data could be used from the C/NAV message.

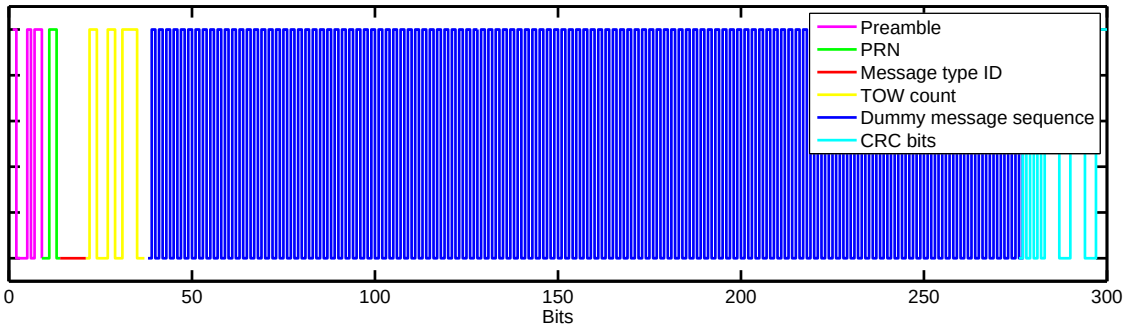



Figure 6.5: Content of received C/NAV message type 0

### 6.3.2 Galileo navigation message

The procedure of obtaining the Galileo navigation data within the I/NAV or F/NAV symbol stream differs from the GPS procedure. According to the Galileo signal in space interface control document (European Commission 2010) the following tasks have to be accomplished:

- Detect, verify, and remove the binary synchronization pattern;
- Deinterleave the page symbols;
- Decode the symbols into data bits using the Viterbi decoding algorithm;
- Verify the CRC;
- Extract information from the page bits.

According to the specifications, after the page synchronization the symbol stream has to be deinterleaved. If too many bits in a continuous block are corrupted (called burst errors) then the FEC algorithm is not able to restore the corrupted bits. Therefore Galileo uses block interleaving to mix the symbols of the navigation message. At the receiver the symbols are then descrambled again. Block interleaving is usually implemented by use of a $n \times k$ matrix. At the message generation the symbol stream is filled into a matrix column by column. The stream is then transmitted by reading the matrix row by row. At the receiver side the incoming symbols are deinterleaved, as shown in Figure 6.6, by performing the inverse operation.

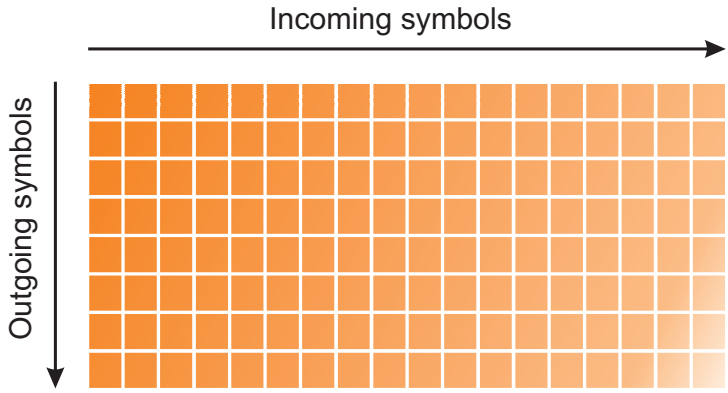


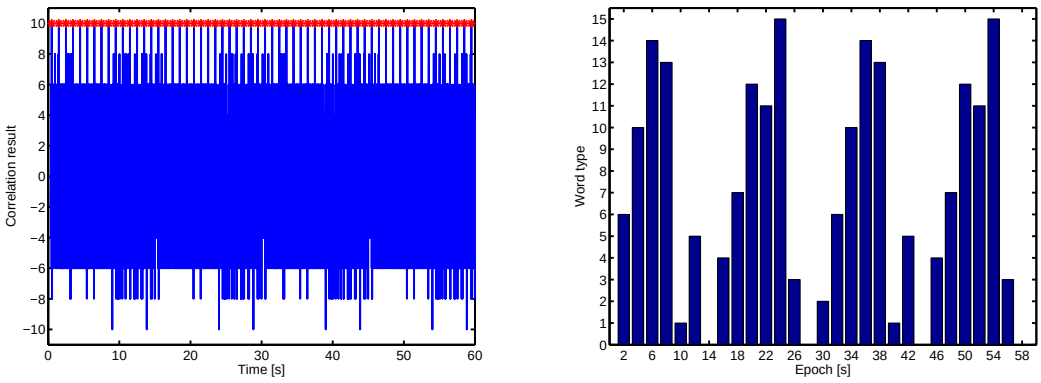

Figure 6.6: Deinterleaving scheme

Thus, the received symbols are fed into the $n \times k$ matrix row by row. The deinterleaved symbols are obtained be reading the matrix column wise. For the Galileo I/NAV and F/NAV messages the deinterleaving parameters are listed in Table 6.1.

Table 6.1: Deinterleaving parameters for the Galileo F/NAV and I/NAV message (European Commission 2010)

| Parameter | Message Type | |
|---|---|---|
| | F/NAV | I/NAV |
| Block deinterleaver size (symbols) | 488 | 240 |
| Block deinterleaver dimension ($n$ columns $\times$ $k$ rows) | $8 \times 61$ | $8 \times 30$ |

If a consecutive block of symbols is corrupted during transmission, then after deinterleaving the corrupted symbols are scattered and, thus, the FEC is able to correct these symbols. The next task within the navigation data decoding process is to convert the symbols into bits. This is achieved by feeding the symbols into the Viterbi decoder. The resulting bit stream is then verified by checking the cyclic redundancy check. If the CRC is passed then navigation data information can be extracted from data bits. Beside the decoding of the ephemerides data also the Galileo signal time of transmission has to be retrieved. The time stamps in case of the Galileo messages are inserted at regular intervals into the page layouts. A detailed specification of the message formats and the necessary processing is provided in European Commission (2010).
Figure 6.7a shows the page synchronization and the occurrence of the synchronization patterns within the Galileo E1B I/NAV message stream. The synchronization patterns occur in regular intervals. The sequence shows three frames of the navigation message. According to the specification each frame consists of 7 out of 15 pages which are transmitted in a specified sequence. The CRC was passed before the decoded ephemerides values are stored in an internal format for further processing.




(a) (b)

Figure 6.7: Page synchronization showing the occurrence of the synchronization patterns (a) and received page numbers (b) in case of Galileo E1B I/NAV message

## 6.4 Pseudorange computation

In general satellite navigation observables are ranges which are based on time or phase differences. Three different types of observations can be distinguished (Hofmann-Wellenhof et al. 2008):

- Code pseudoranges;
- Phase pseudoranges;
- Doppler data.

The computation of the pseudoranges is based on the time of signal emission $t^s$ from the satellite and the time of signal reception $t_r$ at the receiver. Whereas the time of signal emission is related to the beginning of the subframe or page (e.g., TOW). The time of reception is related to the receiver clock. Since the two clocks (i.e., satellite clock and receiver clock) are not synchronized it is necessary to take the errors (or biases) of the clocks with respect to a common time system into account. The difference between the time of transmission and the time of reception is equivalent to the time shift $\Delta t$ of the incoming signal and the replica signal within the tracking. Following Hofmann-Wellenhof et al. (2008) the time difference, which corresponds to the signal travel time, can be formulated as

$$t_r - t^s = (t_r + \delta_r) - (t^s - \delta^s) = \Delta t + \Delta\delta, \tag{6.1}$$

where $\delta^s$ and $\delta_r$ denote the satellite clock error and the receiver clock error respectively. By multiplying the signal travel time $\Delta t$ by the speed of light $c$ the code pseudorange $R$ is obtained

$$R_r^s = c\,(t_r - t^s) = c\Delta t + c\Delta\delta = \rho + c\Delta\delta, \tag{6.2}$$

where $\rho$ corresponds to the geometric distance between satellite and receiver. Note that the measured pseudorange is different from the geometric distance because of the clock errors and the influence of signal propagation errors (Sect. 7.2). The time of reception differs from the time of transmission by $\approx 0.07$ s due to the signal travel time, assuming a height of the satellite of 21 000 km. Following Borre et al. (2007) the pseudorange computation is divided into two steps. The first step is the computation of the initial set of pseudoranges. The second one is the computation of subsequent pseudorange observations based on the initial set. For the initial set of pseudoranges the start of one subframe or page has to be identified for each visible satellite. This has already been achieved at the navigation data decoding process. The starting point of each subframe or page has then to be related to a common timing base within the receiver. Usually the receiver clock is used for this task. Within this implementation, due to the post-processing approach, no receiver clock in the common sense can be used. In the present implementation the absolute file pointer position of the digital IF input file during the processing steps is used as a common timing reference. In a next step the channel with the earliest appearance in terms of file pointer position of a subframe or page is used as a reference channel. The time differences (in this case differences in samples) between all remaining channels and the reference channel is computed and converted into seconds. The signal travel time $\Delta t$ is obtained by adding the approximate signal travel time to the previously computed time differences of the start positions of the subframes. The approximate travel time from the satellites at zenith to the earth is about 68 ms in case of GPS (Hofmann-Wellenhof et al. 2008). For Galileo a travel time of 79 ms is assumed. The reference channel, assuming a GPS satellite, has thus a travel time of 68 ms. The initial set of pseudoranges are obtained by converting the travel time into meters. Note that adding the approximate signal travel time can be omitted because the positioning algorithm (Chap. 7) needs the relative offsets between the satellites only.

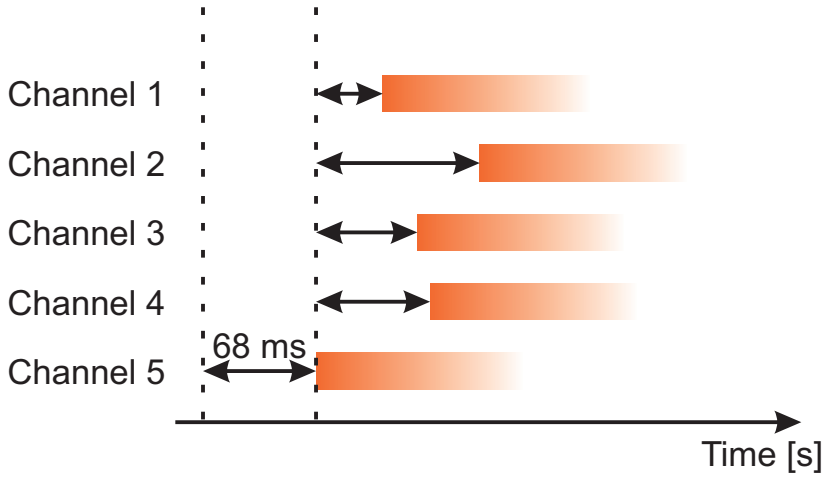



Figure 6.8: Basic example of the pseudorange computation

Figure 6.8 shows a basic example of the pseudorange computation, assuming 5 GPS satellites. Within channel 5, in the present example, the navigation message subframe (red bar) is received first compared to all other channels. The approximate signal travel time

(68 ms) is added to the relative offset between channel 5 and all other channels, as well as to channel 5 itself. Note that the pseudorange accuracy is related to time resolution (samples) of the signal, which is in turn related to the sampling frequency $f_s$ by

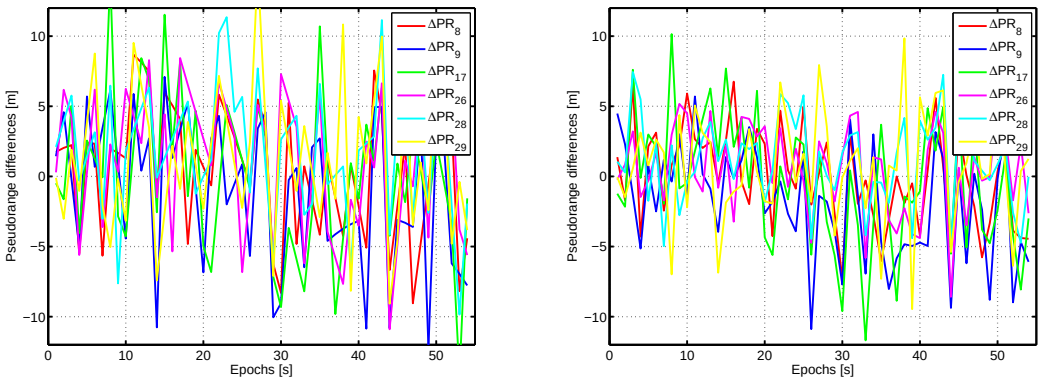
$$\sigma_R = \frac{c}{f_s}, \tag{6.3}$$

where $c$ denotes the speed of light in vacuum. Assuming a sampling frequency of $f_s = 38.192$ MHz, each sample corresponds to $\approx 7.85$ meters. Since the prompt code is precisely aligned with the incoming signal to the nearest sample, the maximum error as a result of the discrete samples will be half the resolution or $< 5$ m. It is possible to use the residual code phase at the end of each ms period to further refine that accuracy.
Using the GIPSIE® simulator it is possible to validate the computed pseudoranges. Table 6.2 compares the computed GPS L1 C/A pseudoranges with the simulated ones. The standard method refers to the algorithm described above, while the enhanced method uses the remaining code phase values for a better accuracy. The mean values are in the range of a few meters.

Table 6.2: Statistics of the difference of computed and simulated GPS pseudoranges

| PRN | Mean [m] | | Std [m] | |
|---|---|---|---|---|
| | standard | enhanced | standard | enhanced |
| 8 | 0.473 | -0.089 | 4.613 | 3.423 |
| 9 | -1.052 | -1.905 | 5.207 | 3.983 |
| 17 | -0.050 | -0.448 | 5.479 | 4.645 |
| 26 | 1.166 | 0.494 | 5.066 | 3.161 |
| 28 | 1.846 | 1.143 | 4.449 | 3.109 |
| 29 | 1.243 | 0.738 | 5.184 | 4.025 |

Note that pseudoranges are not computed for non-tracking channels, for channels which have invalid preambles, or for channels where the CRC of the navigation data fails. Figure 6.9 shows the differences of the computed pseudoranges with respect to the simulated ones. The enhancement using the remaining code phases is visible (Figure 6.9b), compared to the standard method (Figure 6.9a).




(a) (b)

Figure 6.9: Differences between simulated and measured GPS pseudoranges in case of using the standard method (a) and the enhanced method (b)

Figure 6.10 shows the differences of the computed pseudoranges with respect to the simulated ones in case of Galileo. Note that the differences are smaller compared to the GPS pseudoranges.

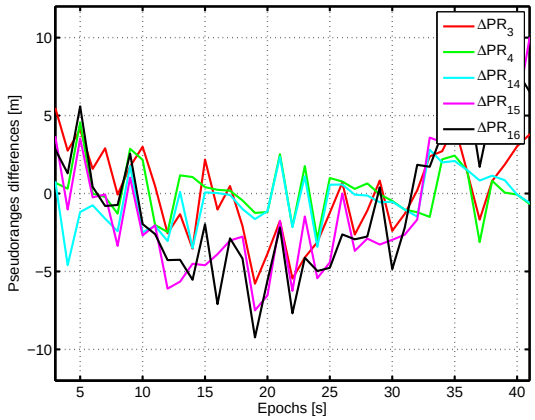


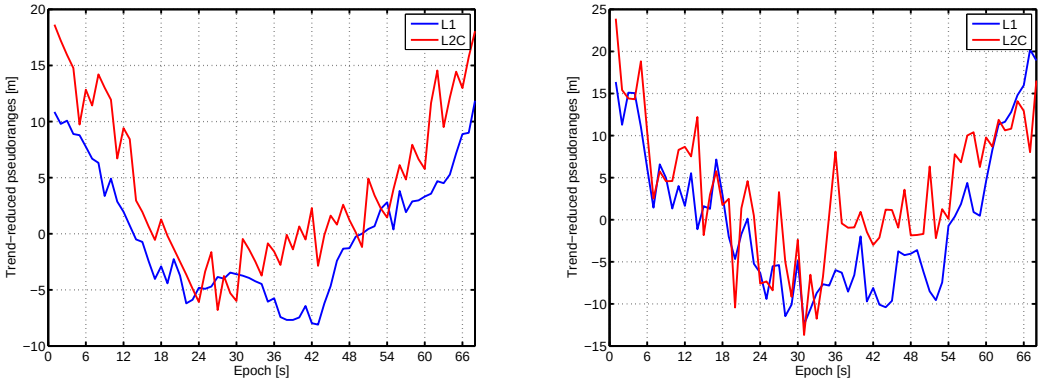

Figure 6.10: Differences between simulated and measured Galileo pseudoranges

Table 6.3 compares the computed Galileo pseudoranges, using the enhanced method, with the simulated ones. The mean values show differences in the order of a few meters, so do the standard deviations.

Table 6.3: Statistics of the difference of computed and simulated Galileo pseudoranges

| PRN | Mean [m] | Std [m] |
|---|---|---|
| 3 | 3.300 | 2.398 |
| 14 | -1.714 | 1.481 |
| 15 | -1.133 | 2.469 |
| 16 | -5.111 | 5.356 |
| 23 | -1.321 | 4.961 |

Figure 6.11 shows a comparison of the computed pseudoranges for GPS L1 C/A and GPS L2C for (a) PRN 29 and (b) PRN 05. Note that a common trend is removed for a better comparability. The L2C pseudoranges have a higher noise than the L1 pseudoranges. Also an offset in the range of a few meters is visible. This offset is mainly due to the ionospheric delay (Sect. 7.3.1).




(a) (b)

Figure 6.11: Differences between GPS L1 C/A and GPS L2C pseudoranges in case of PRN 29 (a) and PRN 05 (b)

At this stage all necessary raw data (i.e., pseudoranges and ephemerides data) are present and, thus, a receiver position computation is possible. The following chapter presents the implemented algorithms for the user position computation. Furthermore, the topics dual-frequency measurements and multi-system measurements are presented in more detail.

# Chapter 7

# Position, velocity, and time estimation

## 7.1 Introduction

The position, velocity, and time module turns the estimated raw measurements (i.e., pseudoranges), the decoded navigation data, and some other attributes into valuable information for the user. Beside position information the module provides, information about velocity, time, and accuracy and alerts the user in case of failures. In todays mass market mainly single-frequency receivers, based on code pseudorange measurements, are used. The positioning accuracy thereby depends on various factors. Hofmann-Wellenhof et al. (2008) mention three different groups of errors, namely satellite-related errors, receiver-related errors, and propagation-medium-related errors. Some of these errors can be modeled, some others not. But even when applying models the resulting position accuracy of a single-frequency receiver is not sufficient for many applications. The increasing number of GNSS signals increases the redundancy and provides the possibility to implement new mitigation and correction strategies. One major advantage of using two different frequencies is the elimination of ionospheric effects and, thus, the significant improvement of the positioning accuracy. Furthermore, due to the redundant measurements to the same satellite, several plausibility and error checks are possible which lead to an improved integrity. Another important aspect is the increasing number of measurements available when combining different GNSS. The GNSS receiver can compute position and time based on a minimum number of four pseudoranges. Depending on the location and time, currently up to 9 or 10 GPS satellites above 10° elevation are visible to the user. If the elevation mask is further reduced up to 12 satellites can be observed. In the future, accounting for Galileo the number of visible satellites will double. And considering multi-frequency measurements the number of total measurements will increase by factor 4 or 5 compared to the situation today. This high redundancy can be used to eliminate error sources or de-weight measurements which can be statistically proven not to fit in the overall solution. This increases the reliability and integrity of the solution as well. After a short introduction of the implemented PVT module this chapter focuses on the treatment of dual-frequency pseudorange observations. Thereby, several strategies for an efficient ionospheric error mitigation are explained and compared. In this context the smoothing of pseudoranges by means of phase observations is discussed as well. The proper combination of different GNSS within the position computation is explained.

## 7.2 Position computation algorithm

In the literature the basic concept for position, velocity, and time estimation is denoted as single point positioning using single-frequency code pseudoranges. It is based on the simplified mathematical model, which can be derived from the geometry between one satellite and the receiver (Hofmann-Wellenhof et al. 2008). The observation equation for one single observation reads:

$$R_r^s(t) = \rho_r^s(t) + c\Delta\delta_r^s(t) + \Delta\text{Iono}_r^s(t) + \Delta\text{Tropo}_r^s(t) + \epsilon_r^s(t), \tag{7.1}$$

where

- $R_r^s(t)$ is the measured code pseudorange in meters at the time epoch $t$ between the receiver $r$ and the satellite $s$;
- $\rho_r^s(t)$ denotes the geometric distance in meters at the time epoch $t$ between the receiver $r$ and the satellite $s$;
- $c$ is the speed of light in vacuum (299 792 458.0 $^{\text{m}}/_{\text{s}}$);
- $\Delta\delta_r^s(t) = \delta_r(t) - \delta^s(t)$ represents the combined clock offset, receiver clock error $\delta_r(t)$ and satellite clock error $\delta^s(t)$, in seconds;
- $\Delta\text{Iono}_r^s(t)$ represents the propagation delay caused by the earth's ionosphere for the signal traveling between satellite $s$ and receiver $r$ in meters;
- $\Delta\text{Tropo}_r^s(t)$ symbolizes the propagation delay caused by the earth's troposphere for the signal traveling between satellite $s$ and receiver $r$ in meters;
- All remaining range errors and corrections (e.g., multipath, measurement noise) are summarized in $\epsilon_r^s(t)$.

