

TALLINN UNIVERSITY OF TECHNOLOGY  
School of Information Technologies

Reto Gähwiler, IVEM165519

**DEVELOPMENT OF A SOFTWARE-BASED  
GNSS-R RECEIVER FOR  
DELAY-DOPPLER MAP GENERATION**

Master's Thesis

Supervisor (TTU): Yannick Le Moullec  
PhD

Co-Supervisor (Chalmers): Thomas Hobiger  
PhD

Tallinn 2018

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Reto Gähwiler, IVEM165519

**TARKVARA BAASIL GNSS-R VASTUVÕTJA  
ARENDUS DELAY-DOPPLER MAP'I  
LOOMISEKS**

Magistritöö

Juhendaja (TTU): Yannick Le Moullec  
PhD

Kaasjuhendaja (Chalmers): Thomas Hobiger  
PhD

Tallinn 2018

## **Author's Declaration of Originality**

I hereby certify that I am the sole author of this thesis and this thesis has not been presented for examination or submitted for defence anywhere else. All used materials, references to the literature and work of others have been cited.

Author: Reto Gähwiler

12th June 2018

## Abstract

Remote sensing is an important tool for Earth Science to observe the environment and changes thereof. In general, the technique allows us to observe large regions with the ease of electro-magnetic waves. Among the different remote sensing techniques, scatterometry is the tool of choice for airborne platforms since the 1960's.

To reduce system complexity while having a great coverage, Global Navigation Satellite System-Reflectometry (GNSS-R) was introduced in the early 1990's allowing to measure surface wind-speeds over the oceans without the need of an active radar-system. Due to the threats from environmental changes and severe weather phenomena in the past decade, the necessity for more accurate and considerably cheaper solutions has increased.

In this thesis a generic implementation of a Delay-Doppler-Map (DDM) receiver was realised. The developed solution is capable of running on low cost systems based on Commercial off-the-shelf (COTS) components rather than making use of a proprietary FPGA implementation. The implemented DDM processing chain shows a Doppler spread of  $\pm 100$  Hz and delay range of  $\approx 10$   $\mu$ s, which matches with the geometrical meta-data of the test data-set.

The thesis is in English and contains 50 pages of text, 7 chapters, 27 figures, 11 tables.

# **Annotatsioon**

## **Tarkvara baasil GNSS-R vastuvõtja arendus**

### **Delay-Doppler Map'i loomiseks**

Kaugmõõtmised on oluline töövahend planeedi Maa keskkonna ja selle muutuste jälgimiseks. Elektromagnetlained võimaldavad meil üldjuhul lihtsasti jälgida suuri piirkondi. Alates 1960. aastatest on õhusõidukite puhul vastavate erinevate meetodite hulgast eelistatud skatteromeetria. Et vähendada süsteemi keerukust suurte piirkondade korral, pakuti 1990. aastatel välja globaalsetel positsioneerimissatelliitidel põhinev reflektomeetria (GNSS-R), mis võimaldab pinnatuule mõõtmist ookeanide kohal ilma aktiivradariteta. Seoses kliimamuutuste ohuga ja tõsiste ilmastikunähtustega viimasel kümnendil on kasvanud vajadus täpsemate ja suhteliselt odavate lahenduste järele. Käesoleva lõputöö raames loodi nn Delay-Doppler-Map (DDM) vastuvõtja üldine lahendus. Erinevalt FPGA-del põhinevatest suletud implementatsioonidest võib loodud lahenduse realiseerida odavate laiatarbekomponentide baasil. Rakendades loodud DDM protsessiahelat testandmetele, määrati Doppleri nihkeks  $\pm 100$  Hz ja signaali viiteks ca  $10 \mu\text{s}$ , mis ühtib andmekogu geomeetriliste metaandmetega.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 50 leheküljel, 7 peatükki, 27 figuret, 11 tabelit.

## Acknowledgments

In this place I would like to express my gratitude towards everyone who supported me during the time of creation of this master thesis, in particular Thomas Hobiger from Chalmers University / Onsala Space Observatory who was hosting and supervising me at Onsala as well as Yannick Le Moullec for his support as the supervisor at Tallinn University of Technology.

I would also like to mention the great support I experienced from the whole team at Onsala. Whether it was laboratory work, help with administration tasks or just having a *fika* I always appreciated the collegial environment.

Also to mention is the Institute of Space Science, CSIC of the Spanish National Research Council, which kindly provided the test data set for the validation of the DDM processing chain. In this place a special thanks goes to Weiqiang for his help with the interpretation of the data.

# List of Terms

**Code-Division Multiple Access (CDMA)** Code-Division multiple access is an access scheme for the physical layer where the access is guaranteed in time and frequency space. The separation of simultaneously transmitted data is based on orthogonal codes.

**Frequency-Division Multiple Access (FDMA)** Frequency-Division multiple access is an access scheme for the physical layer where the access is guaranteed in time but split into channels in the frequency domain.

**Globalnaya Navigatsionnaya Sputnikovaya Sistema (GLONASS)** Satellite based navigation system operated by Roscosmos.

**Global Positioning System (GPS)** Is a navigational system based on satellites broadcasting highly accurate timestamps for the purpose of location determination. GPS is owned and operated by the United States.

**Global Navigation Satellite System-Reflectometry (GNSS-R)** Radiometric measurement based on forward or back-scattered radio waves from global navigational satellite systems.

**GNU** GNU is a freely available operation system and software collection mostly licensed under GPL.

**Gqrx** Gqrx is an open source software defined radio (SDR) receiver powered by GNU Radio and realised with the Qt libraries.

**IQ** Representation of Radio Frequency (RF) signals whereas the I-channel is 90 degree ahead of the Q-channel. Therefore, also referred to as in-phase channel and quadrature channel.

**Low Earth Orbit (LEO)** Satellite orbit at 200-2,000 km altitude.

**Medium Earth Orbit (MEO)** Satellite orbit at 2,000-35,786 km altitude.

**nadir** Straight below, normal vector direction of Gravity.

**Significant Wave Height (SWH)** Statistical mean height of the highest third waves.

**Software Defined Radio (SDR)** Software defined radios are general purpose radio solutions capable to adapt to the current situation. Therefore, important parameters (e.g. coding scheme of the data-stream) for transmitter and receiver can be configured by the software and/or programmable hardware.

**zenit** Straight above, normal vector opposite of Gravity.

# List of Abbreviations

$\tau_{cor}$  Coherent Time

**AMSL** above mean sea level

**API** Application Programming Interface

**C/A** Coarse / Acquisition

**COTS** Commercial off-the-shelf

**DDM** Delay-Doppler-Map

**FFT** Fast Fourier Transform

**GNSS** Global Navigation Satellite System

**LHCP** Left Hand Circular Polarised

**LNA** Low Noise Amplifier

**LOS** Line of Sight

**MAMSL** Meter above mean sea level

**MIMO** Multiple Input - Multiple Output

**NLOS** Non Line of Sight

**PGA** Programable Gain Amplifier

**PRN** Pseudo Random Number

**RF** Radio Frequency

**RHCP** Right Hand Circular Polarised

**SP** Specular Point

**TIA** Transimpedance Amplifier

# Table of contents

<b>List of Terms</b>	<b>vi</b>
<b>List of Abbreviations</b>	<b>vii</b>
<b>List of Figures</b>	<b>x</b>
<b>List of Tables</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Scatterometry and Windspeed Retrieval . . . . .	2
1.2 Thesis Outline . . . . .	4
<b>2 Introduction GNSS-R &amp; DDM</b>	<b>6</b>
2.1 GNSS-Reflectometry . . . . .	7
2.2 Signal Conditioning . . . . .	10
2.3 Delay Doppler Mapping . . . . .	12
2.4 Algorithm Complexity . . . . .	15
<b>3 Offline Processor</b>	<b>18</b>
3.1 Dataset . . . . .	18
3.2 Reference DDM . . . . .	19
<b>4 Implementation</b>	<b>23</b>
4.1 Base Components . . . . .	23
4.2 Architecture . . . . .	24
4.3 Build Environment . . . . .	28
<b>5 Analog Front-End &amp; Digital Back-End</b>	<b>29</b>
5.1 SDR Evaluation . . . . .	29
5.2 LimeSDR . . . . .	30
5.3 Host-Computer . . . . .	36
<b>6 Implementation Results</b>	<b>38</b>
6.1 Dataset . . . . .	38
6.2 Implementation Results . . . . .	39
6.3 Platform Benchmarking . . . . .	41
<b>7 Summary</b>	<b>43</b>
7.1 Discussion . . . . .	43
7.2 Outlook . . . . .	47
<b>References</b>	<b>49</b>