The satellite clock error $\delta^s(t)$ can be modeled using the broadcasted satellite clock parameters (Department of Defense 2011, European Commission 2010). Typically the propagation delays introduced by ionosphere and troposphere are corrected using broadcast model parameters and standard models. The mathematical model for the observation equation, neglecting the propagation errors, yields

$$R_r^s(t) + c\delta^s(t) = \rho_r^s(t) + c\delta_r(t) + \epsilon_r^s(t). \tag{7.2}$$

The left side contains all known respectively measured quantities while the right side represents the unknowns. Using Cartesian coordinates representations for the receiver and satellite positions at the time epoch $t$ the geometric distance $\rho_r^s(t)$ can be expressed as,

$$\rho_r^s(t) = \sqrt{(X^s(t) - X_r(t))^2 + (Y^s(t) - Y_r(t))^2 + (Z^s(t) - Z_r(t))^2}. \tag{7.3}$$

Where $X^s(t)$, $Y^s(t)$, and $Z^s(t)$ denote the earth-centered, earth-fixed (ECEF) satellite coordinates, which are computed out of the ephemerides data, and $X_r(t)$, $Y_r(t)$, and $Z_r(t)$ represent the ECEF receiver coordinate triple. The solution of positioning is performed

within an adjustment process following linearization principles. The set of linearized equations can be written in matrix-vector form

$$l = Ax, \tag{7.4}$$

where $l$ denotes the observation vector and $x$ represents the parameter vector (vector of unknowns). The design matrix $A$ for the adjustment considering one GNSS and single-frequency measurements for two satellites $i$ and $j$ to the receiver $r$ reads

$$A = \begin{bmatrix} -\frac{X^i - X_r}{\rho_r^i} & -\frac{Y^i - Y_r}{\rho_r^i} & -\frac{Z^i - Z_r}{\rho_r^i} & 1 \\ -\frac{X^j - X_r}{\rho_r^j} & -\frac{Y^j - Y_r}{\rho_r^j} & -\frac{Z^j - Z_r}{\rho_r^j} & 1 \end{bmatrix}. \tag{7.5}$$

The first three columns correspond to the unknown receiver coordinate triple ($X_r(t)$, $Y_r(t)$, and $Z_r(t)$). The last column corresponds to the receiver clock error $\delta_r(t)$. Note that for a better condition number of the matrix the last column contains 1 instead of $c$. Note that approximate values for the user coordinates are needed due to the linearization. Following Hofmann-Wellenhof et al. (2008) the approximate values can be set to zero, since the equations have to be applied iteratively. The details about the adjustment approach as well as Kalman filter approaches are discussed, e.g., in Hofmann-Wellenhof et al. (2008), Parkinson and Spilker (1996), Borre et al. (2007), or Kaplan (2006).

## 7.3 Dual-frequency algorithm

Beside the errors introduced by the satellites (i.e., satellite orbit, satellite clock), the troposphere, the receiver clock, and local errors (e.g., multipath), the ionosphere represents one of the major error sources. GNSS signals in general are affected by the ionosphere since the signals have to pass through the ionosphere which is a dispersive medium. It causes a frequency dependent delay of the transmitted signal. Since GNSS code pseudorange measurements can be delayed by the ionosphere up to several tens of meters depending on the ionospheric conditions, the ionospheric effect as a primary source of error must be removed or at least mitigated. In order to obtain accurate positions all single-frequency GNSS receivers use an ionospheric model to compensate the ionospheric delay. Concerning single-frequency code measurements, ionospheric models like Klobuchar and NeQuick or grid ionospheric vertical delay models from SBAS are used to mitigate the influence of ionosphere. For an efficient mitigation of ionospheric delay, dual-frequency code measurements are required. The advantage of using two different frequencies is that it is possible to eliminate the ionospheric effects and, thus, to improve significantly the positioning accuracy. This approach is used in geodetic phase measurement applications since decades. Due to the modernization of GPS and the multi-frequency design of Galileo it is now feasible to track civil codes on more than one frequency and, thus, it is possible to virtually eliminate the ionospheric delay by the use of dual-frequency GNSS measurements.

### 7.3.1 Ionospheric effect

The ionosphere represents the uppermost part of the atmosphere, distinguished because it is ionized by solar radiation. It plays an important part in atmospheric electricity and forms the inner edge of the magnetosphere. It prevents the ultraviolet radiation from the sun to reach the earth's surface. The ionosphere is a shell of electrons and electrically charged atoms and molecules that surround the earth, stretching from a height of about 50 km to more than 1000 km. The degree of ionization depends primarily on the sun and its activity. The amount of ionization in the ionosphere varies greatly with the amount of radiation received from the sun and is directly connected to the sun spots and the sunspot cycle ($\sim$ 11 years). The ionosphere is no static phenomenon but varies in location and time. Due to the earth's rotation and the ecliptic there exists a diurnal (daily) effect, a monthly effect as well as a seasonal effect. In winter when the hemisphere is tilted from the sun less solar radiation is received and, thus, the degree of ionization is less than in summer. Within the diurnal cycle the maximum of ionization is reached at 2 p.m. local time, whereas the minimum occurs around 2 hours after midnight. On the earth's surface the radiation also depends on the geographic location, especially on latitude. Also solar flares cause disturbances within the ionosphere. Electromagnetic waves traveling through the ionosphere are affected by the electron particles depending on the density of the particles (total amount of particles) along the signal path - the so-called total electron content (TEC). Furthermore, the ionospheric effects on the signals depend on the frequency of the waves. Beside several effects caused by the ionosphere (e.g., Faraday effect, ionospheric scintillation, etc.) the signal is delayed. This ionospheric signal delay can be significant and, thus, it is of major concern for satellite communication and navigation. More details about the ionosphere can be found in Anderson and Fuller-Rowell (1999) and Jock et al. (1996). Following Hofmann-Wellenhof et al. (2008) and Kaplan (2006), the ionospheric delays for the code pseudorange and phase pseudorange measurements are based on a truncated series expansion of the refractive index and is given by

$$\begin{aligned} \Delta_{\text{code}}^{\text{Iono}} &= \frac{40.3}{f^2}\ \text{TEC} \\ \Delta_{\text{phase}}^{\text{Iono}} &= -\frac{40.3}{f^2}\ \text{TEC}. \end{aligned} \tag{7.6}$$

The ionospheric terms for code and phase observations (7.6), given in meters, show a dependency on the carrier frequency $f$ and on the TEC. The TEC in turn depends on sunspot activities, diurnal and seasonal variations, the azimuth and elevation angle of the satellite, and the receiver position. The TEC is usually computed along the signal path between the satellite and the receiver. The TEC can be measured, modeled, estimated, or eliminated (Hofmann-Wellenhof et al. 2008). From (7.6) follows that GNSS ranging codes are delayed; therefore, the measured code pseudoranges are too long compared to the true geometric range between receiver and satellite. This is in contrast to the phase, which is advanced and therefore the measurements are too short. Measuring the TEC is a discipline of atmospheric monitoring. The modeling of the ionospheric delay is difficult due to the various time dependent influences on the TEC.
Different ionospheric models exist, having different scales in space and time, and many of

them are derived from GNSS ground-based data analysis. These ionospheric delay models are used within GNSS to correct the single-frequency receivers signals.

#### 7.3.1.1 Klobuchar model

Within the GPS navigation messages, 8 parameter of the Klobuchar ionospheric delay model are broadcast to predict ionospheric corrections for single-frequency GPS measurements. The model was developed by John A Klobuchar (Klobuchar 1986). The Klobuchar ionospheric delay model is a simple but strong utility model. It uses a trigonometric cosine function to describe the characteristics of ionospheric diurnal variations. Due to the fact that the Klobuchar model has been completed in early 1980s, confined to the technology of that era, the parameters and the model type are not so accurate, compared to newer models. The algorithm is based on the broadcasted ionospheric coefficients (8 parameters), the receiver coordinates (latitude and longitude), the elevation and azimuth angle to the satellite, and the actual time. A detailed description of the algorithm for computing the ionospheric correction term is given in Hofmann-Wellenhof et al. (2008). Figure 7.1 shows the variation of the ionospheric correction term over one day for different elevation angles. It is visible that the maximum occurs at 2 p.m. and that during the night the effect is reduced to a constant minimum value.
Jock et al. (1996) state that the delay can occasionally rise up to 50 meters for a signal passing vertically through the ionosphere and that satellites near the horizon can suffer a delay even three times higher due to the obliquity. Thus, typically a cut-off angle of 10° elevation is commonly used.

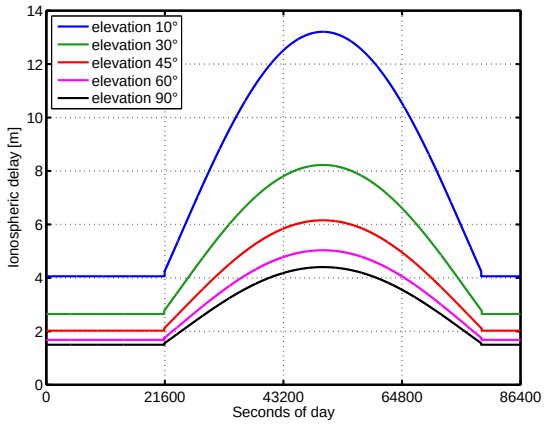
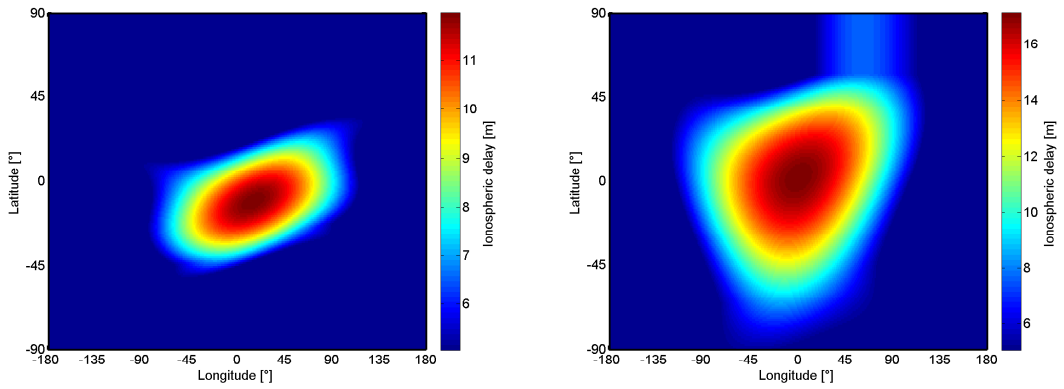



Figure 7.1: Ionospheric delay for different elevation angles over one full day computed using the Klobuchar model ($\phi = 47°$, $\lambda = 15°$, azimuth = 0°, date = May 30th, 2011)

The dependency of the solar ionization level, as modeled by the Klobuchar model, is visualized in Figure 7.2. While Figure 7.2a shows the computed worldwide vertical ionospheric delay using the Klobuchar model at 2 p.m. Graz local time on August 8th, 2009, 7.2b shows the delay on May 30th, 2011, at 2 p.m. Graz local time.
The strengths of the Klobuchar model are its simplicity and the low computation time and processing time needed. The shortcomings of the model, following Farah (2008), are




(a) (b)

Figure 7.2: Klobuchar ionospheric zenith delay for (a) August 8th, 2009, and (b) May 30th, 2011

the low accuracy (only 50% of the ionospheric delay effects can be corrected) and poor behaviour in near-equatorial and high latitude regions.

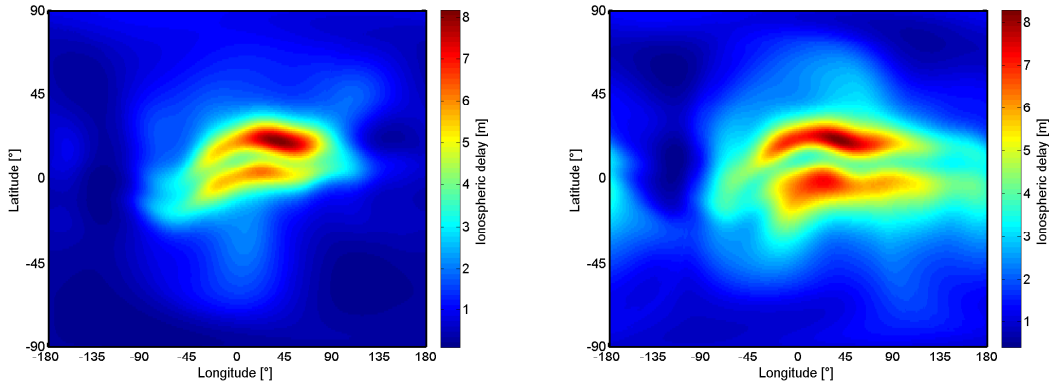
#### 7.3.1.2 NeQuick model

The NeQuick model is the ionospheric model that is foreseen to be used by the Galileo single-frequency user to compensate the ionospheric effect. NeQuick was developed at ICTP Trieste and at the University of Graz (Hochegger et al. 2000, Radicella and Leitinger 2001). The NeQuick model reproduces the TEC along a given ray path as well as electron density distributions for a given month, geographic latitude and longitude, height and time. The model uses different layers for computing the electron density. According to Hochegger et al. (2000) the model consists of two major components. The bottom side model describes the ionosphere below the peak of the F2 layer using a sum of Epstein layers. Whereas the top side model describes the region above the peak of the F2 layer. This component uses a semi-Epstein layer with modeled thickness parameter increasing linearly with height (Arbesser-Rastburg and Cerdeira 2005). A detailed description of the model is provided in Leitinger and Radicella (2002). Following Arbesser-Rastburg (2006) and Arbesser-Rastburg and Jakowsi (2007) the ionospheric correction term by using the NeQuick model can be calculated as follows:

- The receiver retrieves the coefficients $a_0$, $a_1$, $a_2$ for the effective ionization parameter $Az$, the actual time, the satellite position (Keplerian elements), and ionospheric disturbance flags, from the navigation message (European Commission 2010).
- Information about the earth's magnetic field and the ITU, Radiocommunication (sector) (ITU-R) maps are stored in the receiver.
- The receiver computes a rough position estimation ($\phi$, $\lambda$, h) without ionospheric corrections.
- Based on $\phi$ and $\lambda$ the magnetic dip $I$ is computed using the stored magnetic model.

- The modified dip $\mu$ is computed using $\tan\mu = \dfrac{I}{\sqrt{\cos\phi}}$.
- The effective ionization parameter $Az$ is computed using $Az = a_0 + a_1\mu + a_2\mu^2$.
- Based on $Az$ and using the NeQuick model, the electron density is calculated for a point along the satellite-receiver path.
- All discrete points along the signal path are then integrated in order to obtain the TEC.
- The TEC value is then converted to $\Delta^{\text{Iono}}$ using (7.6).

An analytical description of the NeQuick model computation can be found in Memarzadeh (2009). The algorithm requires twelve Comité Consultatif International des Radiocommunications (CCIR) maps, one for each calendar month and providing ionosonde data and a file containing dip latitudes. Some newer versions of the NeQuick algorithm do not depend on the month, and/or use simplified versions of these maps (Hofmann-Wellenhof et al. 2008). Figure 7.3a shows the computed worldwide vertical ionospheric delay using the NeQuick model at 2 p.m. Graz local time on August 8th, 2009, 7.3b shows the delay on May 30th, 2011 at 2 p.m. Graz local time.




(a) (b)

Figure 7.3: NeQuick ionospheric zenith delay for (a) August 8th, 2009 and (b) May 30th, 2011

A comparison of the Klobuchar and NeQuick model was performed within the GAMMA-A project (Rohmer 2011). The strengths of this model according to the investigations are the higher accuracy compared to Klobuchar, and the more appropriate modeling along some areas. Furthermore, the model is able to reflect higher frequencies in local changes. The shortcomings of the model are the high processing power needed for integrating the TEC values along the ray path and the memory needed to load and store the various model parameters. Nevertheless the NeQuick model outperforms the Klobuchar model. Since it is difficult to model the TEC due to the various time-dependent influences, the most efficient treatment of the ionospheric effect is the elimination or estimation of the ionospheric delay

by using dual-frequency observations. Since the ionosphere is a dispersive medium, GNSS observations on different carrier frequencies are affected in a different way (7.6). This fact enables two possible strategies.

### 7.3.2 Elimination algorithm

The first one is the elimination of the ionospheric delay; the second one is the estimation of the ionospheric delay. The elimination as well as the estimation of the ionospheric refraction requires at least dual-frequency observations from one and the same satellite. In case of eliminating the ionospheric effect the observation equation based on code pseudorange measurements (cf. (7.2)) is expanded by a frequency-dependent ionospheric correction term. The observation equations, assuming dual-frequency pseudorange measurements on $f_1$ and $f_2$, read

$$\begin{aligned} R_{f_1} &= \rho + c\Delta\delta + \Delta_{f_1}^{\text{Iono}} + \epsilon_{f_1} \\ R_{f_2} &= \rho + c\Delta\delta + \Delta_{f_2}^{\text{Iono}} + \epsilon_{f_2}. \end{aligned} \tag{7.7}$$

The code pseudoranges between one receiver and one satellite are denoted as $R_{f_1}$ and $R_{f_2}$ and are given in meters. The subscripts $f_1$ and $f_2$ denote the different carrier frequencies where the code is modulated on. The elimination of the ionospheric term can be achieved by using a linear combination of the two equations (7.7). The coefficients $n_1$ and $n_2$ of the linear combination are determined in such a way that the following equation is fulfilled and reads

$$n_1 \cdot \Delta_{f_1}^{\text{Iono}} + n_2 \cdot \Delta_{f_2}^{\text{Iono}} = 0. \tag{7.8}$$

By inserting (7.6) into (7.8) and assuming $n_1 = 1$ the coefficient $n_2$ is given by

$$n_2 = -\frac{f_2^2}{f_1^2}. \tag{7.9}$$

Thus, the ionosphere-free linear combinations reads

$$\left(R_{f_1} - \frac{f_2^2}{f_1^2} R_{f_2}\right) = \left(1 - \frac{f_2^2}{f_1^2}\right)(\rho + c\Delta\delta + \epsilon). \tag{7.10}$$

Bringing the first term of the right side of (7.10) to the left side, the result is the ionosphere-free observation equation for combined code pseudoranges

$$R_{f_1 f_2} = \left(R_{f_1} - \frac{f_2^2}{f_1^2} R_{f_2}\right)\left(\frac{f_1^2}{f_1^2 - f_2^2}\right) = (\rho + c\Delta\delta + \epsilon). \tag{7.11}$$

The unknown parameters, which have to be solved are the receiver coordinates included in $\rho$ and the receiver clock bias $\Delta\delta$. Thus, only the observation vector changes and the design matrix remains untouched compared to the single-frequency algorithm. Note that the term ionosphere-free is not the fully correct terminology because terms of higher order of ionospheric effects are not eliminated using the dual-frequency approach due to some approximations in (7.6) (Hofmann-Wellenhof et al. 2008). The main advantage of

the ionosphere-free combination is the elimination (or more precisely, the mitigation) of ionospheric effects enabling a significant improvement of the positioning accuracy.
The drawback of the ionosphere-free linear combination is the amplification of the measurement noise due to the formation of the linear combination. The amplification can be estimated by applying a variance propagation to (7.11) and reads

$$\sigma^2_{R_{f_1 f_2}} = \left(\frac{f_1^2}{f_1^2 - f_2^2}\right)^2 \cdot \sigma^2_{R_{f_1}} + \left(\frac{f_2^2}{f_1^2 - f_2^2}\right)^2 \cdot \sigma^2_{R_{f_2}}. \tag{7.12}$$

This leads to a factor of about 3 for the amplification of noise in case of L1 and L2 observations. A factor of 2.58 is expected when using L1 and L5 observations. Thus, it is inevitable to put more effort into the computation of the linear combination. Parkinson and Spilker (1996) state that the ionospheric effect can be averaged over time. Thus, a code smoothing by means of phase observations can be applied to the linear combination. The smoothing algorithm is discussed in Sect. 7.5.

### 7.3.3 Estimation algorithm

Another approach, following Hofmann-Wellenhof et al. (2008), targets the estimation of the ionospheric delay. Therefore, the TEC is substituted by a Taylor series expansion (7.6) as function of the observation latitude and the local solar time. The coefficients in the Taylor series are introduced as unknown parameters in the code pseudorange observation equation and estimated together with the other unknown parameters (receiver coordinates and receiver clock error). Based on (7.7) the design matrix for the estimation of the ionospheric influence can be established. The design matrix reads

$$A = \begin{bmatrix} -\dfrac{X^i - X_R}{\rho^i_R} & -\dfrac{Y^i - Y_R}{\rho^i_R} & -\dfrac{Z^i - Z_R}{\rho^i_R} & 1 & 1 & 0 \\ -\dfrac{X^i - X_R}{\rho^i_R} & -\dfrac{Y^i - Y_R}{\rho^i_R} & -\dfrac{Z^i - Z_R}{\rho^i_R} & 1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}. \tag{7.13}$$

The two rows in matrix (7.13) correspond to one $f_1$ and one $f_2$ observation to one single satellite $i$. This simple approach is not possible, because it results in a singular design matrix. Columns 4, 5 and 6 of the matrix above become linearly dependent. Additionally the problem arises that for all satellites in view additional terms have to be considered, since the delay for each signal path is different. Thus, more unknowns (three receiver coordinates, one receiver clock error, and two ionospheric unknowns per satellite) than observables appear and the system of equations is not solvable in this case. The solution of this problem is to estimate a combined ionospheric effect for $f_1$ and $f_2$. The solution is derived by using (7.8) in the ionosphere-free form

$$\Delta^{\text{Iono}}_{f_2} = \frac{f_1^2}{f_2^2} \Delta^{\text{Iono}}_{f_1}. \tag{7.14}$$

The corresponding pseudorange equations can be written as

$$\begin{aligned} R_{f_1} &= \rho + c\Delta\delta + \Delta_{f_1}^{\text{Iono}} \\ R_{f_2} &= \rho + c\Delta\delta + \frac{f_1^2}{f_2^2}\Delta_{f_1}^{\text{Iono}}. \end{aligned} \tag{7.15}$$

The design matrix, assuming two satellites $i$ and $j$ providing pseudorange measurements on two frequencies, for this approach reads

$$A = \begin{bmatrix} -\frac{X^i - X_R}{\rho_R^i} & -\frac{Y^i - Y_R}{\rho_R^i} & -\frac{Z^i - Z_R}{\rho_R^i} & 1 & 1 & 0 & \dots \\ -\frac{X^i - X_R}{\rho_R^i} & -\frac{Y^i - Y_R}{\rho_R^i} & -\frac{Z^i - Z_R}{\rho_R^i} & 1 & \frac{f_1^2}{f_2^2} & 0 & \dots \\ -\frac{X^j - X_R}{\rho_R^j} & -\frac{Y^j - Y_R}{\rho_R^j} & -\frac{Z^j - Z_R}{\rho_R^j} & 1 & 0 & 1 & \dots \\ -\frac{X^j - X_R}{\rho_R^j} & -\frac{Y^j - Y_R}{\rho_R^j} & -\frac{Z^j - Z_R}{\rho_R^j} & 1 & 0 & \frac{f_1^2}{f_2^2} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{7.16}$$

Now the solution of the system of equations becomes possible, nevertheless at least five satellites are needed to solve the system. For the convergence of the algorithm it is necessary to account for the estimated ionospheric delay within the observation vector. Test results of both algorithms - the one for the elimination and the one for the estimation - are shown and discussed in Chap. 8.

### 7.3.4 Ionospheric correction algorithm

Within the current implementation a slightly different approach is used for ionospheric correction. The advantage in this approach is, that the design matrix does not contain any additional parameters or has to be multiplied by a factor. The mathematical models described before allow to estimate the ionospheric error on $f_1$ using $f_1$ and $f_2$ pseudorange measurements. Substituting (7.6) into (7.7) for the corresponding frequencies and neglecting the noise for the moment leads to

$$\begin{aligned} R_{f_1} &= \rho + c\Delta\delta + \frac{40.3}{f_1^2}\text{TEC} \\ R_{f_2} &= \rho + c\Delta\delta + \frac{40.3}{f_2^2}\text{TEC}. \end{aligned} \tag{7.17}$$

Forming the difference between the pseudoranges the so-called geometry related terms vanish and (7.17) yields

$$R_{f_1} - R_{f_2} = \frac{40.3}{f_1^2}\mathrm{TEC} - \frac{40.3}{f_2^2}\mathrm{TEC} \tag{7.18}$$

and, after multiplying the equation by $f_2^2/(f_2^2 - f_1^2)$ and re-substitution of (7.6), the ionospheric delay on $f_1$ can be found as

$$\Delta_{f_1}^{\mathrm{Iono}} = \frac{f_2^2}{f_2^2 - f_1^2}(R_{f_1} - R_{f_2}). \tag{7.19}$$

When computing a position solution using the linear combination of (7.19) no benefit of the dual frequency algorithm is visible (Sect. 8.3). As mentioned in Tetewsky et al. (2009) small variations in signal delay appear among the various signal paths within hardware of the spacecraft. It is not possible for the satellite designers to precisely eliminate them. Thus, GNSS signals do not emerge from the satellite antennas at exactly the same time and from the same location. If these offsets are not accounted within the PVT the calculated positions will be inaccurate. Thus, a slightly different algorithm has to be used (Department of Defense 2011, Tetewsky et al. 2009). This algorithm is denoted as 'the modernized ionosphere-free pseudorange algorithm' and contains a mixture of the previous algorithm and new parameters denoted as inter-signal corrections (ISCs). The modernized ionosphere-free pseudorange linear combination based on (7.11) reads

$$R_{f_1 f_2} = \left(R_{f_1} + c\ \mathrm{ISC_{L1C/A}} - \frac{f_2^2}{f_1^2}\left(R_{f_2} + c\ \mathrm{ISC_{L2C}}\right)\right)\left(\frac{f_1^2}{f_1^2 - f_2^2}\right) - c\ \mathrm{T_{GD}}. \tag{7.20}$$

Thus, the ionospheric delay reads

$$\Delta_{f_1}^{\mathrm{Iono}} = \left(\frac{f_2^2}{f_2^2 - f_1^2}\right)\left(\left(R_{f_1} + c\frac{f_1^2}{f_2^2}\ \mathrm{ISC_{L1C/A}}\right) - \left(R_{f_2} + c\ \mathrm{ISC_{L2C}}\right)\right) + c\ \mathrm{T_{GD}}. \tag{7.21}$$

The ISC values are defined as the delay difference relative to the SV delay error of L1 P(Y)-code. The ISC values are broadcasted within the CNAV navigation message. The TGD value is already transmitted via the L1 navigation message. In case of using P-code measurements (7.21) reduces to (7.19). GPS does not provide ISC parameters but broadcasts only message type 0 (dummy message) instead (Sect. 6.3). Instead of using the ISC parameters, differential correction biass (DCBs), obtained from CODE (Centre for orbit determination in Europe) have been used for the compensation. Additional information regarding the need and the usage of inter-signal corrections can be found in Tetewsky et al. (2009). Following Kaplan (2006) the ionospheric delay on $f_2$ can be obtained by multiplying $\Delta_{f_1}^{\mathrm{Iono}}$ by $f_1^2/f_2^2$. The ionosphere-free pseudorange is, thus, obtained by subtracting the estimated ionospheric delay $\Delta_{f_1}^{\mathrm{Iono}}$ from the pseudorange $R_{f_1}$. This approach has the advantage that beside reducing the ionospheric effect the design matrix does not contain any additional parameters or has to be multiplied by a factor. Note that only one pseudorange measurement per satellite is used in this approach for the adjustment. Using both corrected pseudoranges would not add any additional information to the system of equations. Another advantage is the possibility to compare the estimated delay with the computed ionospheric delay from the Klobuchar and NeQuick model. This makes it possible to estimate roughly the unaccounted ionospheric delay within the model and to

use this information within the weighting of the observations. Since it is unlikely that a receiver gathers dual-frequency measurements from all visible satellites all the time, it is necessary to use a decision tree as shown in Figure 7.4 in order to decide, which corrections or algorithm shall be applied. Furthermore the receiver may perform measurements from different frequency combinations from different systems (e.g., GPS L1/L2, and Galileo E1/E5).

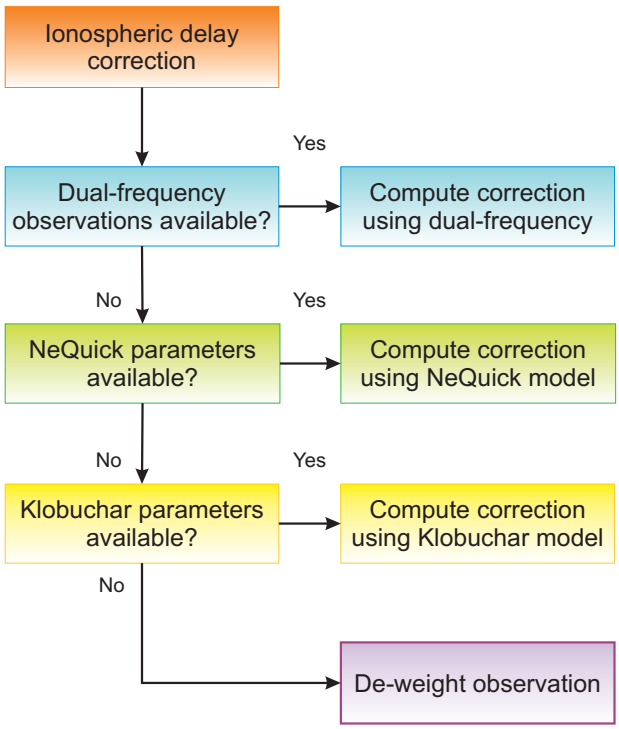



Figure 7.4: Implemented ionospheric model decision tree

The question now arises how all these measurements are corrected for ionosphere and, depending on the way they are corrected, introduced into the observation equation to compute the receiver position finally. The appropriate tool to differentiate the various correction methods in the observation equation is the weight matrix, which is introduced in the adjustment process or the Kalman filter respectively. The weight matrix is a diagonal matrix which weights the different measurements according to different criteria. A high weight will result in a strong reflection of a measurement within the solution, whereas a low weight means that a measurement will add few information to the solution. Numerous different weighting schemes exist in literature. Most of them use the satellite elevation and the signal-to-noise ratio or the carrier-to-noise power density for computing the weight factor. The implemented weighting algorithm uses an elevation-dependent weighting for all observations, as an alternative a weighting scheme based on the signal-to-noise ratios can be used. The problem is, that this algorithm does not take dual-frequency observations into account. Thus, for a proper weighting, in case of having single- and dual-frequency observations, an additional weighting factor is needed. Otherwise the higher accuracy of dual-frequency observations would be neglected. Since the model-based ionospheric corrections are not able to compensate the full ionospheric error (Sect. 7.3.1.1), a measure

about the accuracy of the models are needed. This can be achieved by computing the ratio of the estimated ionospheric delays, based on dual-frequency measurements, and the computed model-based ionospheric corrections. By dividing the estimated ionospheric delay based on dual-frequency measurements by the computed model-based delay a ratio for each satellite, providing dual-frequency measurements, can be obtained. This ratio reflects the uncertainty of the used ionospheric models and, thus, can be used for the weighting. A common ratio, between single- and dual-frequency observations, is obtained by averaging all the individual ratios of all available dual-frequency measurements. This common ratio is then applied inversely as an additional weighting factor to the single-frequency measurements. Thus, by applying this additional weighting factor, both, the dual-frequency ionospheric corrected pseudoranges and the single-frequency model-corrected pseudoranges can be used in a common adjustment. The same principle can be applied when using unsmoothed and smoothed pseudorange measurements (Sect. 7.5).

## 7.4 Tropospheric delay correction

The troposphere is the lower part of the atmosphere and is located between zero height and approximately 40 km. Unlike the ionosphere, the troposphere is a nondispersive medium with respect to radio waves up to frequencies of 15 GHz. Due to this independence of the frequency dual-frequency measurements cannot be used to eliminate the tropospheric effect. The tropospheric delay depends on the local temperature, pressure, and relative humidity (or water vapour pressure). If the effect is left uncompensated, the delay can vary from about 2.3 m for a satellite at zenith and a user at sea level to about 25 m for a satellite at low elevation (Hofmann-Wellenhof et al. 2001). The tropospheric delay along a slant path can be divided into a hydrostatic and a wet component. The zenith delay can be obtained by using mapping functions. While the zenith hydrostatic delay can be easily modeled it is more difficult to model the wet component (Krueger et al. 2004). The excess in path length due to the tropospheric delay influences mainly the height component of the position solution but has a lower impact than the ionospheric delay. The troposphere strongly depends on local weather situation, which cannot be modeled using a global model and transmitted via the GNSS systems like the ionospheric model. Therefore there is no information on the troposphere within the navigation message. For users without access to meteorological data a-priori models have to be used. Standard tropospheric delay correction models are used to compensate this effect. Among others the most prominent are:

- Saastamoinen model (Saastamoinen 1973);
- Hopfield model (Hopfield 1969);
- Modified Hopfield model.

The difficulty in modeling the troposphere is the estimation of the atmospheric parameters (i.e., temperature, pressure, and water vapor) along the signal propagation path. Simple models use surface measurements which do not reflect the accurate atmospheric conditions, especially the water vapor. Since the receiver has no access to meteorological data only data

stored in the receiver, referred to as blind mode, are used in this case. The implemented PVT module uses the modified Hopfield model by default, but the Saastamoinen model and the Hopfield model can be used on the user's choice. More information on the implemented tropospheric models is given in Spilker (1996).

## 7.5 Code pseudorange smoothing

As discussed previously (Sect. 7.3.2), the linear combination of dual-frequency pseudorange measurements amplifies the measurement noise. Thus, under normal conditions the dual-frequency solution will show no benefit, due to the increased noise. One method of reducing noise in the pseudorange measurements relies on the phase observations. The advantage of this procedure is that the phase observations are more precise and have a lower noise characteristic than code observations. Additionally the ambiguity issue, which arises when dealing with phase observations, can be neglected. Thus, an algorithm for code pseudorange smoothing by means of phase pseudoranges is implemented in software. The algorithm uses a time-dependent weight factor for a proper weighting of pseudorange and phase measurements (Hatch filter). The implemented algorithm is stated in Hofmann-Wellenhof et al. (2008). The smoothed pseudorange is obtained by

$$R(t_i)_{sm} = \omega R(t_i) + (1-\omega)(R(t_{i-1})_{sm} + \lambda(\Phi(t_i) - \Phi(t_{i-1}))). \tag{7.22}$$

Where $R(t_i)$ and $\Phi(t_i)$ denote the code pseudoranges and phase measurements for the current epoch; $\omega$ represents a weight factor. For the first epoch $i = 1$ the weight factor is set to 1. Therefore the code pseudorange has the full weight. For the consecutive epochs the weight of the code pseudorange is continuously reduced by a value $\Delta\omega$ and, thus, the relative influence of the phase measurements is increased. The amount of reduction of the weight factor from epoch to epoch controls the behavior of the algorithm. Due to the time-dependent weighting of the observations the algorithm needs several epochs to achieve the best results. Note that if a constant weight factor of $\omega = 0.5$ is used (7.22) represents the unweighted smoothing algorithm. It is possible to use the time-dependent weighting factor, as an additional term within the weight computation. In this case a smoothed pseudorange has a stronger influence on the position solution than an unsmoothed one. The effect of pseudorange smoothing is shown in Figure 7.5. On the top, the raw pseudoranges measured at 1 Hz for 250 epochs (after eliminating the trend due to satellite motion) $\Delta$PR are shown. The second graph from the top shows an unweighted smoothing algorithm (Hofmann-Wellenhof et al. 2001), while the last three plots visualize the weighting effect by using different weight reductions ($\Delta\omega = 0.02,\ 0.01$, and 0.005). The weight reduction from epoch to epoch clearly shows the decreasing influence of the pseudorange measurement and the increasing impact of the carrier measurement. In case of a cycle slip, the algorithm would fail. Therefore the phase measurements have to be checked for cycle slips using consecutive measurement epochs and Doppler shift data. After a cycle slip has occurred, the weight is reset to 1, which eliminates the influence of the cycle slip but also restarts the smoothing algorithm for the specific satellite. It is possible to repair the cycle slip in case of measurement redundancy (Hofmann-Wellenhof et al. 2008). It is also possible to use phase differences obtained by the integrated Doppler shifts.

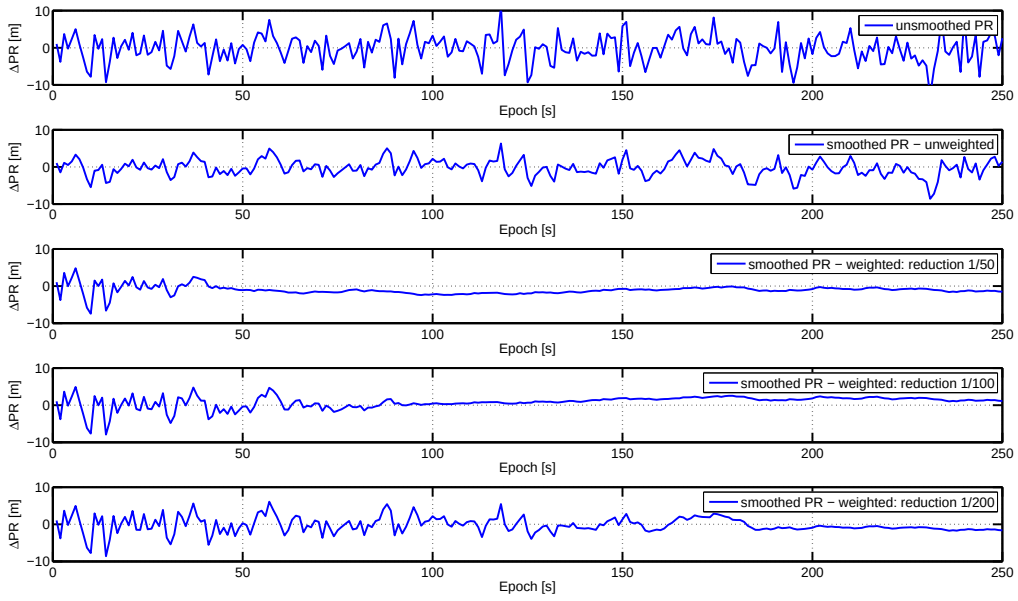



Figure 7.5: Comparison of different smoothing algorithms

The advantage of using integrated Doppler shifts is that they are insensitive to cycle slips. The disadvantage is that they cannot be used for ambiguity determination. In case of dual-frequency observations it is possible to smooth different pseudoranges on different frequencies separately and perform the linear combination afterwards.

## 7.6 Combination of GPS and Galileo

The existence of more than one fully operational GNSS provides benefits to users worldwide. Combining multiple global navigation satellite systems enables improved user capabilities compared to one system alone. A combination of GPS and Galileo, for instance, will at least double the number of GNSS signals available to the user. Thus, several benefits arise from a combination of different GNSS. Several studies have shown that an integration of two systems provides a better geometry and higher availability (O'Keefe 2001, Lachapelle et al. 2002, O'Keefe et al. 2002). This allows a better blunder detection and higher reliability of the combined position solution. It offers an improved satellite availability in difficult environments such as urban canyons. Note that the expected gains in the position accuracy are mainly associated with the improved satellite geometry. As a figure of merit the dilution of precision (DOP) (Hofmann-Wellenhof et al. 2001) is used in this case. The main benefits of such a combination are a higher availability and an improved reliability of the position solution. During the design and definition phase of Galileo the interoperability of the two systems was of major concern. As already discussed in Chap. 2, GPS and Galileo will share two frequency bands (L1/E1 and L5/E5). It is possible to process these signals jointly although the signal modulation is different. Beside

the differences in the signal structure, two other important aspects have to be considered when integrating GPS and Galileo. The first one lies in the differences in the coordinate reference frames. The second one is the different time reference systems. The two aspects are discussed in the following section. Following Hofmann-Wellenhof et al. (2008) different levels of system integration can be distinguished:

- Separately: two systems are used completely separately. If one system fails the other is used.
- Loosely: Each system outputs position information, which is combined to one solution by, e.g., averaging.
- Closely: Raw measurements are obtained from each system independently. Those measurements are then integrated with a PVT solution (adjustment or filter).
- Tightly: The position solution is used to aid the measurement process (predicting Doppler and time shifts).
- Deeply: The observations are not only used to compute a combined position solution, but the information is also used in the other system to aid the measurement process.

For the present development the closely coupled integration was chosen because it is easy to implement and takes into account all the measurements.

### 7.6.1 Time and reference frames for GPS and Galileo

Both Galileo and GPS use the international terrestrial reference system (ITRS) as the basis for their reference system. The realizations for GPS and Galileo are independent. GPS uses the World Geodetic System 1984 (WGS-84) as a coordinate reference frame (Department of Defense 2011), while Galileo uses the Galileo terrestrial reference frame (GTRF) (Gendt et al. 2010, European Commission 2010). Since the reference system influences the satellite coordinates and, thus, the user position, the satellite coordinates must be transformed into a common system. Such a coordinate frame transformation is usually achieved by using a 7-parameter Helmert transformation. Any errors within the transformation can be considered, following Hofmann-Wellenhof et al. (2008), as orbital errors.
Due to the specifications the difference between international terrestrial reference frame (ITRF) and GTRF lies in the order of three centimeters (Diggelen 2009). According to Lachapelle et al. (2006) the differences between WGS-84 and ITRF are in the order of two to three centimeters. Thus the difference between the GTRF and WGS-84 is the order of a few centimeters. In case of GNSS single point positioning the difference can be neglected. For very precise applications the differences between the two frames are significant and must be considered. More information on the interoperability of the reference frames for GPS and Galileo can be found in Lachapelle et al. (2006).
While the differences in the coordinate reference frames can be neglected for SPS, the time difference between GPS and Galileo is of major concern. The pseudoranges determined with GPS are referenced to the GPS system time (GPST), while the Galileo measurements use the Galileo system time (GST) as a reference. The time difference between GPST and

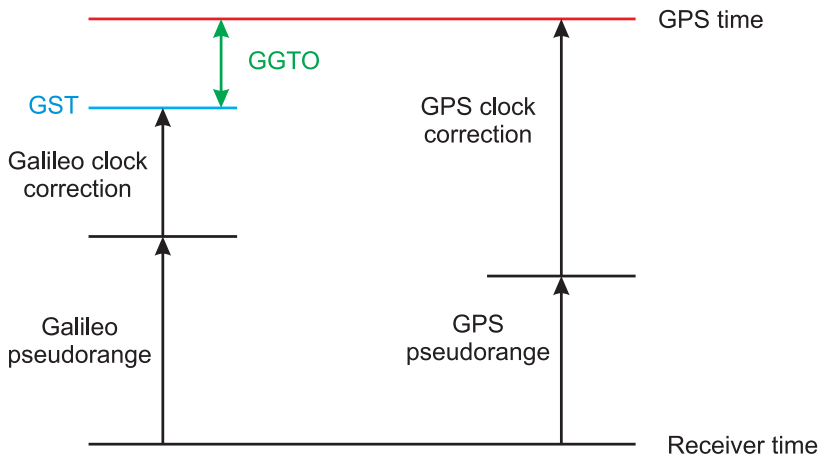



Figure 7.6: Visualization of the Galileo GPS Time Offset (GGTO) with respect to GPST and GST (converted to time)

GST is denoted as GPS to Galileo time offset (GGTO). This resulting offset must be taken into account in order to take advantage of combining GPS and Galileo. Figure 7.6 shows the relation between GST, GPST, clock corrections, and receiver time. The offset between GST and GPS time will be 5 nanoseconds at maximum according to Vanschoenbeek et al. (2007). This offset causes a bias between GPS and Galileo measurements estimated by a combined GNSS receiver. When applying the broadcasted satellite clock correction parameters, measured Galileo pseudoranges will be corrected to GST, and GPS pseudoranges to GPST. This leads to an additional error in the position solution. Following Vanschoenbeek et al. (2007) three different methods for GGTO treatment exist: GGTO determination at user level, at system level, and a combination of these two methods. At user level, GGTO can be estimated from GPS and Galileo measurements as an additional unknown in the user navigation solution. This increases the number of unknowns, requiring at least five measurements to be available to calculate a position solution, receiver clock bias, and GGTO. Especially in environments with limited satellite visibility, such as urban canyons, this might affect negatively the availability. Note that this method is the one used in GPS/GLONASS receivers. In contrast, at system level, the GGTO can be determined by the GPS and Galileo ground segments, predicted and broadcast as a part of the navigation message. In this case within the PVT solution, the observations are corrected by the broadcast GGTO value. Both systems will include the GGTO within the navigation messages. Thus, only four parameters need to be solved. More detailed information on GGTO can be found in Piriz et al. (2006). Both approaches have their pros and cons. At the system level approach the user has to wait until the necessary parameters are received, while at the user level at least five observations are needed for the position computation. In constrained environments the prerequisite of five available satellites might not be fulfilled. For users who want to have a high availability and to calculate the GGTO by their own, a combination of the two previous methods can be used.

### 7.6.2 Combined position estimation

Within the current development the user level approach of determining the GGTO is used. Although at least five satellites are needed for a combined position computation, the receiver has not to wait for the reception of the GGTO parameters. Thereby, the GGTO is determined within the adjustment and the user can select which reference time (GPST or GST) shall be used. Considering two GPS observations (superscript G) and three Galileo observations (superscript E) the design matrix (7.5) and the unknown vector reads

$$Ax = \begin{bmatrix} -\frac{X^{Gi}-X_r}{\rho_r^{Gi}} & -\frac{Y^{Gi}-Y_r}{\rho_r^{Gi}} & -\frac{Z^{Gi}-Z_r}{\rho_r^{Gi}} & 1 & \tau^G \\ -\frac{X^{Gj}-X_r}{\rho_r^{Gj}} & -\frac{Y^{Gj}-Y_r}{\rho_r^{Gj}} & -\frac{Z^{Gj}-Z_r}{\rho_r^{Gj}} & 1 & \tau^G \\ -\frac{X^{Em}-X_r}{\rho_r^{Em}} & -\frac{Y^{Em}-Y_r}{\rho_r^{Em}} & -\frac{Z^{Em}-Z_r}{\rho_r^{Em}} & 1 & \tau^E \\ -\frac{X^{En}-X_r}{\rho_r^{En}} & -\frac{Y^{En}-Y_r}{\rho_r^{En}} & -\frac{Z^{En}-Z_r}{\rho_r^{En}} & 1 & \tau^E \\ -\frac{X^{Ek}-X_r}{\rho_r^{Ek}} & -\frac{Y^{Ek}-Y_r}{\rho_r^{Ek}} & -\frac{Z^{Ek}-Z_r}{\rho_r^{Ek}} & 1 & \tau^E \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \Delta X_r \\ \Delta Y_r \\ \Delta Z_r \\ c\delta t \\ c\delta_{\mathrm{GGTO}} \end{bmatrix}. \tag{7.23}$$

The terms $\tau^G$ and $\tau^E$ define the GST or GPST used as a reference time. In case GPST is used as a reference time $t$, all Galileo measurements need to use the GGTO, thus, $\tau^G = 0$ and $\tau^E = 1$. In this case the receiver clock error $\Delta t$ is related to GPST. It must be mentioned that at least two satellites from one system should be available. If only one observation is available this one will only determine the time offset, but will not contribute to the position solution (Hofmann-Wellenhof et al. 2008). Results of combining GPS and Galileo are discussed in Chap. 8.