# List of Figures

1.1.1 Bi-Static passive scatterometer setup based on Global Navigation Satellite System (GNSS) on board of the ISS (in the center) with different GNSS satellites [6]. . . . .	3
2.2.1 Processing of GNSS-R IQ-Signal and Pseudo Random Number (PRN) sequence. . . . .	11
2.3.1 Bi-Static radar setup using GNSS-R. Signal components with different Doppler shifts (B, B') and originating from Specular Point (SP) (A) are associated with the DDM and their lag. . . . .	12
2.3.2 Generation of DDM from correlation map. Accumulated power over Doppler frequency range (bottom left) and PSD of DDM (bottom right). . . . .	14
2.4.1 Computational complexity for the three algorithms and a variation of 1 / 2 / 4 Msps sampling rate. Logarithmic complexity axis. . . . .	17
3.2.1 DDM for 80 Msps signal with related Doppler power created with interpolated PRN-1. . . . .	20
3.2.2 DDM for downsampled 1 Msps signal with related Doppler power created with interpolated PRN-1. . . . .	20
3.2.3 DDM for 80 Msps signal with related Doppler power created with resampled PRN-1. . . . .	21
3.2.4 DDM for downsampled 1 Msps signal with related Doppler power created with resampled PRN-1. . . . .	21
3.2.5 DDM generated with PRN-1 (on the left hand side) and PRN-3 (on the right hand side). . . . .	22

3.2.6 Spatial distribution of iso-Doppler (Turquoise) and iso-delay (green/red) for PRN-1 on the left and PRN-3 on the right [9]. . . . .	22
4.2.1 Software structure of ramplerecorder. . . . .	25
4.2.2 Software structure of offline processor. . . . .	25
4.2.3 Software structure of Realtime Processor. . . . .	26
5.2.1 Block-diagram fo LimeSDR-USB hardware [10]. . . . .	31
5.2.2 Radio-Frequency frontend including Antenna-Splitter (1), Low-Noise-Amplifier (2) and LimeSDR (3). . . . .	32
5.2.3 RF-channel diagram [11] of the LMS7002M radio-chip used by LimeSDR. . . . .	34
5.2.4 Gqrx SDR interface tuned to 1.575 GHz showing GPS carrier. . . . .	35
6.2.1 Coherently integrated DDM of second 40218 with PRN1 at 1 Msps from up-looking Antenna. . . . .	39
6.2.2 Coherently integrated DDM of second 40217 with PRN1 at 1 Msps. . . . .	39
6.2.3 Coherently integrated DDM of second 40217 with PRN3 at 1 Msps. . . . .	40
6.2.4 Coherently integrated and incoherently averaged DDM of the seconds 40217-40226 with PRN1 at 1 Msps. . . . .	40
6.2.5 Coherently integrated and incoherently averaged DDM of the seconds 40217-40226 with PRN3 at 1 Msps. . . . .	41
7.1.1 1 Msps reference and implementation results compared to aligned 80 Msps reference DDM ( PRN1 on the left hand side, PRN3 on the right hand side). 1 Msps signals are scaled with scaling factor (SF) to fit reference signal. . . . .	44
7.1.2 Power vs. delay of reference and implementation on the left and linear difference between reference and implementation on the right. Both for PRN1 and the first second at 1 Msps. . . . .	44
7.1.3 Power vs. delay of reference and implementation on the left and linear difference between reference and implementation on the right. Both for PRN3 and the first second at 1 Msps. . . . .	44
7.1.4 Linear differential DDM of PRN1 (on the left) and PRN3 (on the right) at 1 Msps. . . . .	45

# List of Tables

2.4.1 Computational complexity to create one DDM over 1 s with three different algorithm. N=number of slices, M=samples per slice, $R_{\#1,2}$ =delay lags, $R_{\#3}$ =Doppler bins. . . . .	16
3.1.1 Data characteristics. . . . .	18
4.3.1 Required dependencies to build the project. . . . .	28
5.1.1 Key features of the LimeSDR . . . . .	30
5.2.1 Components used for RF-testing. . . . .	31
5.2.2 RF port matching and noise figures of the LimeSDR input/output ports . .	33
5.2.3 Available RF-amplifiers and gain levels for the LimeSDR . . . . .	33
5.2.4 LimeSDR sensitivity of channel 0 and all three input ports (peak / noise floor). Precision of the reading $\pm 0.5$ dBm. . . . .	34
6.1.1 Test data-set characteristics. . . . .	38
6.3.1 Reference systems on which solution was tested. . . . .	41
6.3.2 Performance test results. . . . .	42

# 1 Introduction

The observation of environmental parameters is a crucial step to understand the world surrounding us. Shall it be for an everyday application such as weather forecast or for the purpose of environmental science to determine the climatic changes, they both rely on the accuracy and reliability of physical measurements to create precise models to solve the addressed problems. One of these parameters are wind speeds. Wind is a fundamental force which can be measured easily on the continent's surface but its becoming a more complex problem if it should be measured over the ocean's surface.

Before remote sensing, this was solved by retrieving wind information from boats and buoys. This method allows one to do records for historical and statistical purposes limited in space. Due to limitation in coverage and means of communication, a more sophisticated solution is required for taking measurement from hundreds or even thousands of kilometres away from the continent, in the middle of an ocean. To measure under such circumstances, modern technologies such as radio scatterometers are required. Buoys and/or dropsondes are still required to cross validate the measurements of scatterometers [1].

This thesis describes a new approach of a software driven Delay-Doppler-Map (DDM) receiver. The system is consisting of a Software Defined Radio (SDR) and commercially available components, running a Global Navigation Satellite System-Reflectometry (GNSS-R) receiver capable of processing DDMs.

## 1.1 Scatterometry and Windspeed Retrieval

Scatterometry is a well established method to remotely measure the parameters of an object illuminated by a electro-magnetic wave front. Scatterometers are in the general language known as radars. The most common type of such an instrument for environmental monitoring would be a space based Side Looking Airborne Radar (SLAR) or Synthetic Aperture Radar (SAR) scatterometer which is mounted on a satellite orbiting around the Earth. Such a system actively transmits pulses and measures the back-scattered signal reflected by the earth's crust. The first relation between back-scattered radio-waves and surface wind-speed was noted in the late 1960's whereas the first spaceborne mission took place in the context of the Skylab mission in 1973 [2]. The most basic scatterometers are only capable of processing the delay and amplitude of the reflected signal. This results in a cell image representing an ellipse whereas the parameters retrieved from it give information about the distance and surface condition.

A higher resolution can be achieved by using SLAR and SAR systems which also consider the Doppler spread. As a result, besides the delay axis also a Doppler axis is introduced, allocating to each pixel a Doppler and delay location on the surface, hence the name Delay-Doppler mapping. DDM is a way to process the signals in such a way that coherent and incoherent signal parts are processed at the same time. DDM is not new and was already used for in space missions to create surface maps of planets such as from Mars [3]. Nevertheless, such a system still requires both parts, a transmitter and receiver, and hence results in an overall more complex and expensive system solution.

To overcome this limiting system factor, the usage of Global Navigation Satellite System (GNSS) signals for a passive scatterometer was first suggested by Martin-Neira [4]. In 2000, Zavorotny and Voronovich [5] suggested in their publication the usage of GNSS to retrieve wind speeds. This publication among others introduced the field of GNSS-R which can be described as a passive bi-static radar system. Combining the idea of GNSS-R and delay Doppler mapping results in a passive radar with the highest yet possible image resolution. Such a configuration was proposed by Zavorotny, Rodriguez-alvarez, Akos *et al.* [1].

## Space based Scatterometer

To increase the coverage and the continuity of measurements, space based scatterometers are the de facto standard to study the oceans. Receivers are located onboard of satellites in a Low Earth Orbit (LEO) where the transmitted signals typically originate from a GNSS located in a Medium Earth Orbit (MEO). Figure 1.1.1 shows as an example the case of the International Space Station (ISS) which receives simultaneously direct reference signals from GNSS satellites and reflected GNSS signals from the Earth's surface.

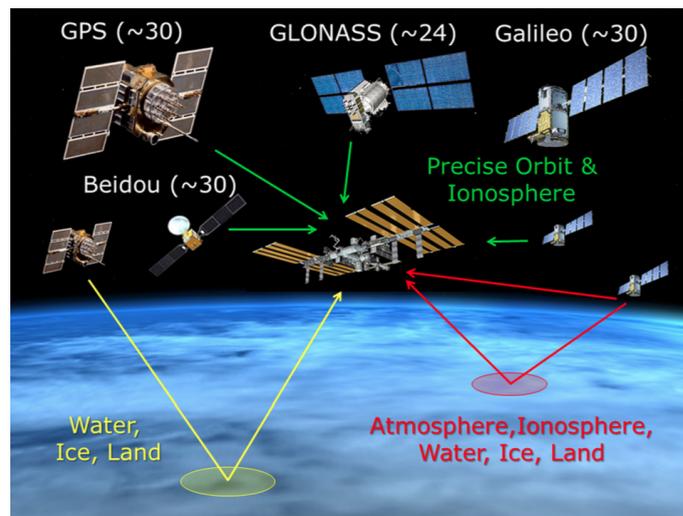


Figure 1.1.1: Bi-Static passive scatterometer setup based on GNSS on board of the ISS (in the center) with different GNSS satellites [6].

As illustrated in Figure 1.1.1, the signals originate from a GNSS satellite are received by a third party platform after being scattered by an illuminated object. Hence, it is a passive bi-static setup. GNSS signals are well suited for this purpose due to their nature of containing a well known sequence which makes it differentiable from other signals and therefore suitable for the purpose of scatterometry. Furthermore, the lower carrier frequency of the Global Positioning System (GPS) L1 carrier compared to dedicated scatterometers which work with the Ku or C-Bands (14 GHz, 5.6 GHz) is beneficial in terms of penetrating clouds and therefore availability.