## 7.7 Implemented PVT module

Figure 7.7 shows the block diagram of the implemented PVT software module. The PVT module is capable of processing pseudorange, carrier-phase, and Doppler measurements, as well as the decoded navigation parameters. Before the position solution computation is performed several preprocessing steps are necessary. In a first step a plausibility check on the received pseudorange measurements is done. The measurement time of the different systems and signals is checked to ensure that only measurements of the same epoch are processed. If there are carrier measurements available, pseudorange smoothing (Sect. 7.5) is performed afterwards. In parallel the necessary correction terms for the ionosphere, troposphere, and satellite clocks are computed and added to the smoothed pseudorange observations. In case of dual-frequency measurements a linear combination for reducing the ionospheric effect is formed (Sect. 7.3). Out of the broadcast ephemerides data and the

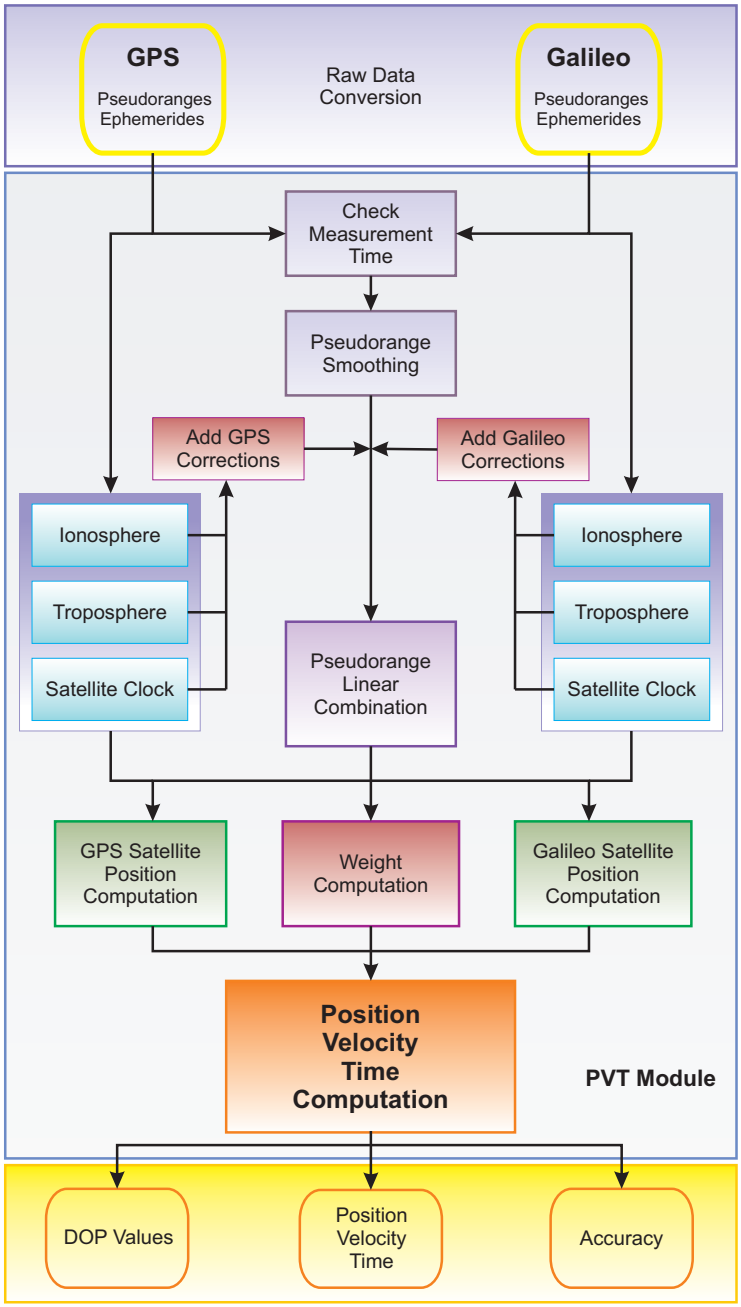



Figure 7.7: High-level architecture of the implemented position, velocity, and time module

measurement time stamps, the positions of the visible satellites are computed. The GPS as well as the Galileo ephemerides processing module computes ECEF coordinates of the corresponding satellites depending on the measurement time. Inputs to the ephemerides processing module are either GPS or Galileo orbit parameters, broadcast through the navigation data in terms of Keplerian parameters and their harmonic correction terms. The GPS ephemerides processing follows the algorithm as given in Department of Defense (2011). The standard reference frame for GPS positioning is the well-known WGS-84. While Galileo ephemerides processing follows the algorithm as outlined in European Com-

mission (2010). The reference frame for Galileo is GTRF. The combination of both systems is discussed in Sect. 7.6. Thereby, a selection is made considering the elevation of the satellite. Afterwards the weight of the observations is computed based on the elevation, the S/N, the weight of the smoothing and the ionospheric correction term. Within the adjustment the position, velocity, and time information is estimated. Thereby, the position and time information is computed within one adjustment, while for the velocity computation the measured Doppler values are used.

Beside the two stage adjustment algorithm the PVT module can be configured to use a Kalman filter for the position, velocity, and time estimation and prediction. The Kalman filter uses a dynamic model which is based on the rules of uniform motion for the measurement prediction. The output to the user of the PVT module contains the navigation solution (position, velocity, and time), statistical data (accuracy and DOP values), as well as integrity information. Either a proprietary format or the National Marine Electronics Association (NMEA) 0813 format can be used. Additional information about visible satellites (azimuth and elevation) can be obtained as well.

The implemented PVT module was not solely developed for this software-based receiver, but for different projects and applications at TeleConsult Austria GmbH by the author of this thesis. The PVT solution is capable of real-time processing raw data (pseudoranges and phase observations) at a maximum data rate of 10 Hz. Beside different proprietary interfaces the PVT has the possibility to use receiver independent exchange (RINEX) 2.11 or RINEX 3.0 data files as an input. There exists the possibility to use AGNSS data, as well as EGNOS data to enhance the position accuracy. Additionally an acquisition aiding algorithm can be used to speed-up the acquisition process. Also algorithms for precise point positioning (PPP) are implemented. The PVT solution was used within the GSA Project GAMMA, the ASAP projects SoftGNSS and RA-PPP, and the GSA project GAMMA-A. Currently the PVT is further enhanced within the GSA projects GENEVA, ASPHALT, and the ASAP project RT-PPP. In the near future it is envisaged to extend the PVT module for GLONASS capabilities.

# Chapter 8

# Results

The tests and verification tasks have been performed not only on the overall software-based receiver but also on the single module level. Within this chapter selected results for different data sets are presented and discussed. Two different types of tests have been performed. In a first step simulated data were used to verify the algorithms. These results are discussed in Sect. 8.2. In a second step recordings of real-world data have been used to evaluate the performance under realistic conditions (Sect. 8.3). At the end of the chapter a few words about the execution time and the computational load are given.

## 8.1 Test cases

Within the testing and verification process of the current development, the two main components - the baseband processing (acquisition and tracking) and the position, velocity, and time software - of the receiver have been tested and validated. The best way of testing are the so-called closed loop tests, using a GNSS constellation and performance simulation tool as an input source. The comparison of the user defined input parameters with the results of either the baseband or the PVT part, provides valuable information about the performance and behavior of the tested module. To verify the results, the performance, and the robustness of both the whole receiver and the individual subroutines (e.g., algorithms) have to be tested.

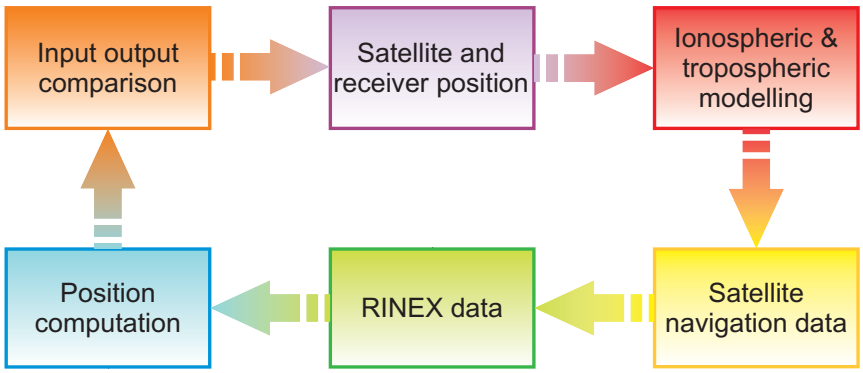



Figure 8.1: Closed loop simulation

Within a closed loop test (Figure 8.1) the operator defines the receiver position, the satellite constellations, and the environmental parameters. Based on these settings the simulator computes the satellite orbits, including satellite errors, the receiver movement and consequently the pseudorange measurements. The propagation effects (e.g., ionosphere and

troposphere) are added to the pseudorange measurements and the navigation messages are generated. Depending on the selected raw data output format either the pseudorange measurements and navigation data are converted into RINEX files or an IF signal is generated. The output of the simulator is then fed into the receiver. While the IF signals are used to verify and test the overall system, the RINEX files are used to evaluate the PVT module only. The position solution, computed by the receiver, is then compared with the input parameters. Simulations provide the necessary flexibility and repeatability which is necessary when developing and implementing new algorithms, but they are not able to simulate all effects which occur in reality. In order to evaluate the performance under realistic conditions recorded real-world data sets have been used. Thus, the testing and verification of the overall system as well as of the implemented modules has been done using different sources of input. For verification purposes, to ensure the proper implementation of the algorithms, simulated data sets have been used. For the evaluation of the performance under real conditions, recorded IF signals and raw data have been used. The following test cases have been investigated:

- Simulated data sets:
  - Raw data (i.e., pseudoranges and navigation data) simulated by the GIPSIE®–SCS for verification of the PVT module;
  - Digital IF signals simulated by the GIPSIE®–IFS for verification of the overall receiver.
- Recorded real-world data sets:
  - Recorded raw data (pseudoranges and navigation data recorded by a commercial receiver) for testing the PVT module;
  - Recorded digital IF signals using the FhG-IIS multiband front-end for testing the overall receiver.

All simulated data sets have been generated by the GIPSIE®–SCS and GIPSIE®–IFS, which has been developed by TCA. The GIPSIE® simulator provides the possibility to generate pseudoranges and navigation data in the RINEX format or digital IF files which then are used within the closed loop simulation for verification and testing. More information about the GIPSIE® simulator and the implemented models can be found in Berglez et al. (2010), Berglez et al. (2009), and Abart (2009). The first part of the simulations covers the investigation of the PVT module, while the second proofs the baseband software in combination with the PVT. The main subroutines under test within the signal processing module are: the acquisition, the tracking loops, the navigation message extraction, and the pseudorange measurement calculation. The results of testing the acquisition as well as the tracking and the preprocessing module are discussed in Sects. 4.6, 5.6, and 6.4. The recordings of the digital IF signals were done using the described RF front-ends. A detailed description of the data sources used is provided in Chap. 3.

## 8.2 Position solutions with simulated data

The nominal GPS and Galileo constellations (Chap. 2) have been used as input for all presented scenarios. For the receiver a static position was simulated at the WGS-84 coordinates summarized in Table 8.1.

Table 8.1: Simulated receiver position in geographical ($\phi$, $\lambda$, $h$) and Cartesian ($X$, $Y$, $Z$) coordinates

| Simulated receiver coordinates | | | |
|---|---|---|---|
| $\phi_{\text{WGS-84}}$ | 47.0° | $X_{\text{WGS-84}}$ | 4209209.69 m |
| $\lambda_{\text{WGS-84}}$ | 15.0° | $Y_{\text{WGS-84}}$ | 1127854.34 m |
| $h_{\text{WGS-84}}$ | 10 m | $Z_{\text{WGS-84}}$ | 4641772.10 m |

The receiver position in combination with the nominal constellation results in 16 visible satellites (8 GPS and 8 Galileo). The satellite specific errors were modeled in all scenarios. Different scenarios containing ionospheric and tropospheric effects were generated to show the impact of the propagation effects on the position solution as well as the impact of compensating these effects by using appropriate models. Each raw data simulation lasts for 60 minutes and was simulated with an update rate of 1 Hz. Figure 8.1 shows a skyplot (azimuth and elevation angle) of the simulated satellite constellation. The GPS satellites are visualized in red, while the Galileo satellites are plotted in blue. Additionally the Galileo satellite ID's, originally numerated from 1 to 92 (European Commission 2010), are numbered from 62 to 153 for a better distinction in this thesis.

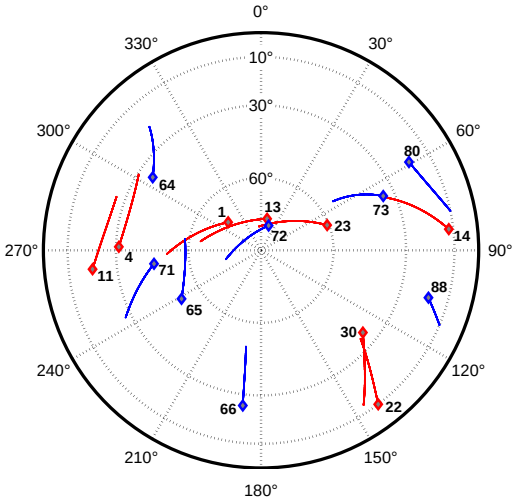



Figure 8.2: Skyplot of the combined GPS/Galileo scenario

The number of visible Galileo satellites varies over the simulation time between six and eight, while eight visible GPS satellites are constantly visible. Note that no obstructions

or multipath effects were simulated. Thus, the total number of visible satellites also varies between 14 and 16 (see Figure 8.3).

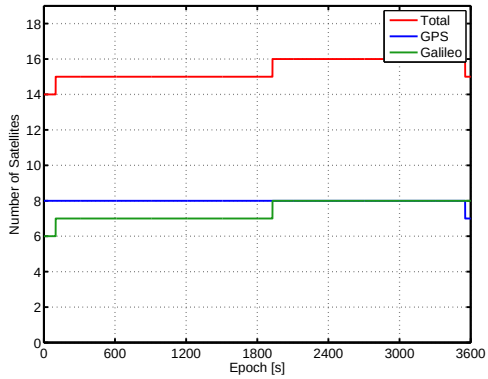



Figure 8.3: Number of satellites of a combined GPS/Galileo scenario

In the following subsection the results of different scenarios are presented. In a first step GPS and Galileo are treated separately using noisefree scenarios to verify the correct implementation of the GPS and Galileo processing chain within the PVT. Note that in this case noisefree means that no random errors and additional effects, beside the orbital errors and satellite clock errors, are modeled.

### 8.2.1 GPS solution

In a first step a GPS-only noisefree scenario is used to verify the implemented satellite position computation and receiver position adjustment algorithms. Figure 8.4 shows the coordinate differences $\Delta X_{\text{WGS-84}}$, $\Delta Y_{\text{WGS-84}}$, and $\Delta Z_{\text{WGS-84}}$ between the computed position solution and the reference coordinates.

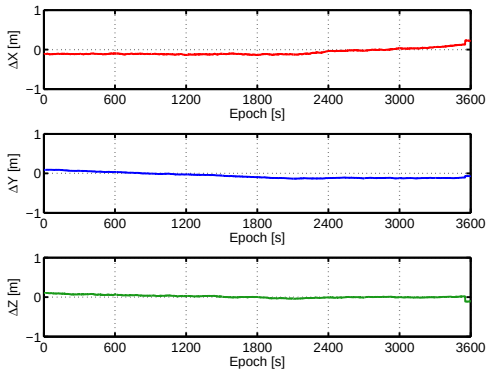



Figure 8.4: Coordinate differences of the GPS-only noisefree solution

The differences are within a few centimeters with respect to the reference position. The remaining errors are mainly due to unmodeled orbital and satellite clock errors within the ephemerides data. Table 8.2 summarizes the mean values of the computed coordinates, the

mean differences with respect to the reference coordinates, and the standard deviations of the differences.

Table 8.2: GPS-only noisefree solution

| | Mean value [m] | Mean difference [m] | Standard deviation [m] |
|---|---|---|---|
| $X_{\text{WGS-84}}$ | 4209209.62 | -0.07 | 0.19 |
| $Y_{\text{WGS-84}}$ | 1127854.28 | -0.06 | 0.07 |
| $Z_{\text{WGS-84}}$ | 4641772.12 | 0.02 | 0.10 |

Since the differences are close to zero the GPS satellite position computation, as well as the GPS satellite clock correction module and the adjustment function are implemented correctly. Also the weighting of the observations, reflected by the estimated position standard deviation, is verified.

### 8.2.2 Galileo solution

In case of Galileo almost the same algorithms as for GPS are used. The only difference is that in case of Galileo the GST is used instead of the GPS time as reference time. The coordinate differences with respect to the reference of a noisefree Galileo-only solution are depicted in Figure 8.5. The mean values, as well as the mean differences with respect to the reference position, and the standard deviations of the position solution are summarized in Table 8.3.

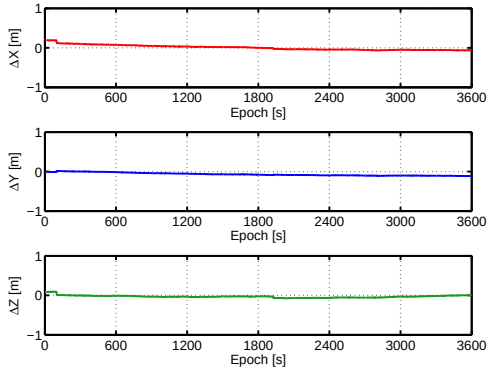



Figure 8.5: Coordinate differences of the Galileo-only noisefree scenario

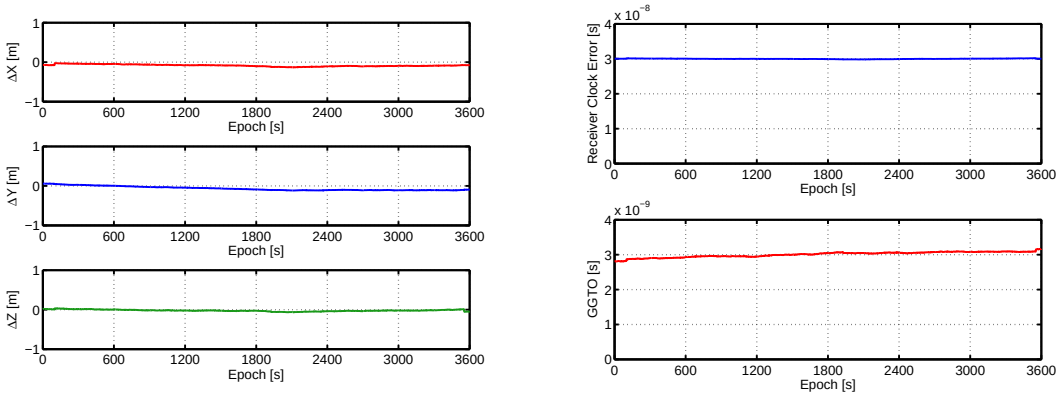
As in the case of the GPS-only solution the Galileo-only solution shows that in the noisefree case the position solution is within a few centimeters of the reference position.
The slightly higher standard deviations are due to the satellite geometry and the changing number of visible satellites. Nevertheless the results show again a correct implementation of the Galileo system within the PVT module. Note that no enhancement in the Galileo position accuracy can be expected since GPS and Galileo have the same service specifications.

Table 8.3: Galileo-only noisefree solution

| | Mean value [m] | Mean difference [m] | Standard deviation [m] |
|---|---|---|---|
| $X_{\text{WGS-84}}$ | 4209209.70 | 0.01 | 0.27 |
| $Y_{\text{WGS-84}}$ | 1127854.27 | -0.07 | 0.19 |
| $Z_{\text{WGS-84}}$ | 4641772.07 | -0.03 | 0.32 |

### 8.2.3 Combined GPS and Galileo solution

As discussed in Sect. 7.6.1 a combination of GPS and Galileo requires a different treatment of the individual systems. While the differences within the coordinate systems are neglected the different time reference systems have to be taken into account. Thus, the design matrix in the adjustment algorithm has to be changed according to (7.23). In this case, instead of estimating one single receiver clock error, one receiver clock error and the GGTO are estimated. The GGTO is estimated using the user level approach (Sect. 7.6.1) and as the common timing base for the receiver the GPS system time is used. Note that the usage of the GST would lead to identical results. A combined noisefree GPS/Galileo position solution is visualized in Figure 8.6a. Figure 8.6b shows the estimated receiver clock error (upper plot) and the GGTO estimation (lower plot).




(a) (b)

Figure 8.6: Position solution of a combined GPS and Galileo solution (a) and the estimated receiver clock error and the GGTO (b)

The differences between the mean position solution and the reference position and the standard deviations of the position solution are again in the centimeter level (Table 8.4).