## 1.2 Thesis Outline

This thesis is split into 7 chapters. In this chapter a technology overview was given together with the field of applications and put in contrast with the proposed solution of a software based GNSS-R DDM-receiver. Chapter 2 is dedicated to the background of DDMs, whereas in Chapter 3 the reference dataset and the resulting DDMs are presented. Chapter 5 gives an overview of the platform for which an implementation was done and Chapter 4 presents the actual implementation for the previously introduced platform. Chapter 6 shows the outcome of the created DDM processor and compares it with the offline processed references. Chapter 7 summarises the work done in this thesis.

### 1.2.1 Problem Statement

For the requirement of signal availability, GPS as a global navigational satellite system provider was preselected as the signal source. GPS fulfils the constraint of being freely available at anytime and from anywhere. In preliminary studies at Chalmers University [7] a GNSS-R receiver was developed and implemented for the use of ocean altimetry. As opposed to the previously developed receiver, this thesis focuses on a software-based DDM-receiver capable of running on Commercial off-the-shelf (COTS) components such as a SDR and an ARM based embedded system. The aim is to contribute to the development of one part of a future low cost platform for remote sensing applications. Therefore, the two following problem statements are made in this thesis:

- How to make use of the coherent and incoherent signal parts of reflected GNSS signals to image the ocean surface roughness (wind-speeds)?
- How can the proposed method of delay-Doppler-maps be applied to low-cost off the shelf component based solutions?

## 1.2.2 Thesis Tasks

In a first step an offline processed reference dataset has to be generated, following [8], [5] and [1], based on a provided data-set [9] to create reference DDMs. The reference DDMs shall consist of a high resolution dataset based on the 80 Msps sampling rate of the raw-data and a low resolution dataset sampled with 1 Msps representing the targeted sample rate.

In a second step a suitable C/C++ implementation for the offline algorithm has to be created. The implementation shall support a preselected SDR so it can sample data in real-time. Finally, the data has to be cross-validated with the previously generated reference DDMs.

## 1.2.3 Hypotheses

With the implementation of this low-end GNSS-R DDM-receiver the following two hypotheses can be made:

- Processing the coherent and incoherent signal parts for creation of a DDM the ocean surface roughness can be imaged in delay and Doppler range and therefore in higher resolution.
- The implemented solution processes the data in realtime and is capable to run on a low cost embedded system based on Linux.

## 2 Introduction GNSS-R & DDM

Radar is a well studied and understood field describing the scattering of microwaves and therefore also known as scatterometry. Radar can come in several different variants such as mono-, bi- or multi-static where the first one is considered to be active only and the two latter ones could appear as a passive system as well. Mono-static systems are based on an integrated transmitter-receiver solution which in most setups also share the same antenna. Bi- and Multi-static systems on the other hand are based on geographically separated locations for the transmitter and receiver. A multi-static system consists typically of several synchronised receivers for each transmitter which then have to be correlated to obtain a meaningful image.

A mono-static system is always an active system due to the requirement of a transmitter and receiver at the same location. To reduce system complexity and thus costs, a system only consisting of a receiver which makes use of a freely available radio signal is desirable. In such a setup the receiver is separated from the transmitter in space and detached from it in any way. Such a radar system is called a passive bi-static setup. As with most radar applications the imaging and therefore the determination of environmental parameters is desired. This requires a back- or forward-scattered signal, transmitted from a source and reflected from an illuminated surface. Additionally, a passive radar requires a highly and freely available - in time and space - radio signal with a well known coding scheme. Martin-Neira [4] first proposed the idea of a passive radar based on GNSS signals which formed the foundation for the field of GNSS-R applications.

## 2.1 GNSS-Reflectometry

Global Navigation Satellite System-Reflectometry (GNSS-R) describes a particular kind of reflectometry that is based on the processing of signals originating from a GNSS system such as GPS or Global Navigation Satellite System (GLONASS). In reflectometry not the actual message modulated into the signal is of interest but rather the information contained in the reflected radio signal scattered from an object. Such a reflection represents the surface which can alter its condition in terms of smoothness/roughness, signal absorption or in a simple location change over time depending on environmental parameters. The interaction between the surface and the microwave signal alters the latter's characteristics.

### 2.1.1 Reflection Types

There are two kinds of reflections of electro-magnetic waves including light and microwaves, i.e. specular reflections and diffuse reflections. The first one describes the reflection in case of a mirror like surface when the angle of incidence equals the angle of reflection,  $\alpha_{in} = \alpha_{out}$ . The second kind are reflections as a result of scattered waves thrown in all possible angles, which is the case of not perfect surfaces. Both reflection types share that the majority of the reflected signals will have a change in their polarisation. While the Line of Sight (LOS) signal is Right Hand Circular Polarised (RHCP), the reflected signal is Left Hand Circular Polarised (LHCP), which requires a corresponding antenna.

The area in which reflections can be recognised is known as the glistening zone (in Figure 1.1.1 red and yellow zones), which in case of visible light would be the glimmering region on a water surface illuminated by the sun. The size of the glistening zone depends on the angle of incidence and surface condition. The smallest possible zone is represented by a perfectly smooth surface with the origin of wave-front at the zenith. In case of GNSS-R, multiple glistening zones at the same time are possible depending on the number of visible satellites.

## 2.1.2 Signal Characterisation

A GPS signal - which this thesis is focusing on - belongs to the family of spread spectrum signals. Such a signal is built from the raw message, in case of GPS the navigation message, which is modulated with a modulo two operator with the so called Coarse / Acquisition (C/A) code. A bit transition in the navigation message results in a 180° phase flip of the modulated signal. The so modulated signal is then modulated onto the L1 carrier of GPS sitting at 1.57542 GHz. The key here is the modulation with the C/A-code. This code is also known as Pseudo Random Number (PRN) sequence and is unique for each GPS satellite and quasi-orthogonal to all the other present C/A codes.

The commercially used C/A code is a 1023 symbol-sequence modulated on a 1,023 MHz carrier and therefore 1 ms long in time. The modulation of a message with a PRN code is also known as spread-spectrum modulation since the generated radio signal will spread over a much wider bandwidth than the raw message's bandwidth. The spread spectrum makes the radio signal immune to noise whereas the PRN sequence gives it an unique pattern which can be detected with the help of a correlator but only if the sequence is known.

Furthermore, radio waves are shifted in frequency if the transmitter and/or receiver move in respect to the ground. This shift is known as Doppler-effect and in case of GNSS-R it is introduced due to the moving of the GPS satellite as well as the receiving platform relative to Earth. Equation 2.1.1 describes such a frequency shift:

$$f_o = \left[ \frac{1 \mp (v/c)}{1 \pm (v/c)} \right]^{1/2} \cdot f_s \quad (2.1.1)$$

The observed frequency  $f_o$  depends on the source frequency  $f_s$  and the ratio of the velocity  $v$  that is the relative movement along the propagation axis of the electro-magnetic wave and speed of light  $c$ . If the source moves away from the reflection,  $f_o$  will decrease and vice versa. In case of reflections, the scattered path has to be considered as a second path resulting in Equation 2.1.2 where  $v_{tx}$  and  $v_{rx}$  are the velocities of the transmitter and receiver, respectively. The evaluated Doppler shift is the mean Doppler shift recognised at the centre of the footprint. Due to geometrics, the frequency deviation will differ towards the edges along the velocity component.

$$f_{rx} = \left[ \frac{1 \mp (v_{tx}/c)}{1 \pm (v_{tx}/c)} \right]^{1/2} \cdot \left[ \frac{1 \mp (v_{rx}/c)}{1 \pm (v_{rx}/c)} \right]^{1/2} \cdot f_{tx} \quad (2.1.2)$$

As illustrated in Figure 2.3.1, the antenna footprint is projected onto the ground covering a certain area. In case of the GPS signal, this footprint covers the entire hemisphere visible to the satellite. Due to the curvature of the Earth as seen from a MEO orbit, the quality of the signal will degrade towards the edge of the projection. On the other hand, the receiver due to its much lower altitude has only a fractional coverage of the GPS footprint. Therefore, the signal can be seen as uniform in terms of geometrical dependencies which makes the receiving antenna sensitive to the overall geometrical alignment to the source. The size and shape of the footprint is a result of projection and relative movements as well as the condition of the surface.

The power of the signal at the receiver  $P_{rx}$  is given by the link budget in the form of Equation 2.1.3 [8], where  $G_{tx}$  and  $G_{rx}$  are the antenna gains of the transmitter and receiver, respectively,  $L_s$  is the path loss,  $G_{diff}$  any other applied amplification such as a Low Noise Amplifier (LNA) and  $L_{diff}$  any other kind of loss such as connectors. :

$$P_{rx} = P_{tx} + G_{tx} - L_s + G_{rx} + G_{diff} - L_{diff} \quad (2.1.3)$$

The total path loss, not considering scatter losses, from the transmitter to Earth and back to the receiver can be calculated by Equation 2.1.4. Here  $d$  is the distance and  $\lambda$  the wavelength of the carrier frequency, whereas  $d$  and  $\lambda$  have to use the same units. Substituting the values for GNSS and an receiver at around 3000 m above mean sea level (AMSL) a path loss of  $\approx 122$  dB is resulting:

$$L_s = 20 \cdot \log_{10} \left( \frac{4\pi \cdot d}{\lambda} \right) \quad (2.1.4)$$

## 2.2 Signal Conditioning

The raw signal received from the SDR has to be conditioned such that it can be used for further processing. The signal received from the SDR has to be compensated for the mean Doppler-shift, the delay differences between slices as well as from the 180° phase changes due to the navigation-message bit flip. This is required to keep coherency over the creation time of the DDM.