Table 8.4: Coordinate, receiver clock, and GGTO differences of the combined GPS/Galileo noisefree single point solution

| | Mean value [m] | Mean difference [m] | Standard deviation [m] |
|---|---|---|---|
| $X_{\text{WGS-84}}$ | 4209209.61 | -0.08 | 0.14 |
| $Y_{\text{WGS-84}}$ | 1127854.27 | -0.06 | 0.08 |
| $Z_{\text{WGS-84}}$ | 4641772.08 | -0.02 | 0.13 |

| | Mean value [s] | Mean difference [s] | Standard deviation [s] |
|---|---|---|---|
| $\delta_R$ | $2.98 \cdot 10^{-8}$ | $-1.34 \cdot 10^{-10}$ | $1.07 \cdot 10^{-10}$ |
| GGTO | $3.01 \cdot 10^{-9}$ | $-9.30 \cdot 10^{-11}$ | $7.56 \cdot 10^{-11}$ |

When comparing different GNSS an important figure of merit are the DOP values. The DOP factor is measure of the instantaneous geometry of the visible satellites. The DOP values can be calculated from the inverse of the normal equation matrix. A low DOP value reflects a good geometry, while a high DOP value indicates a bad satellite dristibution. Several DOP values can be used for comparison. In the following the position dilution of precision (PDOP), geometric dilution of precision (GDOP), and time dilution of precision (TDOP) are used. The computation of the DOP values, as well as their meaning is described in Hofmann-Wellenhof et al. (2008). In Figure 8.7 a comparison of the DOP values of the GPS-only solution (Figure 8.7a), the Galileo-only solution (Figure 8.7b), and the combined solution (Figure 8.7c) can be found. The combined solution shows clearly that the geometry due to the increased number of satellites is better, compared to the single system solutions.
The jumps within the Galileo DOP values is due to the appearance and loss of satellites (Figure 8.3). A comparison of the DOP values of the individual GPS and Galileo solutions reveals that there are no big differences between them. After the verification of the PVT module, the overall receiver has been verified using digital IF signals generated by the GIPSIE®–IFS simulator. The same combined noisefree GPS and Galileo scenario has been used to generate the digital IF signal. Note that only 60 seconds of the digital IF signal have been generated for the evaluation. After a successful acquisition, the tracking has been performed using the algorithms described in Chap. 5. Out of the tracking results the navigation data have been extracted and the pseudoranges have been computed. Afterwards these data have been processed by the PVT to obtain the final position solution. Figure 8.8 shows the coordinate differences with respect to the WGS-84 reference coordinates for the GPS-only (Figure 8.8a) and the Galileo-only (Figure 8.8b) solution.

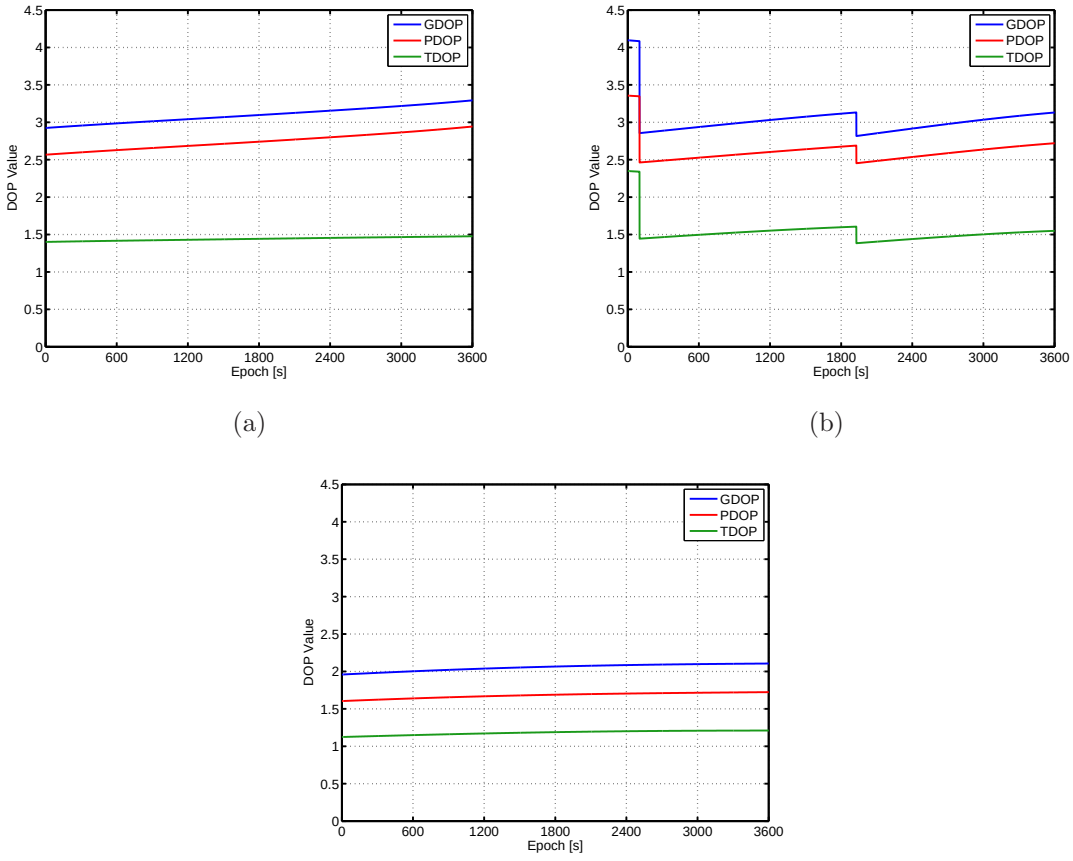



Figure 8.7: Comparison of DOP values for (a) GPS-only, (b) Galileo-only, and (c) combined GPS and Galileo simulation

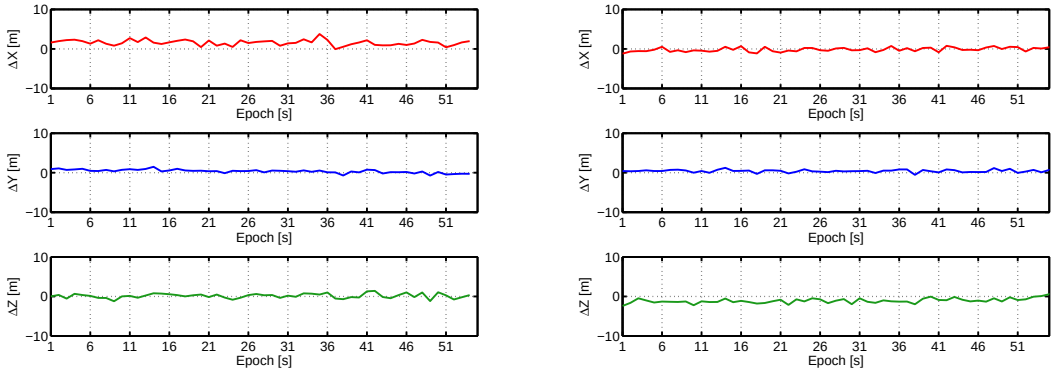



Figure 8.8: Coordinate differences with respect to the reference position for a GPS-only (a) and a Galileo-only (b) simulation using the GIPSIE®–IFS in case of the noisefree scenarios

The mean coordinate values, as well as the mean differences $\Delta$ and the standard deviations

$\sigma$ show deviations in the range of one meter (Table 8.5).

Table 8.5: Coordinate differences in meters of the GPS and the Galileo noisefree scenarios using the GIPSIE®–IFS

| | GPS | | | Galileo | | |
|---|---|---|---|---|---|---|
| | Mean value [m] | $\Delta$ [m] | $\sigma$ [m] | Mean value [m] | $\Delta$ [m] | $\sigma$ [m] |
| $X_{\text{WGS-84}}$ | 4209211.30 | 1.61 | 0.70 | 4209209.55 | -0.14 | 0.53 |
| $Y_{\text{WGS-84}}$ | 1127854.71 | 0.37 | 0.45 | 1127854.76 | 0.42 | 0.34 |
| $Z_{\text{WGS-84}}$ | 4641772.25 | 0.15 | 0.59 | 4641771.02 | -1.08 | 0.60 |

The higher differences compared to the previous simulations are due to the signal processing and the channel synchronization. As described in Sect. 6.4 the pseudorange computation is based on the file pointer which has a resolution of one sample. Thus, according to (6.3), the pseudorange resolution is within a few meters. To get better results it would be necessary to increase the resolution of the receiver timing base mentioned in Sect. 6.4. Figure 8.9 shows the position differences in case of Galileo when using the standard discriminator instead of the multiple gate delay discriminator. In this case the coordinate differences shows larger variations compared to using the MGD discriminator.

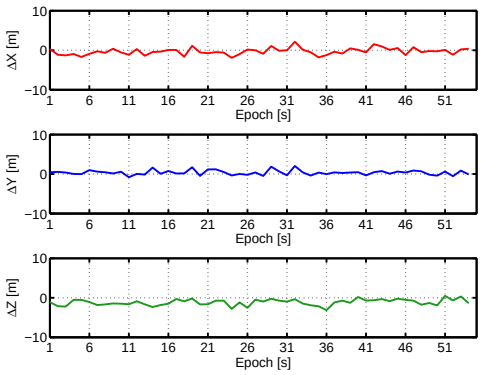



Figure 8.9: Coordinate differences of a Galileo-only scenario with respect to the reference position using standard discriminator instead of multiple gate array discriminator

The combined GPS/Galileo position solution is depicted in Figure 8.10.

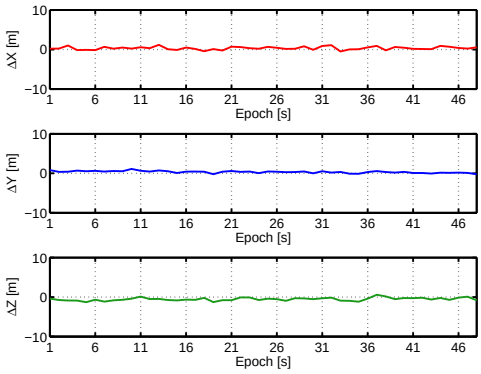


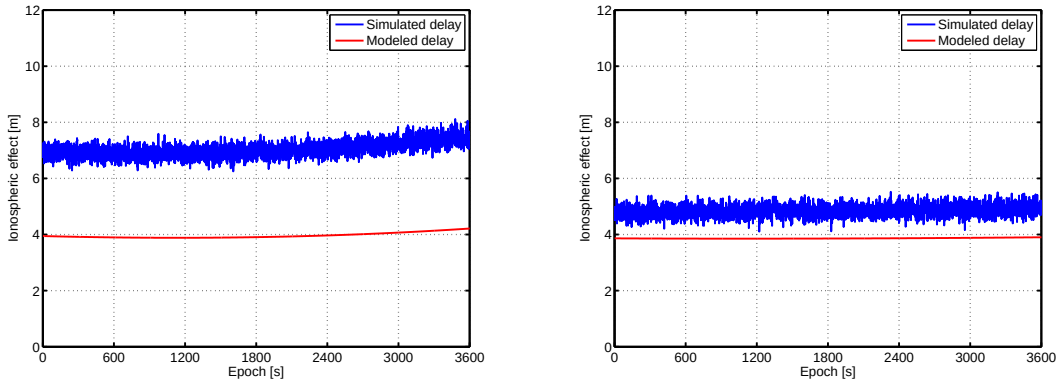

Figure 8.10: Coordinate differences of the combined GPS/Galileo noisefree solution using the GIPSIE®–IFS

Table 8.6 summarizes the combined GPS/Galileo position solution and the differences with respect to the reference coordinates, as well as the standard deviations.

Table 8.6: Coordinate differences of the combined GPS/Galileo noisefree scenarios using the GIPSIE®–IFS

| | Mean value [m] | Mean difference [m] | Standard deviation [m] |
|---|---|---|---|
| $X_{\text{WGS-84}}$ | 4209210.03 | 0.34 | 0.40 |
| $Y_{\text{WGS-84}}$ | 1127854.68 | 0.34 | 0.27 |
| $Z_{\text{WGS-84}}$ | 4641771.57 | -0.53 | 0.39 |

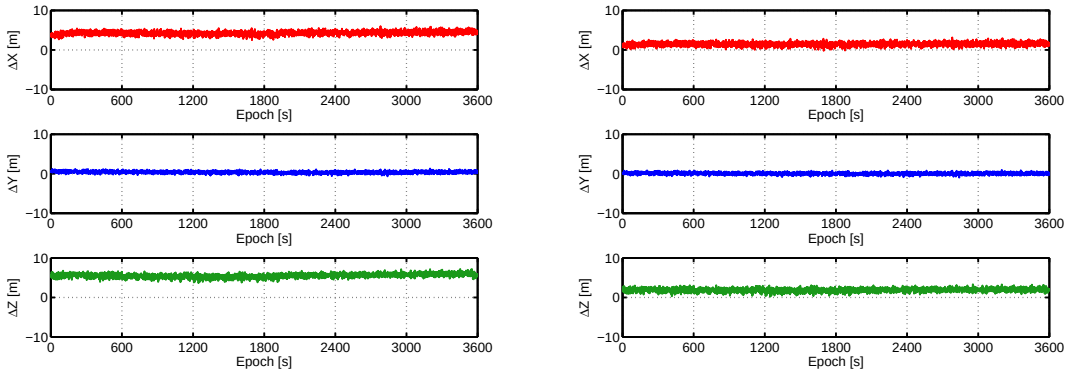
The combination of GPS and Galileo increases the stability of the solution. This is mainly due to the increased number of visible satellites. After verification of the basic receiver functionality, in a next step, the influence of the ionospheric and tropospheric effect as well as the implementation of the corresponding models have been tested. The GIPSIE® simulator uses a NeQuick model in combination with a noise model for simulating the effect of the ionosphere. The troposphere has been simulated using the ESA tropospheric model (Abart 2009). All simulations regarding the propagation effects have been performed using both, GPS and Galileo, systems. In a first approach only the ionospheric effect, including noise, has been simulated. For compensating the ionospheric effect in case of GPS the Klobuchar model has been used, while in case of Galileo the NeQuick model has been used. Figure 8.11 shows a comparison of the simulated ionospheric delays (blue) and the computed ionospheric corrections (red) in case of using the Klobuchar model (left) or the Nequick model (right).




(a) (b)

Figure 8.11: Comparison of simulated ionospheric delays and computed ionospheric correction for (a) GPS satellite using the Klobuchar model and for (b) Galileo satellite using the Nequick model

The differences between simulated and modeled delays of the Klobuchar and the NeQuick model reflect the quality of the models. The effect when applying ionospheric models is visualized in Figure 8.12. Figure 8.12a shows the coordinate differences in case of simulating an ionospheric delay but not modeling it. The differences are in the range of several meters. In contrast Figure 8.12b shows the same simulation in case of modeling the ionosphere. The remaining differences are due to the inaccuracies of the models and due to noise. Thus, it is necessary to compensate the ionospheric propagation delay by applying the corresponding models.




(a) (b)

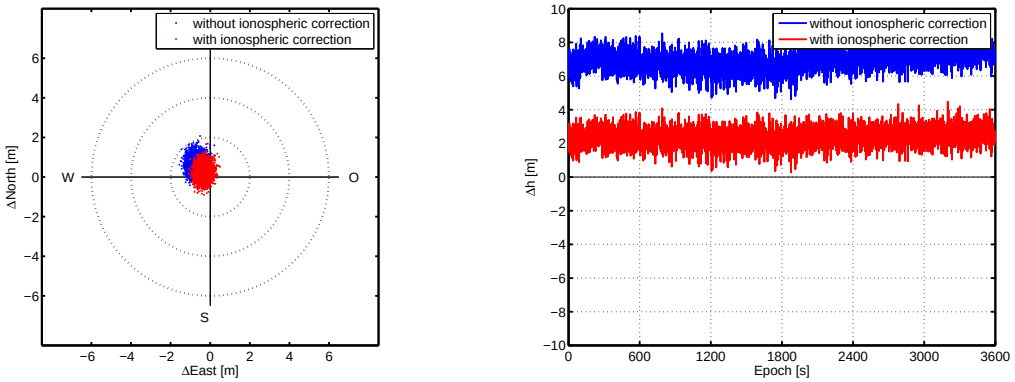
Figure 8.12: Coordinate differences of the combined GPS/Galileo scenario with respect to the reference position in case (a) no ionospheric corrections are applied and (b) if ionospheric corrections are used

Table 8.7 compares the influence of ionospheric modeling on the position solutions. Note that the standard deviations do not differ very much, but the coordinate differences are significant. The main impact of the ionosphere on the position solution usually can be

Table 8.7: Coordinate differences of the combined GPS/Galileo scenario in case of correcting and not correcting the ionospheric delay

| | No ionospheric correction | | | Ionospheric correction | | |
|---|---|---|---|---|---|---|
| | Mean value [m] | $\Delta$ [m] | $\sigma$ [m] | Mean value [m] | $\Delta$ [m] | $\sigma$ [m] |
| $X_{\text{WGS-84}}$ | 4209213.97 | 4.28 | 0.63 | 4209211.16 | 1.47 | 0.50 |
| $Y_{\text{WGS-84}}$ | 1127854.72 | 0.38 | 0.37 | 1127854.39 | 0.05 | 0.29 |
| $Z_{\text{WGS-84}}$ | 4641777.63 | 5.53 | 0.57 | 4641773.99 | 1.88 | 0.45 |

observed in the height component. Therefore the WGS-84 coordinates are projected into a horizontal system using the universal transverse Mercator (UTM) projection. The formulas for this coordinate transformation are provided in Hofmann-Wellenhof et al. (2008). The left side of Figure 8.13 shows the coordinate differences in north $\Delta$N and east $\Delta$E with respect to the reference solution in UTM coordinates. Whereas the right side shows the differences in height for the case when no ionospheric correction is applied (blue) and if the ionosphere is modeled (red).

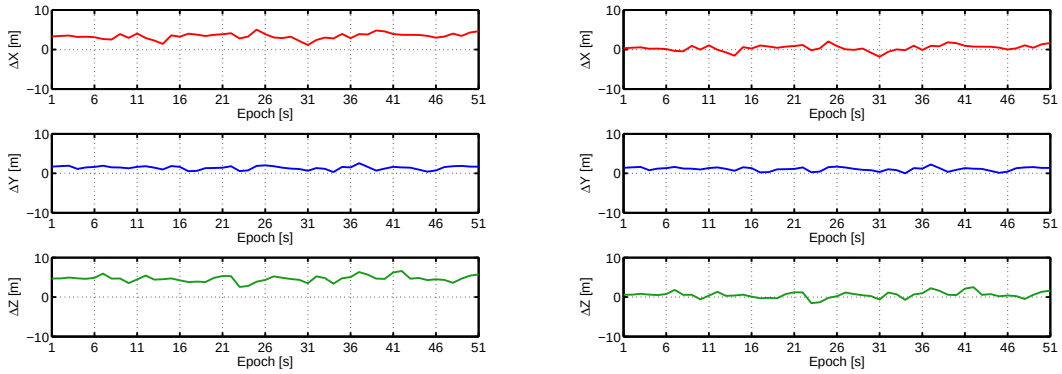



(a) Horizontal differences

(b) Height differences

Figure 8.13: UTM coordinate differences with respect to the reference position in case (blue) no ionospheric correction is applied and (red) if ionospheric corrections are used

The differences in north and east components are much lower than in the height component. Thus, the main influence of the ionosphere on the position solution appears in the height component. The remaining offset is again due to the inaccuracies of the models. The simulations showed that in case of higher solar activities also the remaining errors increase. Ionospheric corrections are thus necessary to obtain a more accurate position solution. The same behavior can be observed when looking at the position solutions computed out of the digital IF signals. Again the same simulation was used to generate the digital IF signal, which then was processed by the software-based receiver. Figure 8.14a shows the WGS-84 coordinate differences in case of no ionospheric compensation. Again the influence of the ionosphere is noticeable especially in the $X$ and $Z$ component. When applying ionospheric corrections these offsets can be reduced significantly (Figure 8.14b).




(a) (b)

Figure 8.14: GPS/Galileo solution using GIPSIE®–IFSdata - no ionospheric correction (a) and ionospheric correction (b)

The mean coordinates, as well as the coordinate differences and the standard deviations are summarized in Table 8.8.

Table 8.8: Coordinate differences in case of not applying and applying ionospheric corrections; GPS/Galileo solution using GIPSIE®–IFS data

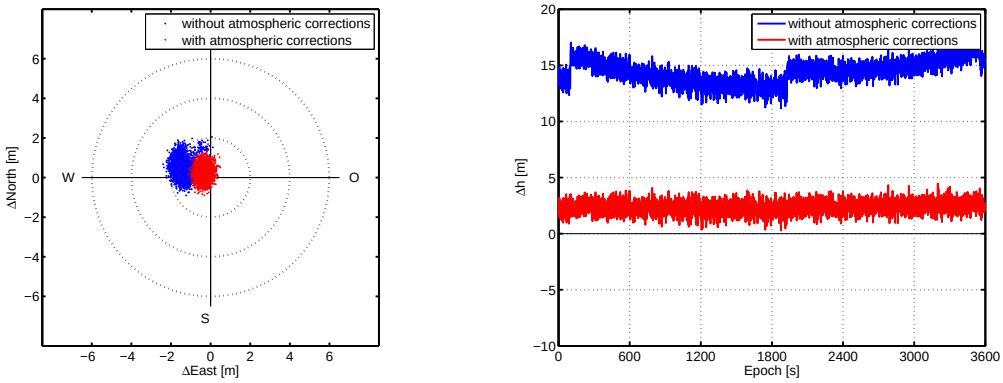
| | No ionospheric correction | | | Ionospheric correction | | |
|---|---|---|---|---|---|---|
| | Mean value [m] | $\Delta$ [m] | $\sigma$ [m] | Mean value [m] | $\Delta$ [m] | $\sigma$ [m] |
| $X_{\text{WGS-84}}$ | 4209213.09 | 3.40 | 0.76 | 4209210.10 | 0.41 | 0.76 |
| $Y_{\text{WGS-84}}$ | 1127855.72 | 1.39 | 0.47 | 1127855.42 | 1.08 | 0.47 |
| $Z_{\text{WGS-84}}$ | 4641776.77 | 4.67 | 0.80 | 4641772.66 | 0.56 | 0.81 |

Thus, by applying ionospheric corrections to the pseudorange measurements an improvement in the position accuracy in the range of several meters can be achieved. When using broadcast parameters the receiver has to wait until the necessary subframe or page is received. Until this happens either no corrections or default model parameters can be used. The PVT module uses the ionospheric model decision tree (Figure 7.4) for deciding which ionospheric correction should be applied. Note that the PVT is also capable of applying ionospheric corrections obtained from EGNOS or global and local ionospheric TEC maps (Huber et al. 2011).

The ionosphere is not the only source of error within the signal propagation, but also the troposphere influences the position solution. Modeling the troposphere is not an easy task, since the difficulty arises from modeling the water vapor. Ground measurements often do not reflect the situation along the signal path. Several models have been published in literature. While the GIPSIE® simulator uses the ESA reference model, within the receiver the modified Hopfield model is used.

To validate the implementation of the tropospheric model, both the ionospheric and the tropospheric effect have been simulated in a next step. The position solutions obtained from the PVT using raw data are shown in Figure 8.15. Again for a better understanding

UTM coordinate differences are visualized. Figure 8.15a shows a scatter plot, while Figure 8.15b shows a comparison of the height components in case of no corrections are applied (blue) and if tropospheric and ionospheric effects are modeled (red).