### 2.2.1 Baseband/Doppler Compensation

The received signal has to be baseband transformed under consideration of the mean Doppler shift at the Specular Point (SP). Equation 2.2.1 describes the baseband shifted signal  $U_B$  which is the down converted signal from the carrier frequency  $f_c$  considering the Doppler shift  $f_d$ . The information for the Doppler shift and carrier frequency has to be given by a reference receiver. In most cases,  $f_c$  will be 0 except if the GPS-signal is off the centre from the configured  $f_{rx}$  at the SDR.

$$u_B = S \cdot \exp^{2\pi i(f_c + f_d)t} \quad (2.2.1)$$

### 2.2.2 De-Spreading of GPS-Signal

GPS-signals are quite faint and hidden under the noise-floor. To successfully decode a message, the C/A code used for spreading during the modulation has to be known.

Figure 2.2.1 shows the required segments to de-spread a message. On the top: the baseband converted raw message  $U_B$  from the receiver as a complex IQ-stream; on the bottom left: a sequence of the PRN  $p$  which is circular cross-correlated with a equally long signal segment. Finally, on the bottom right: the output representing the cross correlation  $Y_{t_0, \tau}$  showing the correlation peak.

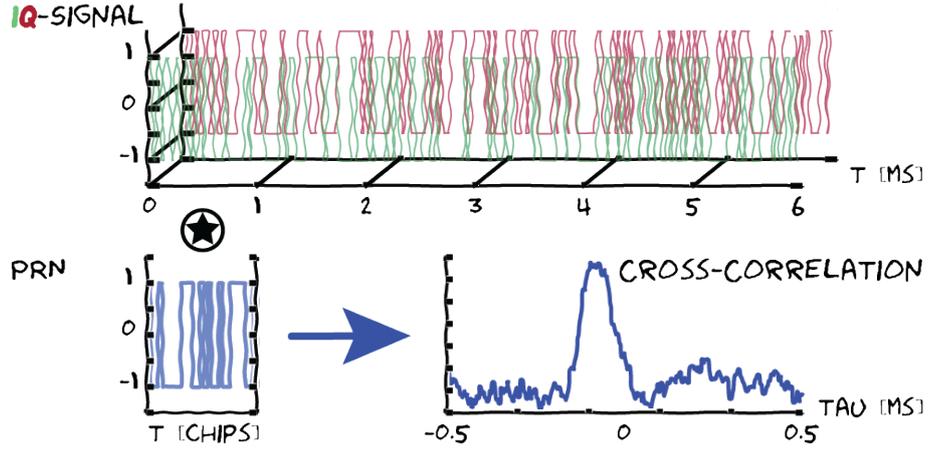


Figure 2.2.1: Processing of GNSS-R IQ-Signal and PRN sequence.

Equation 2.2.2 describes the regular formula for correlating two signals. To save computational time, the cross-correlation is done in the frequency domain as shown by Equation 2.2.3.  $T_i$  represents the coherent integration time over one PRN sequence length of 1 ms.  $\mathcal{F}$  is the Fourier transformation applied to the raw-signal  $u_B$  and the conjugate complex transformation to the PRN-sequence  $p$ . The resulting signal is the correlated spectra  $U_{Bp}$ .

$$Y_{t_0, \tau} = \int_0^{T_i} U_B(t_0 + t) \cdot p(t_0 + t + \tau) \cdot dt \quad (2.2.2)$$

$$U_{Bp} = \mathcal{F} \{u_B \star c\} = \mathcal{F} \{u_B\} \cdot \mathcal{F} \{p\}^* \quad (2.2.3)$$

### 2.2.3 Delay and Navigation-Bit Compensation

To align each correlation slice, delay and navigation bit sign have to be compensated and therefor tracked by the reference receiver. To gain the highest possible precision, this shift is applied in the frequency domain rather than over discrete indexes in the time domain. Equation 2.2.4 describes the index shifting:

$$U_{Bpc}(f) = \text{bitsign} \cdot \mathcal{F} \{u_B\} \cdot \exp(i2\pi f \cdot dly(f)) \quad (2.2.4)$$

The transformation is applied to the baseband-converted and fourier transformed signal before the correlation with the PRN sequence. A bit flip in the navigation message results in a sign change since the correlation coefficient becomes negative. Both compensations,

*bitsign* and *dly*, are available through meta data coming from the same receiver as the Doppler information.

### 2.2.4 Power Normalisation

To keep the signal immune to fading power, e.g. changing weather, the resulting correlation-slices are individually normalised with the square root of the power products from the signal and the PRN-sequence as shown in Equation 2.2.5:

$$U_{Bnorm} = \frac{U_B}{\sqrt{P_{UB} \cdot P_{CA}}} \quad (2.2.5)$$

## 2.3 Delay Doppler Mapping

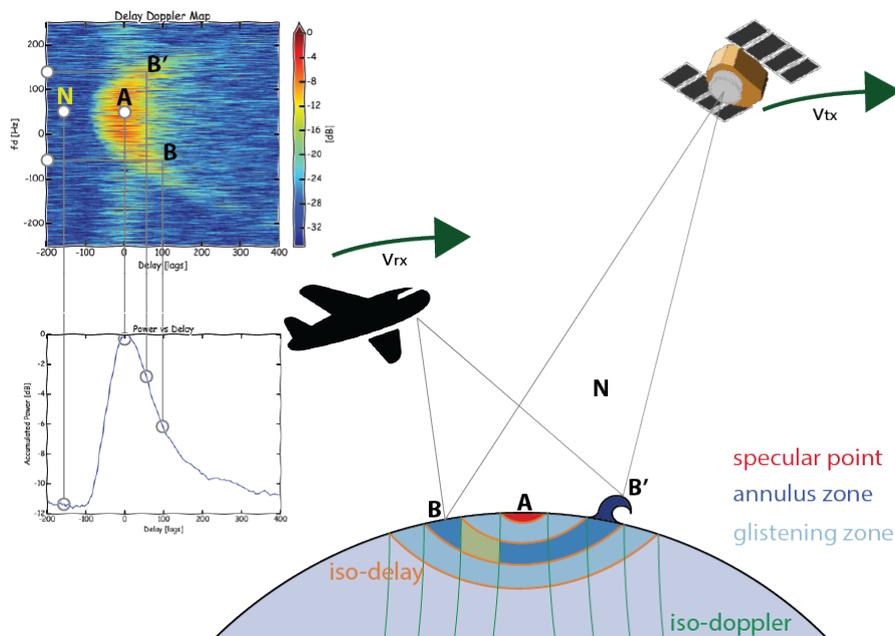


Figure 2.3.1: Bi-Static radar setup using GNSS-R. Signal components with different Doppler shifts (B, B') and originating from SP (A) are associated with the DDM and their lag.

Simple path length related measurements such as altimetry can be achieved by comparing a LOS signal with a reflected signal from the same GPS satellite which yields a significant delay due to the longer path of the reflected signal. Knowing the geometry, the vertical distance between the receiving platform and the surface from which the signal is reflected can be estimated.

To increase the precision as well as to measure other parameters such as the Significant Wave Height (SWH) which leads to surface wind speeds, a more sophisticated approach using DDMs is required. Delay Doppler Mapping as the name implies images not only the spread over the delay but also over the Doppler domain. This increases the precision in terms of image cell resolution.

Figure 2.3.1 illustrates the setup of a transmitter in form of a satellite and a receiver located on an aircraft at a much lower altitude. The illustration also contains the iso-distance ellipses around the specular point as well as the iso-Doppler lines perpendicular to the projected moving direction of the source. Signal parts originating from one annulus segment formed by the grid of iso-Doppler and iso-delay lines, the yellow area, would be represented in the same delay/Doppler bin. Additionally, a small wave is symbolised to illustrate how variation in the surface changes the mapping. The left hand side of the graphic shows the association of the different reflection points as well as the noise floor.

### 2.3.1 Creation of Coherent DDM

Figure 2.3.2 illustrates the creation of a DDM. The foundation for a coherent DDM is the Doppler and delay compensated correlations slices described previously in Subsection 2.2.2 and shown in Figure 2.2.1. It is important that those slices were Doppler and delay tracked, as well as compensated for the navigation-message-bit-flip. Figure 2.3.2 shows the formation of a coherent DDM from a set of coherent cross-correlation slices.

For the DDM transformation, a Fast Fourier Transform (FFT) is applied to each lag bin over all slices. Therefore, the input vector for the FFT consists of one point of each slice which can be expressed in  $y_{DDM}(\tau) = [x_0(\tau), \dots, x_N(\tau)]$ . Considering that Doppler and delay compensation took place, the specular point should become visible in the centre of the map.