(a) Horizontal differences (b) Height differences

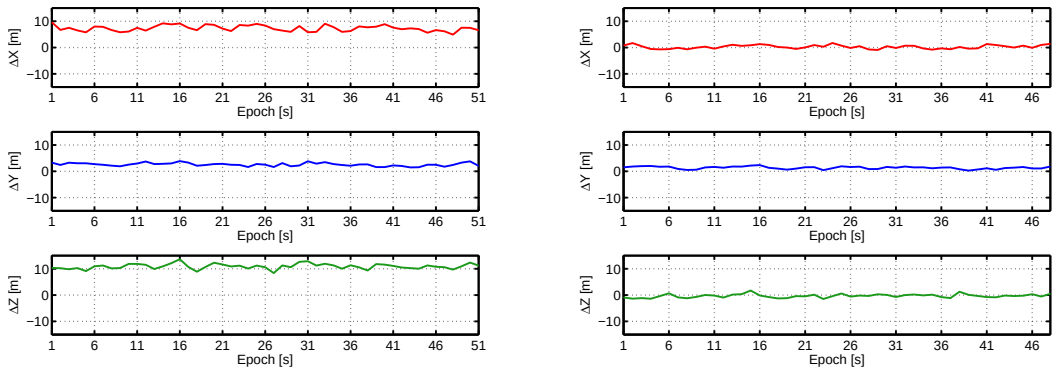
Figure 8.15: UTM coordinate differences with respect to the reference position in case no ionospheric and tropospheric corrections (not even blind models) are applied (blue) and if both ionospheric and tropospheric corrections are used (red)

Like the ionosphere, the troposphere shows a larger influence on the height component. When applying the ionospheric and tropospheric corrections to the measurements an enhancement in the accuracy is visible again. The corresponding mean coordinate differences and the standard deviations are compared in Table 8.9.

Table 8.9: Coordinate differences when modeling ionospheric and tropospheric effects; GPS/Galileo solution using GIPSIE® data as PVT input information

| | Without corrections | | | With corrections | | |
|---|---|---|---|---|---|---|
| | Mean value [m] | $\Delta$ [m] | $\sigma$ [m] | Mean value [m] | $\Delta$ [m] | $\sigma$ [m] |
| $X_{\text{WGS-84}}$ | 4209219.19 | 9.50 | 1.35 | 4209211.16 | 1.47 | 0.50 |
| $Y_{\text{WGS-84}}$ | 1127855.43 | 1.09 | 0.78 | 1127854.39 | 0.05 | 0.29 |
| $Z_{\text{WGS-84}}$ | 4641782.98 | 10.88 | 1.21 | 4641773.98 | 1.89 | 0.45 |

The jumps in height are due to the changing satellite geometry (sudden appearance of a satellite). The remaining errors, in case of no models are used, are in the range of up to ten meters. When correcting the measurements the position offset is significantly reduced to the meter level. The differences in case of modeling both error sources is larger than in the scenario before (ionospheric errors only). This is due to the imperfectness (differences between reality and models) of the two models. In case computing the position solution, based on the GIPSIE®–IFS digital IF output using the same simulation parameters, a similar behavior can be observed. Figure 8.16a shows the WGS-84 coordinate differences in the case when no corrections are applied, while Figure 8.16b shows the impact when applying the correction terms.




(a) (b)

Figure 8.16: Comparison of (a) unmodeled propagation effects and (b) applying tropospheric and ionospheric models using the GIPSIE®–IFS

Table 8.10 compares the results of the two simulations. Without corrections the mean coordinate differences are in the range of ten meters for the $X$ and the $Z$ component. When correcting the pseudorange measurements a significant improvement of the position accuracy can be obtained.

Table 8.10: Coordinate differences when applying or not applying tropospheric and ionospheric models; GPS/Galileo solution using GIPSIE®–IFS data

| | Without corrections | | | With corrections | | |
|---|---|---|---|---|---|---|
| | Mean value [m] | $\Delta$ [m] | $\sigma$ [m] | Mean value [m] | $\Delta$ [m] | $\sigma$ [m] |
| $X_{\text{WGS-84}}$ | 4209217.06 | 7.37 | 1.12 | 4209209.95 | 0.26 | 0.68 |
| $Y_{\text{WGS-84}}$ | 1127856.91 | 2.57 | 0.61 | 1127855.67 | 1.33 | 0.49 |
| $Z_{\text{WGS-84}}$ | 4641783.02 | 10.91 | 1.01 | 4641771.81 | -0.28 | 0.68 |

The implemented algorithms and models for single-frequency observations, as well as the combination of GPS and Galileo are thus verified. In case of simulating propagation effects, the simulations show that a higher accuracy can be achieved when applying ionospheric and tropospheric corrections. The combination of GPS and Galileo requires the estimation of the GGTO for computing a combined position solution, while the differences in the coordinate frames can be neglected.

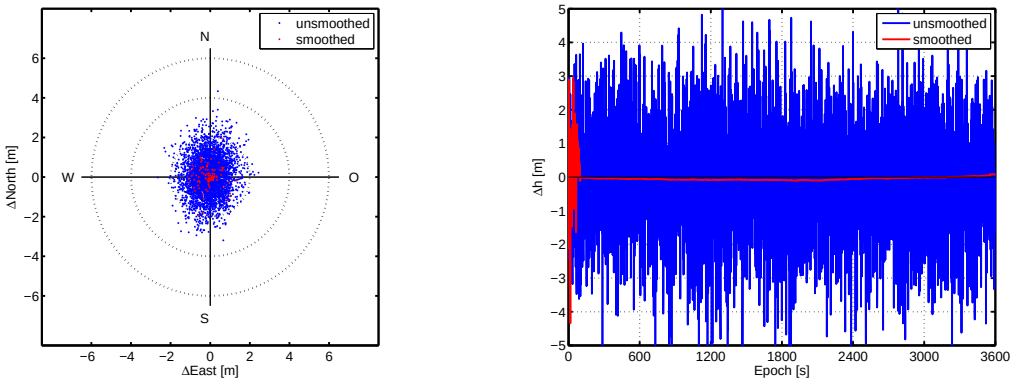
### 8.2.4 Dual-frequency solution

As shown in the previous section, the propagation effects are a main error source. Although models are used to mitigate the errors, not all effects can be eliminated. Since the ionosphere is a dispersive medium (Chap 7) dual-frequency observations can be used to eliminate the ionospheric errors. The corresponding mathematical models have been derived in Sect. 7.3.4. Since GIPSIE® in its current development stage is not able to simulate dual-frequency observations, an additional module was implemented to simulate

dual-frequency observations. This additional module allows the simulation of pseudoranges on two different carrier frequencies. Therefore the simulated ionospheric delay on L1 is multiplied by the factor

$$\frac{f_{L_1}^2}{f_{L_2}^2}. \tag{8.1}$$

Note that it would not be accurate to simulate the same measurement noise in the two resulting pseudoranges. Thus, a new noise signal was simulated for the second pseudorange. The resulting measurements have been exported in the RINEX format for further processing. Carrier phase measurements have been simulated and stored together with the pseudorange measurements in the RINEX file. The previously used single-frequency scenario containing ionospheric delays has been used to generate the presented dual-frequency observation scenario. As already mentioned in Sect. 7.3.2 the linear combination of the pseudoranges increases the noise level by a factor of 3. The blue solution in Figure 8.17 shows the UTM coordinate differences in case of eliminating the ionospheric delay without smoothing. Although the mean coordinate differences in north and east component are close to zero, the individual position solutions show large variations (Figure 8.17a). The height component (Figure 8.17b) varies three times more than compared to the corresponding single-frequency solution (Figure 8.13). The standard deviations in Table 8.11 reflect this effect. Thus, countermeasures must be taken to reduce the noise. One possible method is to implement a pseudorange smoothing by means of phase observations. When applying the smoothing algorithm described in Sect. 7.5 the noise within the position solution is reduced. Since the weighted smoothing algorithm is used the algorithm needs approximately 100 epochs until the smoothing converges (red graphs in Figure 8.17).




(a) Horizontal differences

(b) Height differences

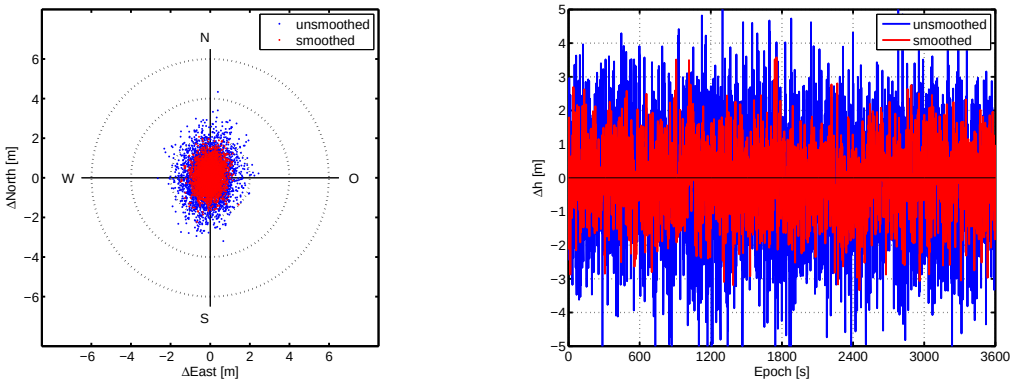
Figure 8.17: UTM coordinate differences with respect to the reference position in case of using dual-frequency measurements without smoothing (blue) and with smoothing (red)

The coordinate differences in both cases are close to zero. The smoothing reduces the standard deviations (Table 8.11). The resulting position solution has both a high accuracy and a low standard deviation.

Table 8.11: Coordinate differences using unsmoothed and smoothed dual-frequency measurements

| | Unsmoothed | | | Smoothed | | |
|---|---|---|---|---|---|---|
| | Mean value [m] | $\Delta$ [m] | $\sigma$ [m] | Mean value [m] | $\Delta$ [m] | $\sigma$ [m] |
| $X_{\text{WGS-84}}$ | 4209209.58 | -0.11 | 3.17 | 4209209.76 | 0.07 | 0.65 |
| $Y_{\text{WGS-84}}$ | 1127854.26 | -0.08 | 1.83 | 1127854.33 | -0.01 | 0.38 |
| $Z_{\text{WGS-84}}$ | 4641772.05 | -0.05 | 2.84 | 4641771.99 | -0.12 | 0.58 |

Figure 8.18 shows the position solutions of the same scenario but instead of the weighted smoothing algorithm the unweighted smoothing algorithm is used (red) (Sect. 7.5). Compared to the unsmoothed solution (blue) an improvement is visible but the noise is not reduced as much as in the case of the weighted algorithm.

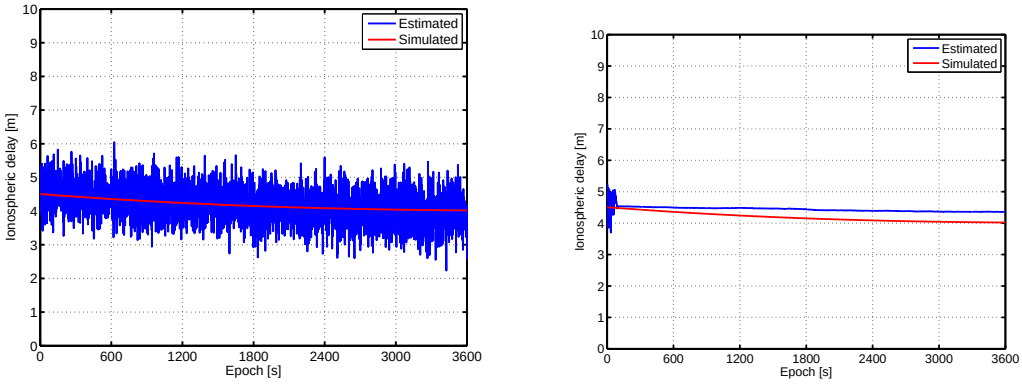



(a) Horizontal differences

(b) Height differences

Figure 8.18: UTM coordinate differences with respect to the reference position in case of using dual-frequency without smoothing (blue) and with unweighted smoothing (red)

Note that the advantage of the dual-frequency approach is the elimination of the ionospheric effect, which results in a better position accuracy compared to the single-frequency approach. As described in Sect. 7.3.4 the PVT estimates the ionospheric delay based on dual-frequency observations. This approach provides the same results as the elimination algorithm, but has the advantage that the information about the ionospheric delay can be used for weighting the observations. Figure 8.19 shows a comparison of the estimated and simulated ionospheric delay in case of unsmoothed observations (Figure 8.19a) and smoothed observations (Figure 8.19b).
In both cases the estimated ionospheric delays match the simulated ones. The estimated delay, in case of using unsmoothed observations, shows higher variations compared to the smoothed one. For the weighting of the observations the smoothed approach is preferred, since the estimated delay and, thus, the weight does not vary very much between two consecutive epochs.
The simulations showed that the developed software-based receiver is able to process single-frequency observations and digital IF signals from GPS and Galileo. The implemented




(a) (b)

Figure 8.19: Comparison of the simulated and estimated ionospheric delay (in vertical direction) using dual-frequency observations with and without smoothing algorithms

algorithms for satellite position computation, receiver position adjustment, weighting, and ionospheric and tropospheric models are verified using simulated data. Further the simulations showed that a combination of GPS and Galileo does not improve significantly the accuracy of the position solution but positively influences the geometry (DOP values) and leads to a more stable solution. The use of dual-frequency observations enables a higher position accuracy but introduces additional noise. Thus, a pseudorange smoothing is applied to reduce the noise.

## 8.3 Position solutions with real-world data

After verification of the implemented algorithms and models, the receiver has been tested using recorded real-world data. For testing with real data two different groups of data sets have been used. The first group of data sets was recorded by a Javad Sigma receiver, capable of receiving GPS L1, L2, and L5 signals. The receiver provided the raw observation data (pseudoranges and phases, as well as ephemerides data) in RINEX format. The second group represents real data recorded using the FhG-IIS multiband front-end (Sect. 3.3). As of June 1st, 2012, there were in total 9 GPS satellites in orbit transmitting the L2C signal. The L2C signals are healthy although the C/NAV message contains only dummy data. Different tests showed a maximum number of L2C observations of five out of twelve visible satellites. This is sufficient for forming a linear combination and a position computation. For a critical investigation on the performance of the linear combinations the P2 (Precise Code on L2) has been used instead. All real data sets were recorded at the roof of the Geodesy building in Graz Steyrergasse 30 using a geodetic pillar. The evaluation of the algorithms' performance was done by comparing the calculated coordinates with the coordinates of the pillar (reference coordinates). The measurements using the Javad Sigma receiver took place on August 13th, 2010, starting at 11:50. The WGS-84 reference coordinates are given in Table 8.12.

Table 8.12: Reference position in Cartesian ($X$, $Y$, $Z$) coordinates determined by geodetic survey

| Reference coordinates | |
|---|---|
| $X_{\text{WGS-84}}$ | 4195390.368 m |
| $Y_{\text{WGS-84}}$ | 1159799.911 m |
| $Z_{\text{WGS-84}}$ | 4646943.418 m |

### 8.3.1 Pseudorange and phase observations

In a first step the data recorded by the Javad receiver are evaluated. Figure 8.20 shows a skyplot (azimuth and elevation angle) of the recorded scenario, where in total 6 GPS satellites were visible. The geometric distribution shows, that due to a building near by, no satellites appear in the south.

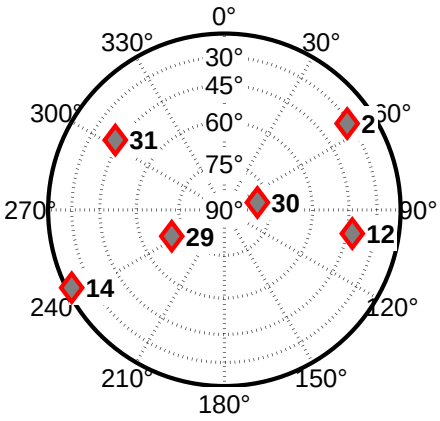
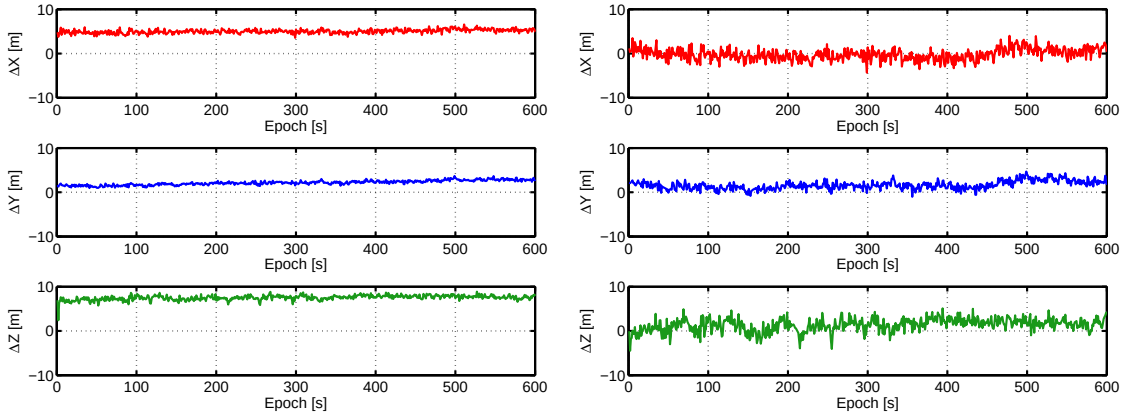



Figure 8.20: Skyplot of the real data scenario

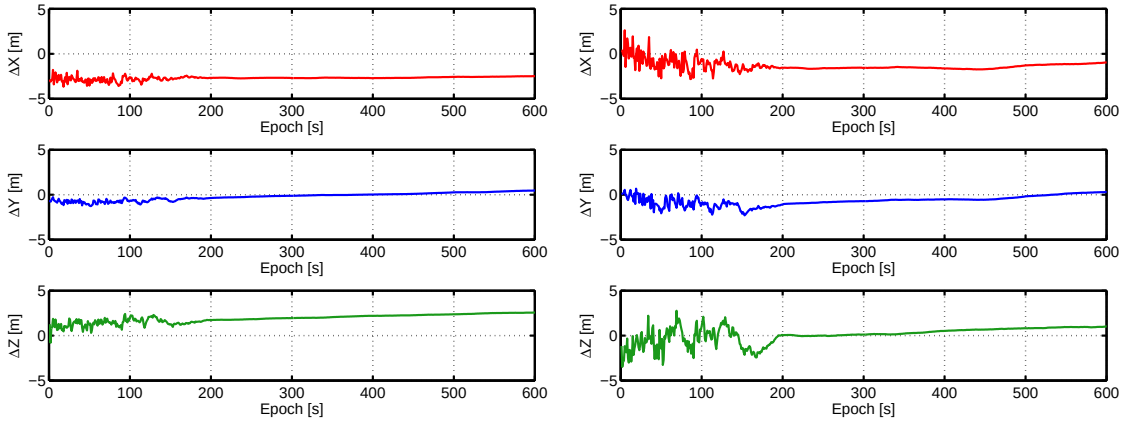
Figure 8.21a shows the WGS-84 coordinate differences between the reference coordinates and the calculated coordinates when no compensation of the troposphere and ionosphere is done. The coordinate differences in the $X$ and $Z$ directions are in the range of several meters. The large deviations are mainly due to the neglected ionospheric and tropospheric effects. Figure 8.21b shows the corresponding dual-frequency solution. The deviations from the reference position are much lower compared to the single-frequency solution, but the noise is increased. The remaining errors in the dual-frequency solution are due to the unmodeled tropospheric errors.




(a) (b)

Figure 8.21: Comparison of single-frequency (no atmospheric corrections) (a) and dual-frequency (ionosphere-free combination) solution (b) using observations from the Javad receiver

It takes several epochs for the smoothing algorithm to compensate the noise. In a next step the tropospheric Hopfield model is applied to the phase-smoothed pseudoranges for both cases. In case of single-frequency observations the ionospheric corrections, obtained by the Klobuchar model, are applied in addition. Figure 8.22a shows the coordinate differences with respect to the reference position of the single frequency solution, while Figure 8.22b shows the solution of the dual-frequency observation case.




(a) (b)

Figure 8.22: Comparison of single-frequency (a) and dual-frequency (b) solution when applying ionospheric and tropospheric corrections using observations of the Javad receiver

The height offset in case of the single-frequency solution is caused by the global ionospheric model, which obviously overcompensates the ionospheric delay. The dual-frequency solution also shows an offset. As described in Department of Defense (2011) and Tetewsky et al. (2009) delays between different signal components within the signal generation occur. Thus, the civil dual-frequency user shall correct for the group delay and ionospheric effects by applying a modified linear combination using inter-signal correction (ISC) parameters.

The ISC parameters will be available through the C/NAV message as message type 30 data. As mentioned in Sect. 6.3.1, currently only message type 0 (dummy message) is broadcasted and thus no ISC parameters are available. Additional information regarding the need and the usage of inter-signal corrections can be found in Tetewsky et al. (2009). Instead of using the ISC parameters, which were not available, differential correction biass (DCBs), obtained from CODE (Centre for orbit determination in Europe) were used for the next step of computation. Figure 8.23 shows the benefit of using DCB values within the dual-frequency solution.

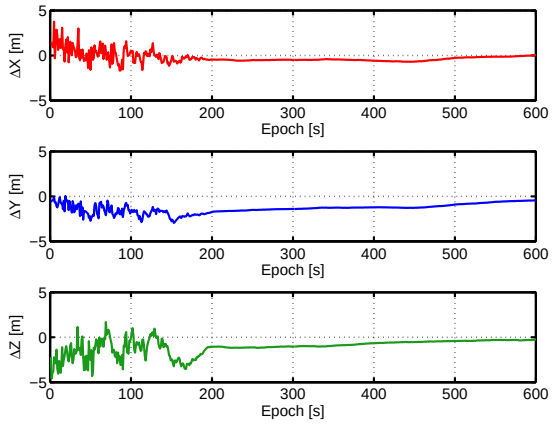



Figure 8.23: Coordinate differences when using DCB values using observations of the Javad receiver

### 8.3.2 Digital intermediate frequency signals

Different recordings of digital IF signals have been performed and evaluated. Most of the data sets contained only three satellites providing dual-frequency observations. The presented data set has been recorded on November 23rd, 2011, at 8:35. In total 60 seconds of data were recorded. After acquisition and tracking 7 satellites were visible in total. Out of those seven satellites it was possible to track four satellites on L1 C/A and L2C. Figure 8.24 shows the visible satellites. Satellites which provided dual-frequency observations are marked in red, while the others, delivering single-frequency pseudoranges only, are visualized in blue.

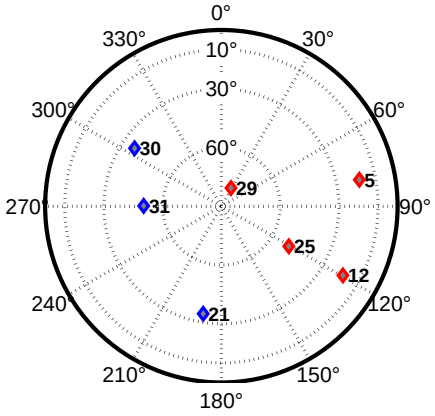
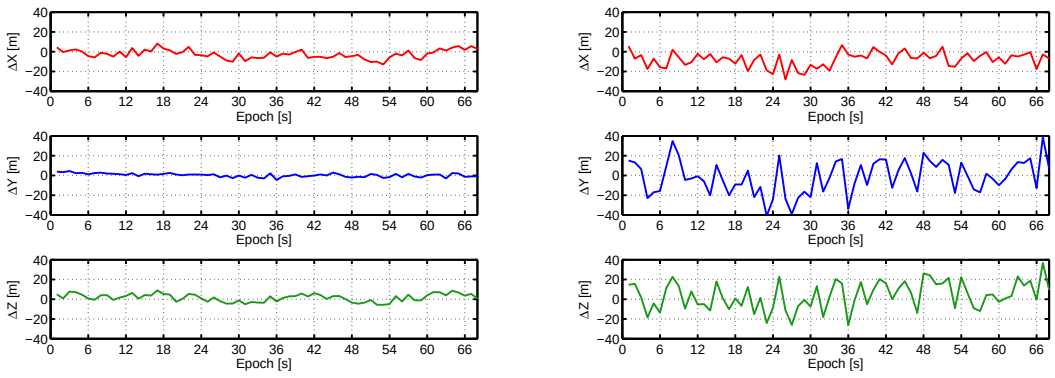



Figure 8.24: Skyplot of visible satellites delivering single frequency (blue) and dual-frequency (red) observations

### 8.3.3 Single-frequency GPS solution

In a first step the individual L1 C/A and L2C solutions are computed. In case of L1 C/A it was possible to measure pseudoranges for all seven satellites. The coordinate differences for the single-frequency solution are depicted in Figure 8.25a. In case of L2C observations only four satellites could be used. In both cases ionospheric and tropospheric corrections have been applied to the measurements. In case of L2C measurements the computed Klobuchar corrections are multiplied by the factor from (8.1) and applied to the measurements. The WGS-84 coordinate differences are shown in Figure 8.25b.




(a) (b)

Figure 8.25: Coordinate differences of the L1 C/A (a) and L2C (b) single-frequency solutions using the FhG-IIS front-end IF data

Although in case of L2C the mean values are off by few meters, the computed standard deviation exceeds 15 meters. This is due to the measurement principle. The main problem in this case is that there are only four visible GPS satellites providing L2C measurements

and they are poorly distributed over the sky (Figure 8.24), resulting in a degraded positioning performance. Additionally, as discussed previously, the file pointer index is used for measurement synchronization. In case of L2C the resolution is not sufficient thus the measurements get very noisy. To solve this problem an advanced counter would be needed. The statistical parameters for the two solutions are provided in Table 8.13.

Table 8.13: Coordinate differences of the L1 C/A and L2C single-frequency solutions using the FhG-IIS front-end IF data

| | L1 | | | L2C | | |
|---|---|---|---|---|---|---|
| | Mean value [m] | $\Delta$ [m] | $\sigma$ [m] | Mean value [m] | $\Delta$ [m] | $\sigma$ [m] |
| $X_{\text{WGS-84}}$ | 4195388.00 | -2.36 | 4.48 | 4195382.60 | -7.76 | 7.45 |
| $Y_{\text{WGS-84}}$ | 1159800.24 | -0.08 | 1.87 | 1159798.56 | -1.35 | 17.01 |
| $Z_{\text{WGS-84}}$ | 4646945.02 | -0.05 | 3.91 | 4646947.23 | 3.81 | 14.14 |

The results of the L1 C/A single-frequency solution are shown in Figure 8.26, where the computed positions are overlaid using a keyhole markup language (KML) file in Google$^{\text{TM}}$Earth.

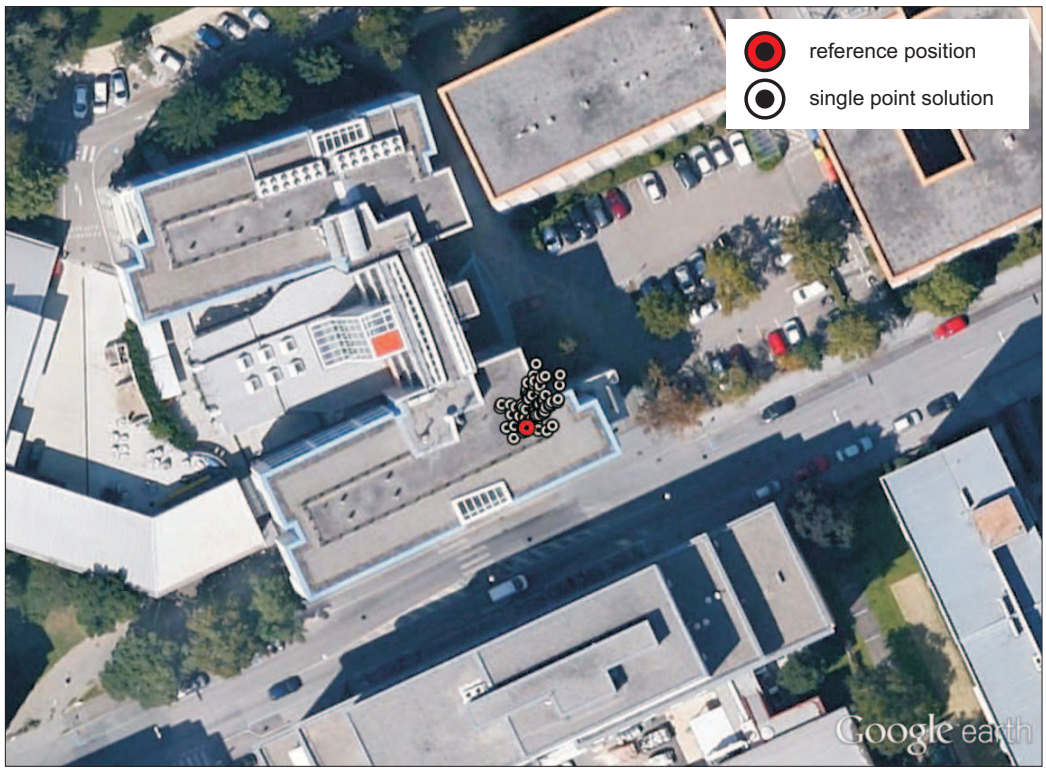



Figure 8.26: L1 C/A position solution overlaid with Google$^{\text{TM}}$Earth

The L2C-only position solution overlaid in Google$^{\text{TM}}$Earth is shown in Figure 8.27.


Figure 8.27: L2C position solution overlaid with Google$^{TM}$Earth

The L2C position solutions show larger offsets with respect to the reference position than the L1 C/A solutions. This is mainly due to the higher noise within the measurements and the bad satellite geometry. All four visible L2C satellites are close together in the sky resulting in high DOP values. The high noise within the measurements is introduced at the tracking and pseudorange computation stages. The problem is that for a common timing source the file pointer position is used. The tracking is performed using the L2CM-code neglecting the L2CL chips. This also reduces the tracking accuracy and results in a coarse resolution within the file pointer position. To enhance the tracking it is necessary to improve the resolution by either tracking the complete L2C-code (L2CM and L2CL) or introducing another timing reference by counting the exact number of processed samples.

### 8.3.4 Dual-frequency GPS solution

The dual-frequency solution is obtained by taking all available measurements into account. In case of available dual-frequency measurements the ionospheric term is eliminated using the pseudorange linear combination and applying a tropospheric model correction. In case of single-frequency observations the ionospheric and the tropospheric corrections are applied to the pseudorange measurements. DCB values are applied to the measurements but due to the high measurement noise the benefit of applying the DCB values is not visible. The final dual-frequency solution is shown in Figure 8.28.

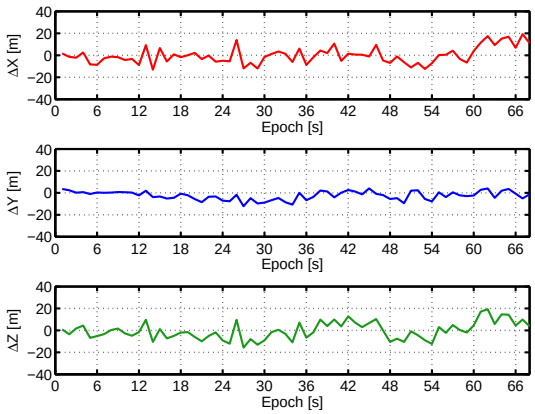
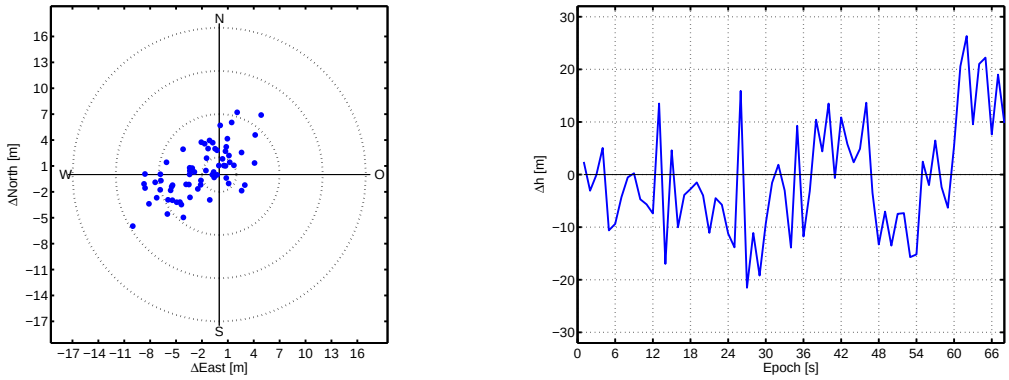



Figure 8.28: Coordinate differences of the dual-frequency solutions using the FhG-IIS front-end IF data

The dual-frequency solution, compared to the single-frequency solution, shows no benefit. The available number of dual-frequency observations, their geometric distribution, and the noisy measurements are the limiting factor in this case. Figure 8.29 shows the UTM coordinate differences in the horizontal components, as well in the height component of the dual-frequency position solutions. Although strong variations in the height component are visible due to the noise, the mean value of the height component is almost zero.




(a) Horizontal differences (b) Height differences

Figure 8.29: UTM coordinate differences of the dual-frequency solutions using the FhG-IIS front-end IF data

Table 8.14 shows the obtained coordinates, the mean coordinate values, and the standard deviations of the computed dual-frequency position solution.
The before mentioned amplification of noise due to the linear combination is clearly visible. The amplification factor, compared to the single-frequency solution, is about 3. The worse result is mainly due to the large measurement noise within the computed pseudoranges. As already mentioned, a code smoothing by means of phase observations is necessary in order to reduce the measurements noise. Since no phase observations are available the

Table 8.14: Coordinate differences of the dual-frequency solutions using the FhG-IIS front-end IF data

| | Mean value [m] | Difference [m] | Standard deviation [m] |
|---|---|---|---|
| $X_{\text{WGS-84}}$ | 4195390.24 | -0.13 | 7.57 |
| $Y_{\text{WGS-84}}$ | 1159797.52 | -2.38 | 3.94 |
| $Z_{\text{WGS-84}}$ | 4646943.11 | -0.30 | 7.87 |

smoothing cannot be performed. Figure 8.30 shows the dual-frequency solution overlaid using a KML file in Google$^{\text{TM}}$Earth.

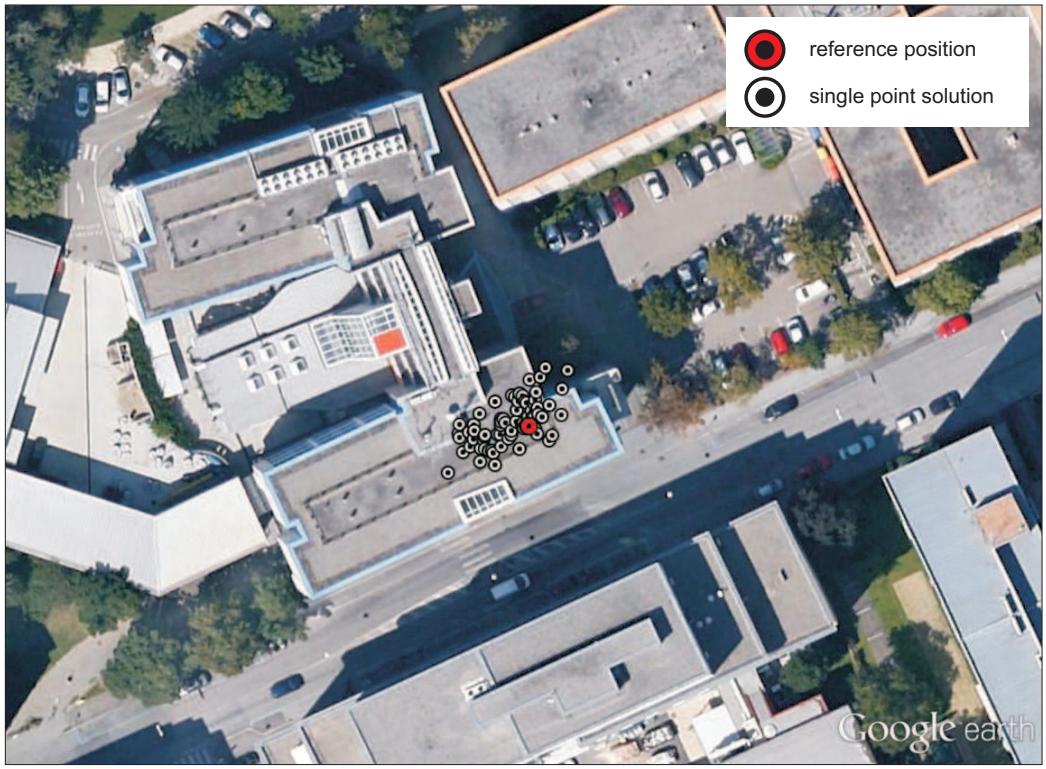



Figure 8.30: Dual-frequency position solution overlaid with Google$^{\text{TM}}$Earth

Thus, several steps are necessary in order to improve the accuracy. The first one is the implementation of enhanced tracking loops and loop filters, which are able to reduce the measurement noise. The second one is the investigation of increasing the used sampling rate in order to obtain more accurate pseudorange measurements.

## 8.4 Execution time

The developed software-based receiver is capable to compute position, velocity, and time using multi-frequency and multi-system signals. To this end, it applies intelligent weighting strategies and dual-frequency mitigation strategies to achieve higher accuracies. The

receiver is written in C++ however it is not designed or optimized to run in real-time. A standard PC (Intel Core2Duo, 3.16 GHz, 2GB RAM) was used for the computations. The execution time was measured several times and the mean values (rounded to minutes) have been taken for comparison. As mentioned in Borre et al. (2007) the execution time of the acquisition module depends on the implemented acquisition algorithm. The implemented parallel code phase algorithm uses the fastest Fourier transform in the west (FFTW) library (Frigo and Johnson 2005). No other external software libraries have been used. Table 8.15 summarizes the execution time for the different modules. Note that the processing time of the tracking depends on the number of visible satellites. The given values of the tracking module represent one single satellite and 60 seconds of digital IF data.

Table 8.15: Measured execution time [minutes]

| Signal | Acquisition | Tracking | PVT | Total |
|---|---|---|---|---|
| GPS L1 C/A | 1 | 20 | 1 | 22 |
| GPS L2 L2C | 2 | 30 | 1 | 33 |
| Galileo E1B | 4 | 24 | 1 | 25 |

The bottleneck and therefore the limiting factor regarding real-time is the amount of data which has to be processed within the acquisition and tracking modules. Tracking is the most time consuming task within the processing chain and depends on the code length and the modulation. Within the tracking module especially the correlation computation of the input signal with the replica codes is very time-consuming. To speed up the computation time different strategies can be applied. One possibility would be the implementation of the digital signal processing in an FPGA board. Another possible strategy is to use the graphics board of a PC in order to enhance the computation time. Figure 8.31 shows the reduction of data starting from the IF data file and resulting in a NMEA output file. Note that the figure is not drawn to scale.

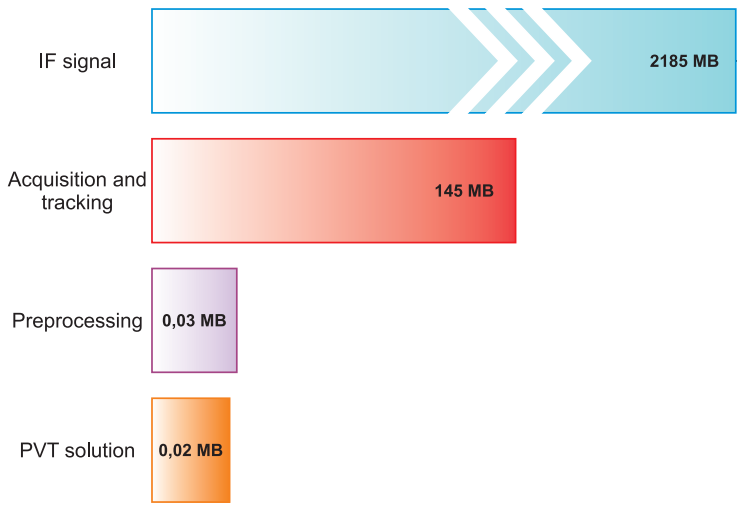



Figure 8.31: Comparison of the amount of data which is processed by each receiver module

In this case 60 seconds of digital IF signal have been assumed. After the tracking the necessary data is reduced to the correlation values and other important loop and processing parameters. After the preprocessing the amount of data is reduced further resulting in RINEX files having a size of a few kilobytes. The PVT solution in this case produces NMEA output which contains the position solution as well as quality parameters and other information. Thus, by using the developed software-based receiver, out of several – for the end-user useless – gigabytes of data a few kilobytes of valuable information can be gained.

# Chapter 9

# Conclusions and outlook

This thesis focuses on the development and verification of a dual-frequency software-based global navigation satellite system receiver, capable of processing GPS C/A L1, GPS L2C, and Galileo E1B signals.
In the last decades the term global navigation satellite system expanded continuously. Nowadays GNSS represents a couple of systems (i.e., GPS, Galileo, GLONASS, Beidou/Compass) which are utilizing different signals. Also the GNSS receiver market expands and in this context also software-based receivers become more and more important due to cost optimization and miniaturization. Thus, in a first step, the actual software-based receiver market is studied and analyzed. The study shows that up to now there are several solutions for software-based receivers on the market, but at the time of starting this thesis a software-based receiver was just an upcoming research tool. After providing a rough overview on the existing global navigation satellite systems, tailored to GPS and Galileo, a brief introduction to signal processing basics is given and the civil and open-service signal structures are highlighted. The civil and open service signals, their properties and composition are presented. Afterwards the system architecture of the developed software-based receiver is presented and a short introduction to the GNSS front-end module and the resulting raw sampled data characteristics is provided. The basic components of the receiver and their interrelation within the developed receiver are discussed.
In the second part, the implemented digital signal processing (i.e., acquisition and tracking) is described in detail. After giving some basics on signal acquisition the implementation of parallel code phase search acquisition method is presented. The rough estimation of the signal properties, e.g., Doppler frequency and code phase, is explained. The necessity of a zero-padding strategy for the acquisition of several civil signals is examined and demonstrated. The acquisition topic concludes with different acquisition results from different systems and signals showing the performance and accuracy of the implemented acquisition module. To this end, simulated data as well as real-world data are used. The successful acquisition of GIOVE-A and GIOVE-B is presented as well. At the beginning of the chapter dealing with the code and the carrier tracking a brief summary on tracking loops and navigation data demodulation is provided. The mathematical basics for the carrier and the code tracking are derived and the important tracking loop parameters are discussed. Then different strategies as well as discriminator functions for tracking GPS and Galileo signals are described in more detail. In case of BOC modulated signal tracking the problem of tracking the wrong correlation peak is considered and possible solutions are described. Thereby, the implemented multiple gate discriminator technique is investigated in more detail. The chapter concludes with some analysis regarding the performance of the implemented tracking loops which is based on simulated data as well as on real-world

data. Additionally the effect of using different loop parameters (e.g., noise bandwidth and damping ratio) is shown. The results of the multiple gate discriminator technique as well as the results of the dual-frequency tracking are discussed. The computation of the raw observables (i.e., pseudoranges) and the recovery of the ephemerides data, which are based on the tracking results, is presented in a next step. In this context the topics CRC, FEC, and block de-interleaving which are necessary for the navigation data extraction are reviewed. The navigation data recovery is explained and the successful decoding of the different navigation message formats is presented. The pseudorange computation method as well as the achievable accuracy are explained in more detail. The pseudorange computation is verified using both data from an IF signal generator and real-world data. The accuracy of dual-frequency pseudorange measurements are compared as well.
In the third part the positioning algorithm for a combined GPS and Galileo receiver capable of processing multi-frequency measurements is presented. In this point, especially the different system time and their influence on the position computation and possible solutions are discussed. The advantage of using more than one systems lies in the increase of availability. In the near future it will be possible to receive more than 30 satellites simultaneously instead of, e.g., 6 today. This enhances the availability and the integrity of the position computation, especially in urban canyons. The availability of new signals enables an improvement of the position accuracy by eliminating the ionospheric effect. The treatment of dual-frequency measurements for reducing the ionospheric error is presented. Several methods based on forming pseudorange linear combinations are used. The different approaches are compared and the issue of noise amplification in case of using linear combinations is examined. For reducing the measurement noise a smoothing algorithm has to be used and a scheme for weighting the different observations is presented. At the end of this chapter a detailed description of the implemented PVT module is given.
For testing and evaluating the performance of the receiver developed, several tests and analysis are presented in the last part. In a first step, the software-based receiver is tested using the GIPSIE® constellation and performance simulator. Thereby, the obtained results are compared with the simulated parameters in order to validate the implementation. Different test results utilizing different system and signal combinations are presented and discussed. In a next step, real-world data collected by the Fraunhofer multiband front-end are used to evaluate the performance under realistic conditions. Results of dual-frequency observations are presented and the need and benefit of carrier smoothing is pointed out. Also the problem of currently missing inter-signal correction parameters is discussed. A real-world dual-frequency GPS solution, containing L1 C/A and L2C measurements, of the software-based receiver, is presented. A summary on the processing time and on the amount of data used within the receiver concludes the thesis.
The current implementation is designed for post-processing. The bottleneck and thus the limiting factor is the amount of data which has to be processed within the acquisition and tracking modules. Especially the correlation computation of the input signal with the replica codes is time-consuming. Other software based receiver implementations use field programmable gate array boards in order to achieve real-time capability. It is possible to implement the designed tracking loops and especially the correlators in hardware. Another possible strategy is to use the graphics board of a PC in order to swap out the correlation computation. Graphics cards are designed especially for handling large amount of data

and performing correlations. Several software libraries exist which can handle the data exchange between processor and graphics card. The disadvantage is that not all types of graphic cards are supported. In order to have platform independent solution this option was not taken into account in this thesis.
The current implementation is ideal to be used for developing new algorithms for GNSS receivers, like the weighting strategy for ionospheric correction, or the optimization of loop parameters. It provides the necessary flexibility to implement and to test new algorithms, either in the signal processing part or in the PVT section. Especially when developing a tight or even an ultra tight coupling with autonomous sensors, a software-based approach safes time and costs. The tight integration, for example, requires a combined processing of IF signal and sensor output within a, e.g., Kalman Filter and a feedback to the tracking loops. Further for monitoring purposes (e.g., monitoring land slides), when real-time capability is not a limiting factor, a software based receiver can be used. Also the application of the developed receiver within a geodetic monitoring station network can be envisaged. Perhaps the most promising field of application is the use of the presented receiver for signal authentication. In the near future the implementation of satellite based augmentation systems as well as additional GNSS signal components (e.g., Galileo PRS service) is intended to further increase the accuracy and the integrity. Also the use of multi-frequency measurements for the mitigation of multipath and interference effects are currently analyzed within a TeleConsult Austria's research project, using software-based receivers. The exponential increasing number of global navigation satellite systems promise a bright future for navigation applications especially with enhanced performance requirements. In this context multi-frequency software-based GNSS receivers are a powerful tool to explore new algorithms and investigate new ideas.