### 2.3.2 Geometry

As described in the publication of Zavorotny and Voronovich [5], the geometry of the annulus zone depends on the delay and the bi-static geometrical setup which is a function

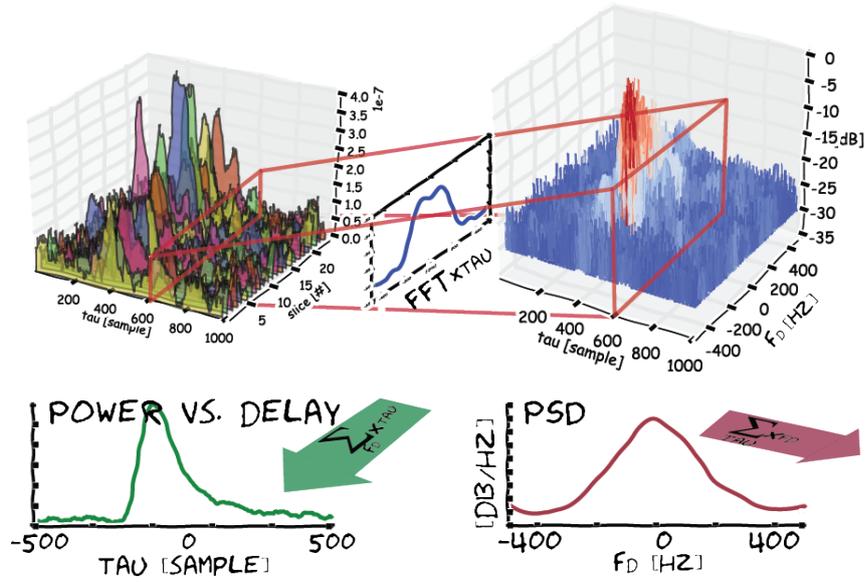


Figure 2.3.2: Generation of DDM from correlation map. Accumulated power over Doppler frequency range (bottom left) and PSD of DDM (bottom right).

of the locations and movements of the transmitter and receiver relative to the Earth’s surface. On the other hand, the glistening zone is formed by the surface condition driven by the environment (e.g. surface-winds). Together with the antenna pattern of the receiver, the shape of the outcoming DDM is defined. Those dependencies can lead to skewed DDMs where the Doppler distribution in the two arms will be asymmetric.

### 2.3.3 DDM-Parameters

Delay Doppler Mapping is a process over multiple frames which requires the previously described compensations so that the creation of a Doppler map stays coherent. The Doppler range and resolution is given by the number of slices over which the final FFT is applied according to  $f_{range} = [-f_s/2 \dots f_s/2]$  and  $\delta f = f_s/N$ , respectively. Where  $N$  is the number of slices and  $f_s$  the sampling frequency of slice generation which is  $\frac{1}{f_{s:signal} \cdot T_i}$ . Here  $T_i$  is the time over which the cross-correlation is applied to a signal slice and PRN sequence.  $T_i$  has to be no longer than the duration of a C/A sequence which in case of GPS is 1 ms.

Additionally, it is assumed that the environmental changes are much slower than the creation of a DDM. A further aspect which has to be considered is the relative movement of the receiver and transmitter in respect to the ground while changing the angular position

relative to the glistening zone which then results in changes of the actual annulus shape and therefore cell-size as mentioned in Subsection 2.3.2.

### **2.3.4 Power vs. Lag Graph**

One of the most important graphs for interpreting the condition of the glistening zone is the power vs. lag plot which is shown in Figure 2.3.2 on the bottom left. This graph depicts the accumulated power from all frequency bins of the same lag. For an up-looking antenna or a perfect reflection, this signal would show a sharp peak with a base width of  $\pm 1$  lag around the centre which represents a perfect correlation.

For a glistening zone which is not perfectly smooth this graph will show a trailing tail as shown in Figure 2.3.2 which comes from the multiple reflection points. Furthermore, waves close to the SP contribute power to the rising edge and just so that the slope decreases its steepness by some extent.

## **2.4 Algorithm Complexity**

During the course of this thesis, three different approaches were discussed for the creation of a DDM. Two of them are basically the same with the difference of the Doppler range. The third one computes a DDM for every slice at a coarse Doppler resolution. Table 2.4.1 shows the complexity of the three approaches. Each correlation comes at a cost of one forward and one backward FFT. The computed costs are not considering minor operations and all operators are complex.

Table 2.4.1: Computational complexity to create one DDM over 1 s with three different algorithm.  $N$ =number of slices,  $M$ =samples per slice,  $R_{\#1,2}$ =delay lags,  $R_{\#3}$ =Doppler bins.

Desc.	cMul.	cAdd.	cDiv.
#1, #2			
FFT & iFFT	$N \cdot M \log_2(M)$	$2 \cdot N \cdot M \log_2(M)$	
Correlator	$N \cdot M$		
Power	$2 \times N \cdot M$	$2 \times N \cdot M$	$2 \times N \cdot M$
Tracking	$2 \cdot N \cdot M$		
Doppler FFT	$R \cdot N / 2 \cdot \log_2(N)$	$R \cdot N \cdot \log_2(N)$	
#3			
FFT & iFFT	$(R + 1) \cdot N \cdot M / 2 \cdot \log_2(M)$	$(R + 1) \cdot N \cdot M \cdot \log_2(M)$	
Correlator	$R \cdot N \cdot M$	$R \cdot N \cdot M$	
Power	$R \cdot N \cdot M$	$R \cdot N \cdot M$	$R \cdot N \cdot M$
Tracking	$2R \cdot N \cdot M$	$2R \cdot N \cdot M$	

### 2.4.1 #1: Coherent Tracked DDM - Low Range

This algorithm [8] is based on the steps described throughout this chapter. The DDM is built from 1000 slices, each of it 1000 samples wide. Therefore, it offers a delay resolution of  $1 \mu\text{s}$  and a Doppler range of  $\pm 500 \text{ Hz}$ . The final DDM is imaged over much less than 40 delay lags and  $\pm 300 \text{ Hz}$  Doppler spread.

This algorithm was chosen for implementation since it offers a sufficient Doppler range for a aircraft flights. Additionally, the resulting 1000 by 1000 DDM matrix (delay x Doppler) simplifies the implementation. Table 2.4.1 shows the complexity of this algorithm.

### 2.4.2 #2: Coherent Tracked DDM - High Range

Since the low range implementation only has a Doppler range of  $\pm 500 \text{ Hz}$  and therefore only usable for low flying receivers a solution with a  $\pm 5 \text{ kHz}$  was calculated. The algorithm bases on the same principal but instead of 1000 correlation slices per DDM 10,000 slices are taken. To keep the 1 s per DDM, the time over which we correlate  $T_i$  is reduced form 1 ms to  $100 \mu\text{s}$ . Therefore, the DDM is generated form 10,000 slices each of it 100 samples. The computational complexity for this can be seen in Table 2.4.1.

### 2.4.3 #3: Arbitrary Doppler Resolution and Range

The two previous algorithm have a relatively high Doppler resolution of 1 Hz. With the algorithm described here [1], an arbitrary Doppler resolution and range can be chosen. This algorithm applies for every slice of  $1 \mu\text{s}$  length multiple frequency down conversions with different Doppler shifts. Therefore, the resulting delay range is 1,000 lags where the resulting Doppler range and resolution depends on the frequency conversions applied to the slice. The computational complexity of this algorithm is shown in Table 2.4.1.

### 2.4.4 Approximate Complexity Results

Figure 2.4.1 shows the logarithmic result for Table 2.4.1 with substituted values for the three algorithm. For the implemented algorithm #1  $M = f_s \cdot 0.001 = 1000, N = f_s/M = 1000, R = 40$ , for #2  $M = f_s \cdot 0.0001 = 100, N = f_s/M = 10,000, R = 40$  and for #3 the values are  $M = 1000, N = 1000, R = 41$ , with  $f_s$  being the sampling frequency. In case of the first two  $R$  represents the number of lags the DDM will be created for whereas for the latter one  $R$  represents the number of discrete Doppler frequencies.

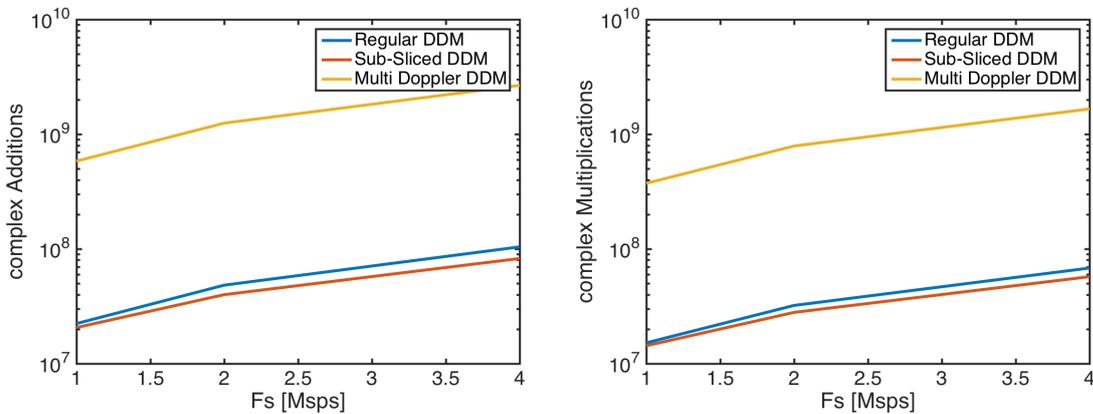


Figure 2.4.1: Computational complexity for the three algorithms and a variation of 1 / 2 / 4 MSPS sampling rate. Logarithmic complexity axis.