# List of Figures

# List of Tables

# References


Abart C (2009): Development of a GNSS IF signal generator. PhD dissertation, Institute of Navigation and Satellite Geodesy, Graz University of Technology, Austria.

Akopian D, Valio H, Turunen S (2002): Fine frequency resolution acquisition methods for GPS receivers. In: Proceedings of ION GPS 2002, 15th International Technical Meeting of the Satellite Division of the Institute of Navigation, Portland, Oregon, September 24–27: 2515–2523.

Akos DM, Normark PL, Lee JT, Gromov KG, Tsui JBY, Schamus J (2000): Low power Global Navigation Satellite System (GNSS) signal detection and processing. In: Proceedings of ION GPS 2000, 13th International Technical Meeting of the Satellite Division of the Institute of Navigation, Salt Lake City, Utah, September 19–22: 784–791.

Anderson D, Fuller-Rowell T (1999): Space environment topics: The ionosphere. SE-14, Space Environment Center, Boulder, Colorado, Available at www.sec.noaa.gov/Education.

Arbesser-Rastburg B, Cerdeira RP (2005): Galileo ionospheric model for single frequency receivers. European Space Agency, ESA-APPNGSPEC/00344-BAR, Noordwijk, The Netherlands.

Arbesser-Rastburg B (2006): The Galileo single frequency ionospheric correction algorithm. Paper presented at the Third European Space Weather Week, Brussels, Belgium, November 13–17.

Arbesser-Rastburg B, Jakowsi N (2007): Effects on satellite navigation. In: Bothmer V, Daglis IA (eds): Space weather – physics and effects. Springer, Berlin, Heidelberg, New York: 383–402.

Bartenev V, Kosenko V, Chebotarev V (2006): Builders notes: Russian GLONASS at the stage of active implementation. Inside GNSS, 4(4): 40–43.

Bello PA, Fante RL (2005): Code tracking performance for novel unambiguous M-code time discriminators. In: Proceedings of The National Technical Meeting of the Institute of Navigation, San Diego, California, January 24–26: 293–298.

Berglez P (2008): Development of a GPS, Galileo and SBAS receiver. In: Proceedings of Electronics in Marine (ELMAR), 50th International Symposium, Zadar, Croatia, September 10–13: 573–576.

Berglez P, Wasle E, Seybold J, Hofmann-Wellenhof B (2009): GNSS constellation and performance simulator for testing and certification. In: Proceedings of ION GNSS 2009, 22nd International Technical Meeting of the Satellite Division of the Institute of Navigation, Savannah, Georgia, September 22–25: 2220–2228.

Berglez P (2010): SoftGNSS 2 – A dual-frequency software-based GNSS receiver – final report, version 1.0. Technical Report, Austrian Research Promotion Agency, Austrian Space Applications Programme, contract number 819682.

Berglez P, Seybold J, Geiger B, Soudan M, Vogel C, Abart C, Singh A, Hofmann-Wellenhof B (2010): Development of a dual-frequency software-based GNSS receiver. In: Proceedings of ION GNSS 2010, 23rd International Technical Meeting of the Satellite Division of the Institute of Navigation, Portland, Oregon, September 21–24: 1967–1974.

Bertelsen N, Borre K, Rinder P (2004): The GPS code software receiver at Aalborg University. In: Proceedings of the 2nd ESA Workshop on Satellite Navigation, NAVITEC 2004, ESTEC, Noordwijk, The Netherlands, December 8–10: 373–380.

Blunt P (2007): Advanced Global Navigation Satellite System receiver design. PhD dissertation, Surrey Space Centre, University of Surrey, England.

Borio D (2008): A statistical theory for GNSS signal acquisition. PhD dissertation, Dipartimento di Elettronica, Politecnico di Torino, Italy.

Borre K, Akos D, Bertelsen N, Rinder P, Jensen S (2007): A software-defined GPS and Galileo receiver, a single-frequency approach – applied and numerical harmonic analysis. Birkhäuser, Boston Basel Berlin.

Buracchini E (2000): The software radio concept. IEEE Communication Magazine, 38(9): 138–143.

Canalda Pedrós R (2009): Galileo signal generation - simulation analysis. Master thesis, Department of Computer and Electronic Engineering, University of Limerick, Ireland.

Charkhandeh S (2007): X86-based real time L1 GPS software receiver. Master thesis, Department of Geomatics Engineering, University of Calgary, Canada.

Chung BY, Chien C, Samueli H, Jain R (1993): Performance analysis of an all-digital BPSK direct-sequence spread-spectrum IF receiver architecture. IEEE Journal on Selected Areas in Communications, 11(7): 1096–1107.

Cojocaru S, Birsan E, Batrinca G, Arsenie P (2009): GPS-GLONASS-GALILEO: a dynamical comparison. The Journal of Navigation, 62(1): 135–150.

Dempster A (2006): Correlators for L2C – some considerations. Inside GNSS, 1(7): 32–37.

Department of Defense (2008): Federal Register Notice Vol. 73, No. 185 (Ref. 31), September 23.

Department of Defense (2011): Global Positioning System standard positioning service performance standard, 6th edition. Available from the US Assistant for GPS, Positioning and Navigation, Defense Pentagon, Washington DC.

Dierendonck AJ van (1996): GPS receivers. In: Parkinson BW, Spilker JJ (eds): Global Positioning System: theory and applications. American Institute of Aeronautics and Astronautics, Washington DC, vol 1: 329–406.

Diggelen F van (2009): A-GPS: assisted GPS, GNSS, and SBAS. Artech House, Norwood.

European Commission (2010): Galileo open service. Signal in space interface control document (OS SIS ICD). Issue 1.1, September. Available at www.ec.europa.eu/enterprise/policies/satnav/galileo/.

European Commission (2011): Report from the Commission to the European Parliament and the Council: Mid-term review of the European satellite radio navigation programmes. COM(2011) 5, January. Available at www.ec.europa.eu/enterprise/newsroom/cf/_getdocument.cfm?doc_id=6321.

European Space Agency (2008): GIOVE-A + B navigation signal-in-space interface control document. Issue 1.1, August 8. Available at www.giove.esa.int.

European Space Agency (2011): One Soyuz launcher, two Galileo satellites, three successes for Europe. Press release, October 21. Available at www.esa.int/esaCP/.

Falcone M, Erhard P, Hein GW (2006): Galileo. In: Kaplan ED, Hegarty CJ (eds): Understanding GPS: principles and applications, 2nd edition. Artech House, Norwood: 559–594.

Fantino M, Molino A, Nicola A (2009): N-Gene GNSS receiver: benefits of software radio in navigation. In: Proceedings of the European Navigation Conference GNSS-2009, Napoli, Italy, May 3–6.

Farah AMA (2008): Comparison of GPS/Galileo single frequency ionospheric models with vertical TEC maps. Artificial Satellites, 43(2): 75–90.

Fraunhofer (2010): Fraunhofer IIS multiband GNSS frontend manual. Power Efficient Systems Department at Fraunhofer Institute for Integrated Circuits, Nuremberg, Germany.

Frigo M, Johnson SG (2005): The design and implementation of FFTW3. Proceedings of the IEEE, 93(2): 216–231.

Gardner F (2005): Phaselock techniques, 3rd edition. John Wiley & Sons. New Jersey.

Geiger B, Soudan M, Vogel C (2010): On the detection probability of parallel code phase search algorithms in GPS receivers. In: Proceedings of PIMRC 2010, IEEE 21st International Symposium on Personal, Indoor and Mobile Radio Communications, Istanbul, Turkey, September 6–29: 865–870.

Geiger B, Vogel C, Soudan M (2012): A comparison between ratio detection and threshold comparison for GNSS applications. IEEE Transactions on Aerospace and Electronic Systems, vol 48: 1772–1779.

Gendt G, Altamimi Z, Dach R, Söhne W, Springer T (2010): GGSP: realisation and maintenance of the Galileo terrestrial reference frame. Advances in Space Research, 47(2): 174–185.

Giordanengo G (2009): Impact of notch filtering on tracking loops for GNSS applications. Master thesis, Facoltà di Ingegneria dell'informazione, Politecnico di Torino, Italy.

Gleason S, Gebre-Egziabher D (2009): GNSS applications and methods. GNSS Technology and Applications, Artech House, Norwood.

Haykin S, Veen BV (2005): Signals and systems, Interactive Solutions Edition. John Wiley & Sons. Chichester, England.

Hein GW, Avila-Rodríguez JA, Wallner S, Betz JW, Hegarty CJ, Rushanan JJ, Kraay AL, Pratt AR, Lenahan S, Owen J, Issler JL, Stansell TA (2006): MBOC: The new optimized spreading modulation. Recommended for Galileo L1 OS and GPS L1C. Inside GNSS, 1(4): 57–66.

Hochegger G, Nava B, Radicella SM, Leitinger R (2000): A family of ionospheric models for different uses. Physics and Chemistry of the Earth, Part C: Solar, Terrestrial and Planetary Science, 25(4): 307–310.

Hofmann-Wellenhof B, Lichtenegger H, Collins J (2001): GPS – theory and practice, 5th edition. Springer, Wien New York.

Hofmann-Wellenhof B, Lichtenegger H, Wasle E (2008): GNSS – global navigation satellite systems: GPS, GLONASS, Galileo, and more. Springer, Wien New York.

Hopfield HS (1969): Two-quartic tropospheric refractivity profile for correcting satellite data. Journal of Geophysical Research, 74(18): 4487–4499.

Huber K, Berglez P, Hofmann-Wellenhof B, Weber R, Troger M (2011): The development of enhanced algorithms for rapid precise point positioning. In: Austrian Contributions to the XXV General Assembly of the International Union of Geodesy and Geophysics (IUGG), VGI - Österreichische Zeitschrift für Vermessung und Geoinformation, reviewed paper: 114–121.

Hurskainen H, Lohan ES, Hu X, Raasakka J, Nurmi J (2008): Multiple gate delay tracking structures for GNSS signals and their evaluation with Simulink, SystemC, and VHDL. International Journal of Navigation and Observation: 1–17.

Ipatov V (2005): Spread spectrum and CDMA: principles and applications. John Wiley & Sons, Chichester, England.

Jan SS, Sun CC (2010): Signal existence verification (SEV) for GPS low received power signal detection using the time-frequency approach. Sensors, 10(5): 4717–4738.

Jock CRI, Parkinson BW, Enge PK (1996): The effect of the ionosphere and C/A frequency on GPS signal shape: considerations for GNSS2. In: Proceedings ION GPS 1996, 9th International Technical Meeting of the Satellite Division of the Institute of Navigation, Kansas City, Missouri, September 17–20: 647–653.

Julien O (2005): Design of Galileo L1F receiver tracking loops. PhD dissertation, Department of Geomatics Engineering, University of Calgary, Canada.

Jung J (2004): Implementation of correlation power peak ratio based signal detection model. In: Proceedings of ION GNSS 2004, 17th International Technical Meeting of the Satellite Division of the Institute of Navigation, Long Beach, California, September 21–24: 486–490.

Kaplan ED (2006): Introduction. In: Kaplan ED, Hegarty CJ (eds): Understanding GPS: principles and applications, 2nd edition. Artech House, Norwood.

Kay SM (1993): Fundamentals of statistical signal processing: Estimation theory. Vol. 1. Prentice-Hall, New Jersey.

Klobuchar J (1986): Design and characteristics of the GPS ionospheric time-delay algorithm for single-frequency GPS users. In: Proceedings of PLANS'86 – Position Location and Navigation Symposium, Las Vegas, Nevada, November 4–7: 280–286.

Krueger E, Schueler T, Hein G, Martellucci A, Blarzino G (2004): Galileo tropospheric correction approaches developed within GSTB-V1. In: Proceedings of the European Navigation Conference GNSS 2004, Rotterdam, The Netherlands, May 17 – 19.

Krumvieda K, Madhani P, Cloman C, Olson E, Thomas J, Axelrad P, Kober W (2001): A complete IF software GPS receiver: a tutorial about the details. In: Proceedings of ION GPS 2001, 14th International Technical Meeting of the Satellite Division of the Institute of Navigation, Salt Lake City, Utah, September 11–14: 789–829.

Lachapelle G, Cannon ME, O'Keefe K, Alves P (2002): How will Galileo Improve Positioning Performance? GPS World, 13(9): 38–48.

Lachapelle G, Petovello M, Fenton P, McCarthy DD, Swann J (2006): Reference Systems, UTC Leap Second, and L2C Receivers? Inside GNSS, 1(1): 20–24.

Langley R (2009): Expert advice: cause identified for pseudorange error from new GPS satellite SVN-49. GPS World, 20(8): 8–12.

Leitinger R, Radicella SM (2002): NeQuick ionospheric model - software documentation.

Lou HL (1995): Implementing the Viterbi algorithm. IEEE Signal Processing Magazine, 12(5): 42–52.

Macchi F (2010): Development and testing of an L1 combined GPS-Galileo software receiver. PhD dissertation, Department of Geomatics Engineering, University of Calgary, Canada.

McDonald KD (2002): The modernization of GPS: plans, new capabilities and future relationship to Galileo. Journal of Global Positioning System, 1(1): 1–17.

Memarzadeh Y (2009): Ionospheric modelling for precise GNSS applications. PhD dissertation, Department of Remote Sensing, Delft University of Technology, The Netherlands.

Misra P, Enge P (2006): Global Positioning System: signals, measurements, and performance, 2nd edition. Ganga-Jamuna, Lincoln.

Mitola J (1995): The software radio architecture. IEEE Communication Magazine, 33(5): 26–38.

Mongrédien C (2008): GPS L5 software receiver development for high-accuracy applications. PhD dissertation, Department of Geomatics Engineering, University of Calgary, Canada.

Mongrédien C, Rügamer A, Overbeck M, Rohmer G, Berglez P, Wasle E (2011): Opportunities and challenges for multi-constellation, multi-frequency automotive GNSS receiver. In: Heuberger A, Elst G, Randolf H (eds): Microelectronic Systems – circuits, systems and applications, Springer, Heidelberg Dordrecht London New York: 157–170.

Morton J (2007): Expert Advice: Software Defines Future. GPS World, 18(8): 12.

Muthuraman K, Shanmugam SK, Lachapelle G (2007): Evaluation of data/pilot tracking algorithms for GPS L2C signals using software receiver. In: Proceedings of ION GNSS 2007, 20th International Technical Meeting of the Satellite Division of the Institute of Navigation, Fort Worth, Texas, September 25–28: 2499–2509.

Muthuraman K (2010): Tracking techniques for GNSS data/pilot signals. PhD dissertation, Department of Geomatics Engineering, University of Calgary, Canada.

O'Keefe K (2001): Availability and reliability advantages of GPS/Galileo integration. In: Proceedings of ION GPS 2001, 14th International Technical Meeting of the Satellite Division of the Institute of Navigation, Salt Lake City, Utah, September 11–14: 2096–2104.

O'Keefe K, Ryan S, Lachapelle G (2002): Global availability and reliability assessment of the GPS and Galileo Global Navigation Satellite Systems. In: Canadian Aeronautics and Space Journal, 48(2): 123–132.

Oppenheim AV, Schafer RW, Buck JR (1999): Discrete-time signal processing, 2nd edition. Prentice Hall, London.

Overbeck M, Rohmer G, Wasle E, Berglez P, Seybold J, Euler HJ, Kahmann A (2010): Galileo receiver for mass market applications in the automotive area. In: Proceedings of the 5th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing, NAVITEC, ESTEC, Noordwijk, The Netherlands, December 8–10: 1–8.

Pajala E (2005): Code-frequency acquisition algorithms for BOC modulated CDMA signals with applications in Galileo and GPS systems. Master thesis, Department of Communications Engineering, Tampere University of Technology, Tampere, Finland.

Pany T (2010): Navigation signal processing for GNSS software receivers. GNSS Technology and Applications, Artech House, Norwood.

Parkinson B, Spilker J (1996): Global positioning system: theory and applications. American Institute of Aeronautics and Astronautics, Washington DC, vol. 1.

Petovello MG, Lachapelle G (2008): Centimeter-level positioning using an efficient new baseband mixing and despreading method for software GNSS receivers. EURASIP J. Adv. Signal Process, 2008, doi:10.1155/2008/194276.

Piriz R, Cueto M, Fernandez V, Tavella P, Sesia I, Cerretto G, Hahn J (2006): GPS/GALILEO interoperability: GGTO, timing biases, and GIOVE-A experience. In: Proceedings of the 38th Annual Precise Time and Time Interval Systems and Application Meeting (PTTI), Washington DC, December 7–9: 49–68.

Press WH, Teukolsky SA, Vetterling WT, Flannery BP (2007): Numerical recipes: the art of scientific computing, 3rd edition. Cambridge University Press, New York.

Radicella SM, Leitinger R (2001): The evolution of the DGR approach to model electron density profiles. Advances in Space Research, 27(1): 35–40.

Rohmer G (2011): Galileo receiver for mass market applications in the automotive area (GAMMA-A) – final report, version 1.0. European Space Agency. Seventh EU Framework Programme, contract number 228339. Technical Report, Available at www.gamma-project.info.

Rügamer A, Köhler S, Rohmer G, Berglez P, Wasle E, Seybold J (2010): GNSS constellation simulator and digital signal generator for receiver test, qualification and certification. In: Proceedings of CERGAL 2010, International Symposium on Certification of GNSS Systems and Services, Warnemünde, Germany, April 28–29: 1–7.

Saastamoinen J (1973): Contribution to the theory of atmospheric refraction. Bulletin Géodésique, 107: 13–34.

Samad MF (2009): Effects of MBOC Modulation on GNSS Acquisition Stage. Master thesis, Department of Communications Engineering, Tampere University of Technology, Tampere, Finland.

Samper JM, Pérez RB, Berenguer R, Lagunilla JM, Meléndez J (2008): GPS & Galileo: dual RF front-end receiver and design, fabrication, and test. Communication Engineering, McGraw-Hill, New York.

SiGe Semiconductor (2011): SiGe Semiconductor SE4120L GNSS Receiver IC Manual. Woburn, Massachusetts.

Skournetou D, Lohan ES (2007): Non-coherent multiple correlator delay structures and their tracking performance for Galileo signals. In: Proceedings of the European Navigation Conference GNSS 2007, Geneva, Switzerland, May 29 – June 1: 247–258.

Spilker J (1996): Tropospheric effects on GPS. In: Parkinson BW, Spilker JJ (eds): Global Positioning System: theory and applications. American Institute of Aeronautics and Astronautics, Washington DC, vol. 1: 517-546.

TeleConsult Austria GmbH (2010): GNSS multisystem performance simulation environment Manual. Version 1.1. Graz, Austria.

Tetewsky A, Ross J, Soltz A, Vaughn N, Anszperger J, O'Brien C, Graham D, Craig D, Lozow J (2009): Making sense of inter-signal corrections. Accounting for GPS satellite calibration parameters in legacy and modernize ionospheric correction algorithms. Inside GNSS, 4(4): 37–48.

Tran M, Hegarty CJ (2002): Receiver algorithms for the new civil GPS signals. In: Proceedings of The National Technical Meeting of the Institute of Navigation, San Diego, California, January 28–30: 778–789.

Tran M, Hegarty CJ (2003): Performance evaluation of the new GPS L5 and L2 civil (L2C) signals. In: Proceedings of The National Technical Meeting of the Institute of Navigation, Anaheim, California, January 22–24: 521–535.

Tsui JBY (2004): Digital techniques for wideband receivers, 2nd edition. Artech House, Norwood.

Tsui JBY (2005): Fundamentals of Global Positioning System receivers: a software approach, 2nd edition. John Wiley & Sons, Chichester, England, Hoboken.

Vanschoenbeek I, Bonhoure B, Boschetti M, Legenne J (2007): GNSS time offset: Effects on GPS - Galileo interoperability performance. Inside GNSS, 2(6): 60–70.

Viterbi AJ (1995): CDMA: principles of spread spectrum communication. Addison-Wesley, Redwood City.

Ward PW, Betz JW, Hegarty CJ (2006a): GPS satellite signal characteristics. In: Kaplan ED, Hegarty CJ (eds): Understanding GPS: principles and applications, 2nd edition. Artech House, Boston London: 113–142.

Ward PW, Betz JW, Hegarty CJ (2006b): Satellite signal acquisition, tracking, and data demodulation. In: Kaplan ED, Hegarty CJ (eds): Understanding GPS: principles and applications, 2nd edition. Artech House, Boston London: 153–241.

Won JH, Pany T, Hein GW (2006): GNSS software defined radio: real receiver or just a tool for experts? Inside GNSS, 1(5): 48–56.

Wooden W (1985): NAVSTAR Global Positioning System – 1985. In: Proceedings of the First International Symposium on Precise Positioning with the Global Positioning System, Rockville, Maryland, April 15–19: 23–32.

Yang C, Hegarty CJ, Tran M (2004): Acquisition of the GPS L5 signal using coherent combining of I5 and Q5. In: Proceedings of ION GNSS 2004, 17th International Technical Meeting of the Satellite Division of the Institute of Navigation, Long Beach, California, September 21–24: 2184–2195.

Yang C, Miller M, Blasch E, Nguyen T (2007): Comparative study of coherent, non-coherent, and semi-coherent integration schemes for GNSS receivers. In: Proceedings of the 63rd Annual Meeting of the Institute of Navigation, Cambridge, Massachusetts, April 23–25: 572–588.

Ziedan NI (2006): GNSS receivers for weak signals. GNSS technology and applications, Artech House, Norwood.