Substituting values of 1, 2 and 4 MSPS for  $f_s$  into the equations from Table 2.4.1 the approximate complexity in numbers are  $15.17 \times 10^6$  complex multiplications,  $22.33 \times 10^6$  complex additions and  $1.0 \times 10^6$  complex divisions for the implemented solution (Regular DDM in Figure 2.4.1).

## 3 Offline Processor

### 3.1 Dataset

The algorithm was first implemented in Matlab/Octave to have a baseline to verify the system. The data set used consists of real sampled data taken from an aircraft. In total 10 s of sampled data at 80 Msps taken from about 3000 m AMSL are available. The raw-data is split into 10 equal files representing 1 s each. The up-looking stream is available too and is required to retrieve the meta-data (mean Doppler, delay, nav-bit). The meta-data is retrieved by the mean of a software GPS-receiver. The data was supplied by the Institute of Space Sciences, CSIC of the Spanish National Research Council [9]. Table 3.1.1 summarises the characteristics of the available raw data.

Table 3.1.1: Data characteristics.

Parameter	Value
Antennas	RHCP, LHCP
Sampling Rate	80 Msps
Total Length	10 s
Filename	40217 ... 40226
Filelength	1 s
Available C/A	1, 3
Altitude	3022 m AMSL
Latitude	60.1750488861
Longitude	26.1498803139
Date	2015-12-03

For the actual implementation, a down-sampled version of the dataset is also required. The down-sampled data has to have a sampling rate of 1 Msps to keep the load on the SDR and host processor low. Therefore, the offline processor was ran twice, once with the 80 Msps dataset to generate a high resolution reference and once with the 1 Msps dataset to have a direct comparison with the implementation. Additionally, new meta-data was generated from the down-converted signal matching the 1 Msps data-rate to keep the results as realistic and consistent as possible.

## 3.2 Reference DDM

For the offline processor the DDM-algorithm was implemented in Matlab/Octave. The power was normalised to the peak value for both plots, the accumulated power vs. delay plot as well as for the actual DDM image. The plot axes are limited to the actual range required to display the part of the data with a high enough peak SNR (full range  $\pm 500$  Hz vs. 1000 chips).

The PRN-sequence for the commercially available GPS signal has a length of 1023 symbols and represents 1 ms in time. This sequence has to match the signal's data-rate and therefore requires a down- or up-sampling. For the 80 Msps signal this sequence has to be up-sampled to a sample length of 80'000 and for the 1 Msps signal it has to be down-sampled to a sample length of 1'000, respectively. For comparison, the up-sampling was realised through linear interpolation and by Matlab's resample module (from the signal processing toolbox) which is using a FIR filter with equivalent phase (filter delay compensation).

### 3.2.1 Interpolated PRN

The reference datasets presented in Figure 3.2.1 and 3.2.2 show DDMs generated with linearly interpolated PRN sequences. The interpolation type is a 1 D nearest neighbour method.

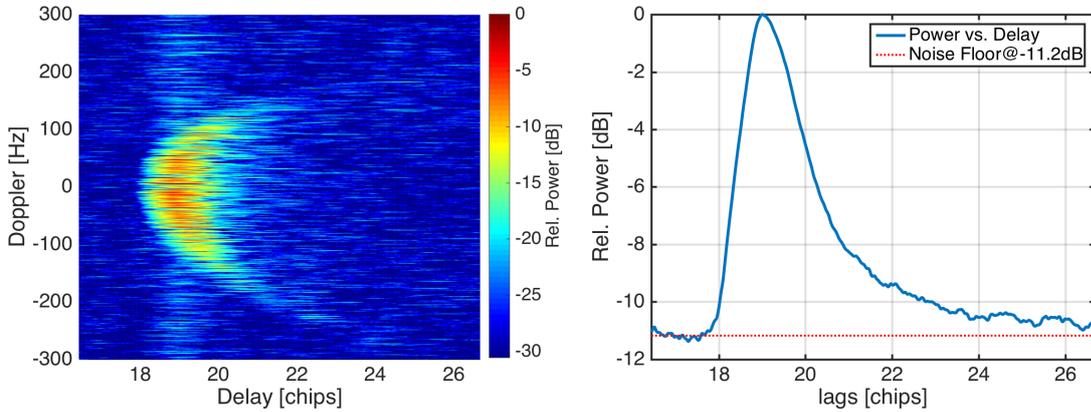


Figure 3.2.1: DDM for 80 Msps signal with related Doppler power created with interpolated PRN-1.

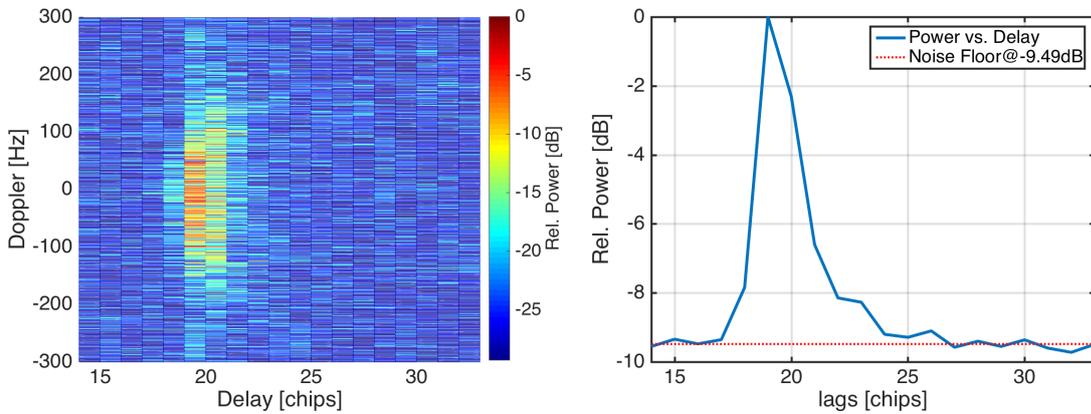


Figure 3.2.2: DDM for downsampled 1 Msps signal with related Doppler power created with interpolated PRN-1.

### 3.2.2 Resampled PRN

The results shown in Figures 3.2.3 and 3.2.4 show the DDMs generated with the *resample* module of Matlab. Comparing the DDMs in Figures 3.2.1 and 3.2.3 shows that the resampling with a FIR filter introduces some reflexions ahead of the actual DDM. Also the power plot also shows a lack of details (visible in the tail).

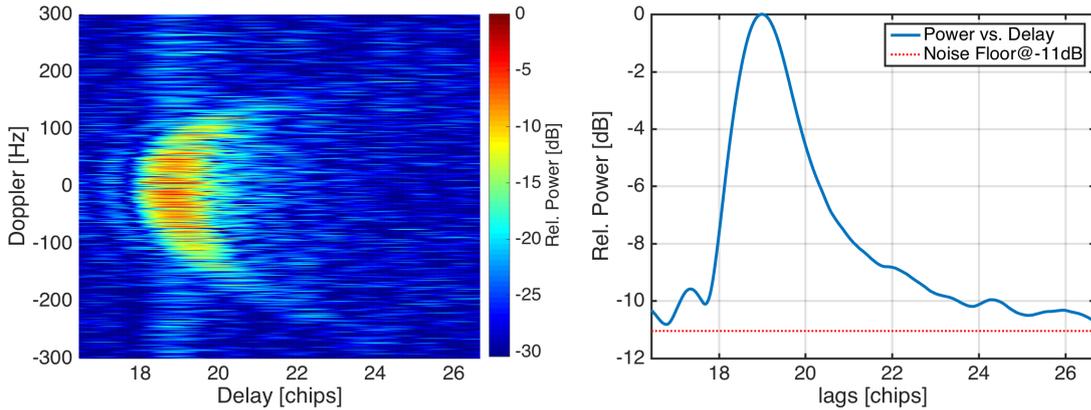


Figure 3.2.3: DDM for 80 Msps signal with related Doppler power created with resampled PRN-1.

Comparing the 1 Msps downsampled data represented in Figure 3.2.2 and 3.2.4 again, some increased signal levels can be recognised ahead of the DDM peak. More importantly, compared to the 80 Msps power plot, the 1 Msps power plot is also smoother but did not introduce distorting waves as seen in Figure 3.2.3 on the RHS.

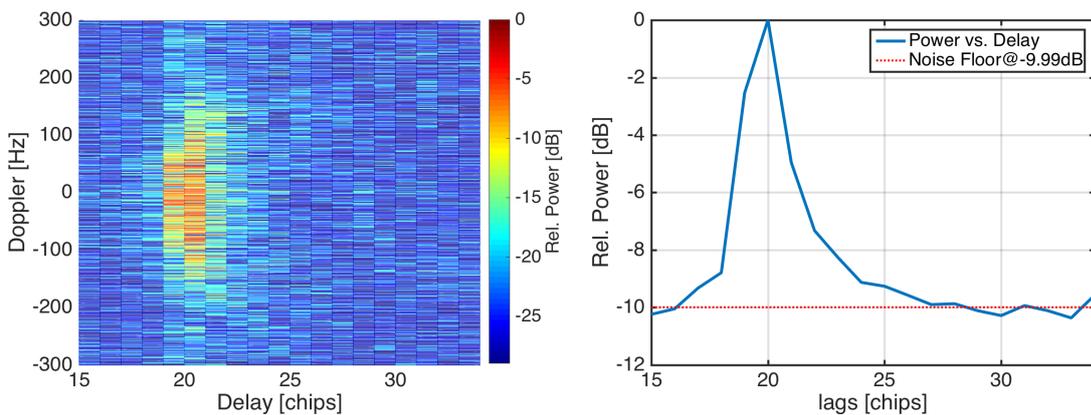


Figure 3.2.4: DDM for downsampled 1 Msps signal with related Doppler power created with resampled PRN-1.

### 3.2.3 Geometrical Difference

The test dataset contains data from multiple satellites of varying quality of which PRN-1 and PRN-3 are the two strongest ones. Due to the preferable geometry of the bi-static setup between the aircraft and the PRN-1 satellite this set was selected as a reference and PRN-3 as a secondary signal, respectively. Figure 3.2.5. compares the two DDMs for the PRN-1 and PRN-3.

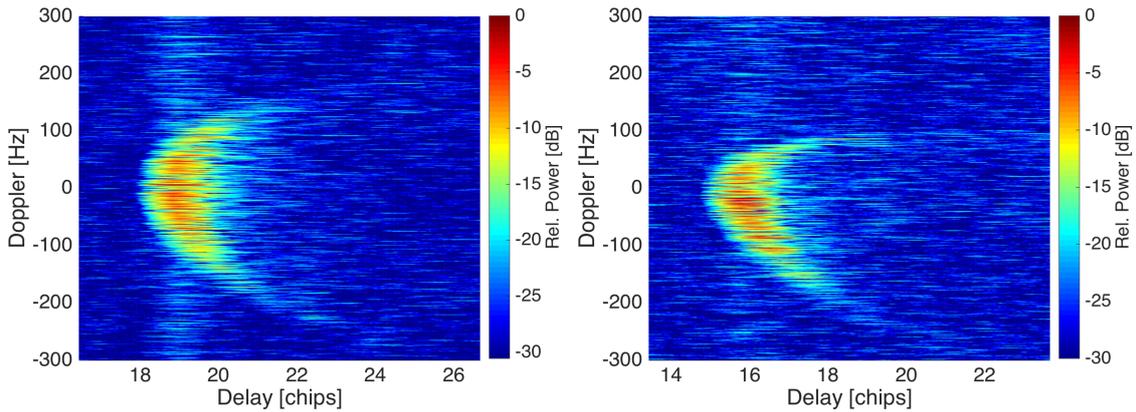


Figure 3.2.5: DDM generated with PRN-1 (on the left hand side) and PRN-3 (on the right hand side).

Figure 3.2.6 shows the glistening zones seen from the aircraft during the recording of the test samples. It can be seen that PRN-1 was almost centred and shows therefore the best characteristics in terms of geometry. The glistening zone of PRN-3 is off the centre due to a lower position of the satellite. This gives the glistening zone a certain distortion and an asymmetry in the distribution of the iso-Doppler and iso-Delay lines as can be seen in Figure 3.2.5.

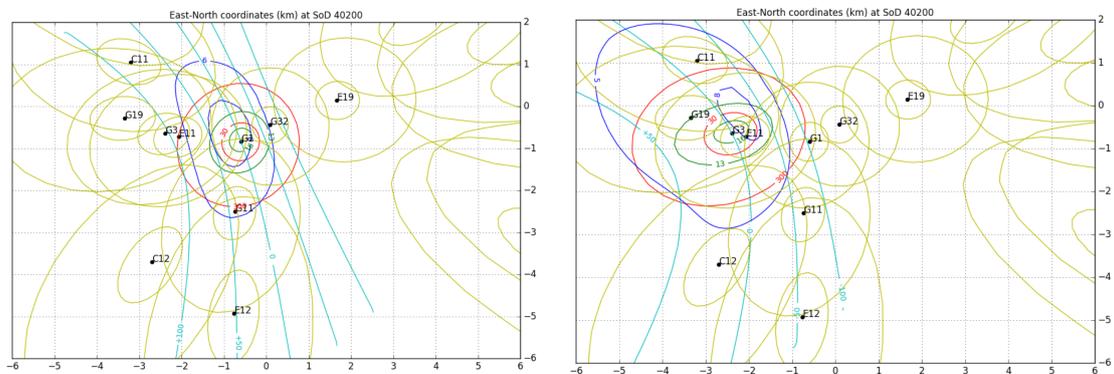


Figure 3.2.6: Spatial distribution of iso-Doppler (Turquoise) and iso-delay (green/red) for PRN-1 on the left and PRN-3 on the right [9].

# 4 Implementation

The processing chain from the raw-data to a DDM can be split into three major steps executed by the related base components. At the beginning is the sampling and buffering of the incoming data, in a second step the data are preprocessed, whereas in a last step the preprocessed data are transformed into a DDM and stored to disk. This chapter shows the realisation of an open-source based solution runnable on an Linux based embedded system fulfilling the requirements mentioned in Section 5.3.

## 4.1 Base Components

A base component is a block which combined with another base component forms an application. It forms the base of one processing step to create a DDM.

### 4.1.1 Sampler

The main task of the sampler is to collect all the data required by the pre-processor and DDM-processor. This includes receiving a data-stream sampled by a Software Defined Radio (SDR) and a meta-data stream, more about this in Sub-Section 4.2.6, coming from a reference receiver. Alternatively, the two mentioned streams can also be substituted by a file-reader. To reduce the overall USB access time to a minimum and therefore the time during which the processor is locked reading from the resource, the requested number of bytes to be read from the device should be much larger than one slice.

### **4.1.2 Pre-Processor**

The raw-data has to be baseband converted and Doppler-shifted as explained in Chapter 2. Furthermore, a cross-correlation between a data-slice and the PRN-sequence is required to de-spread the signal and therefore to get the signal out of the noise-floor. The processing is based on slices of the size of 1 ms to which the cross-correlation is applied. The pre-processor is located between the two main buffers, iqBuffer (Sub-Section 4.2.4) and doubleBuffer (Sub-Section 4.2.5), which form a loose connection between the different threads.

### **4.1.3 DDM-Processor**

The end of the processing chain is formed by the DDM-Processor which creates the actual Delay-Doppler-Map and the Power plot. The pre-processor first has to convert one second of raw-data before the DDM-Processor can be applied. Once the process is finished, the DDM together with the accumulated Doppler power is stored into a netCDF file.

## **4.2 Architecture**

The implementation is realised for three different applications which share the modules but serve a different purpose. All the applications consist of at least two working threads for processing the data. Multi-threading was chosen since the processing chain can be split into non-overlapping sections. In this way, different stages of the processing chain can process data simultaneously and therefore use the resources of a nowadays multicore embedded system. This means that the preprocessor can prepare data for the DDM-processor while that one is processing the previously prepared data and at the same time the sampler keeps sampling from the source to feed the preprocessor.

## 4.2.1 Sample-Recorder

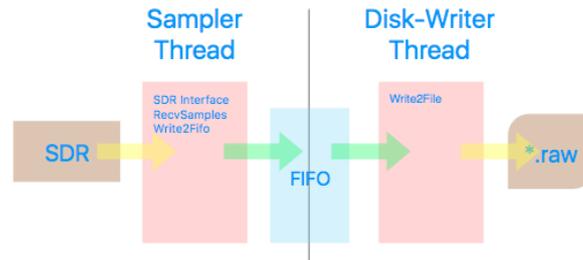


Figure 4.2.1: Software structure of sample recorder.

The sampler as shown in Figure 4.2.1 consists of a SDR sampler thread and a disk-writer thread. It is a simple raw-sample recorder which interacts through a SDR-reader with a connected radio. The second thread realises a disk-writer which simply writes the binary data into a file-sink. The output format stores the sample in an interleaved manner, whereas the i-channel and q-channel are written after each other. Hence, the size of a sample is two signed shorts, 2 Bytes each. The two threads are connected through the iqBuffer which is an implementation of a FIFO.

## 4.2.2 Offline-Processor

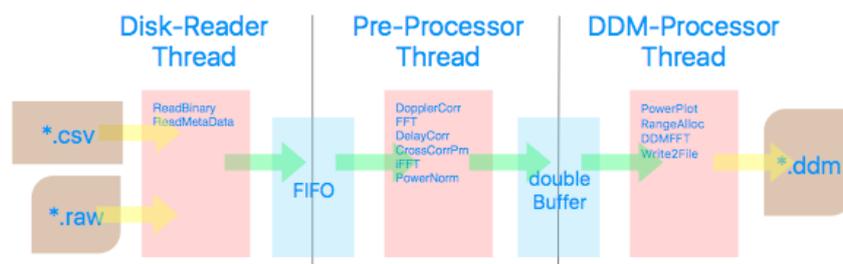


Figure 4.2.2: Software structure of offline processor.

As opposed to the Sample-Recorder, the Offline-Processor shown in Figure 4.2.2 is processing pre-recorded data. This can be used to either test the system with reference data or simply because the chosen embedded platform is not computationally powerful enough to process in realtime. The Offline-Processor consists of a disk-reader, a pre-processor and a DDM-processor thread. To process data an additional meta-data file is required with the file-format described in subsection 4.1.1. This meta-data stream is also handled by the disk-reader.

### 4.2.3 Realtime-Processor

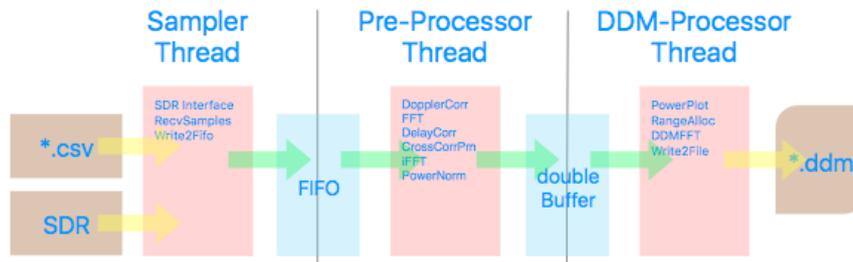


Figure 4.2.3: Software structure of Realtime Processor.

The Realtime-Processor combines the Sampler with the Offline-Processor. This results in a fully working chain capable of processing incoming data to DDMs. The performance and therefore the rate at which the DDMs are generated depend on the underlying embedded system.

### 4.2.4 iqBuffer

All the different architectures are relying on the same concept. One of the two buffers to synchronise the data-flow between the threads is the `iqBuffer`. The `iqBuffer` consumes complex `int16` samples coming from a producer such as a SDR-interface. A consuming thread can then request a block of data from the `FIFO`. The constraint of data coherence for the creation of one DDM implies that the buffer has to be at least of the size which is required to generate one DDM. In case the processing platform is fast enough, this constraint might be softened.

In this implementation the sampling rate is set to 1 Msps and the DDM size is 1 s. Together with a complex sample size of  $2 \cdot \text{sizeof}(\text{int16}) = 4 \text{ Bytes}$ , this buffer results in a total size of 4 MB.

### 4.2.5 doubleBuffer

The `doubleBuffer` is located between the pre-processor and the DDM-processor. It consists of two identical buffers of which each contains a data-set to generate one DDM. In this way the two working threads connected to the buffer can work simultaneously while both sides, producer and consumer, apply operations on the data. While the pre-processor is

filling one buffer the DDM-Processor is processing the other one. The references to the buffers are swapped as soon as both sides finished their tasks related to the buffer so that they can restart with a new set of data.

This doubleBuffer is relatively big in size. Each of the identical buffers contains 1 s of data at a data-rate of 1 Msps. Since the chosen output format of the FFT is a *double*, this results in a total size of 16 MB each or 32 MB in total, respectively.

#### **4.2.6 Meta-Data**

The meta-data provides information about the Doppler-shift and delay at the SP as well as the state of the navigation bit. The GNSS-R signal bouncing back from the ocean is rather noisy compared to a LOS signal from the up-looking antenna. Therefore, the meta-data has to be retrieved from an up-looking antenna and processed by a reference receiver. The decoded information and the required parameters are stored into a meta-data file. For every millisecond of data there has to be one meta-data block of the following csv-format: *delay;doppler;navbit*

## 4.3 Build Environment

The build environment is based on the GNU building tools consisting of *gcc/g++* and *cMake*. For *cMake* at least version 3.7 should be supported whereas *GCC* has to support *c++11*.

### 4.3.1 Dependencies

The project uses C++11 language constructs and links to several external libraries which are listed in Table 4.3.1 below.

Table 4.3.1: Required dependencies to build the project.

Lib Name	Descriptor	Used by/for	License
boostlog	Logging Maccros	Filtered log to file	free
pthread	Posix Multithread Lib	Runnable	LGPL
LimeSuite	API for LimeSDR	limeSDRReader	Apache License2.0
netcdf_c++4	API to write netCDF	DDMProcessor	free
fftw3	FFT implementation	Pre-/DDM-Proc.	GPL
m	math lib	fftw3	free
rt	realtime	boost log	free

# **5 Analog Front-End & Digital Back-End**

To record or process in realtime data, a receiver is required for the processing chain. Since this thesis' focus is set on COTS components, the available receivers are based on Software Defined Radio (SDR). Such a radio combines the analog front-end with a digital back-end which allows to adapt the receiver to a broad field of applications such as Global Navigation Satellite System-Reflectometry (GNSS-R). Furthermore, a SDR is a highly integrated piece of electronics which reduces the overall count of hardware components and therefore complexity of a system.

The analog front-end integrates all Radio Frequency (RF)-components required to down-convert a radio signal into the baseband. Depending on the SDR also some further filter banks are available, as in case of the LimeSDR. Furthermore, also the analog to digital conversion (ADC) takes place in the RF chip. The digital back-end usually consists of an FPGA having the primary task of connecting the front-end to the host system and if resources are left free it might also serve as a coprocessor.

## **5.1 SDR Evaluation**

For this thesis the SDR platform was already given with the LimeSDR[10]. Nevertheless, a few words related to the selection criteria are given in what follows. The requirements for the RF are the bandwidth (1 MHz for GPS), sampling rate (1 Msps) as well as the carrier frequency range (1.57542 GHz for GPS L1). Additionally, for further projects the platform should also support two RX and two TX channels. Table 5.1.1 shows the key

features of three different SDRs platforms including the LimeSDR.

Table 5.1.1: Key features of the LimeSDR<sup>1</sup>.

	<b>LimeSDR</b>	<b>Ettus B210</b>	<b>BladeRF</b>
Frequency	100 kHz-3.8 GHz	70 MHz-6 GHz	300 MHz-3.8 GHz
RF Bandwidth	61.44 MHz	61.44 MHz	40 MHz
Sample Depth	12 bits	12 bits	12 bits
Sample Rate	61.44 Msps	61.44 Msps	40 Msps
RF-Frontends	2 Rx/Tx	2 Rx/TX	1 Rx/Tx
RF-Chipset	LMS7002M	AD9361	LMS6002M
Interface	USB 3.0	USB 3.0	USB 2.0
FPGA	40 k LEs	100 k LEs	40/115 k LEs
Open Source	Full	Firmware, Schematics	Firmware, Schematics
Price	\$ 299	\$ 1'119	\$ 420

As can be seen, most given criteria are covered by all three SDRs. However, the BladeRF does not offer two dedicated channels which makes it an unsuitable candidate. If the price tag of the two remaining platform is compared it is clear that the LimeSDR also fulfils the criteria of the lowest price.

## 5.2 LimeSDR

The LimeSDR comes with the required RF front-end as well as a digital back-end to receive and stream the baseband signal from the radio board over USB to a connected host device. The overall hardware architecture of the LimeSDR is shown in the block-diagram in Figure 5.2.1.

The design can be roughly split into three groups. On the right hand side of Figure 5.2.1 the analog front-end with its *LMS7002M* controller is located. The front-end consists of a dual transceiver architecture offering two Rx and two Tx channels. The number of components is relatively small due to the high integration of the *LMS7002M* RF-chip [11]. Furthermore, a memory for the configuration as well as external RF matching networks are also in this block.

In the center of the same figure the FPGA, a *Cyclone IV* from Intel (former Altera), is located together with volatile memory. On the left hand side the USB3.0 controller

<sup>1</sup><https://www.crowdsupply.com/lime-micro/limesdr>, last accessed 2018-04-30

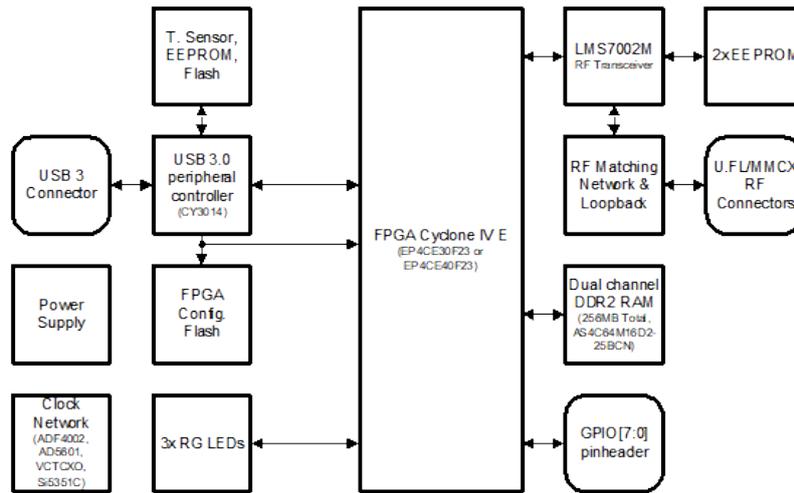


Figure 5.2.1: Block-diagram for LimeSDR-USB hardware [10].

and peripherals are located. The onboard FPGA can be turned into a versatile baseband processor if required.

### 5.2.1 GNSS RF-Front-End

Table 5.2.1: Components used for RF-testing.

Object	Descriptor
GPS-Antenna	Leica, AR10 1.0, 12 VDC/100 mA
GPS-Splitter	WR, Inc., NHILDCBS1X2, Antenna 12 V / J2 DC Block
LNA	Mini-Circuits, ZRL-2400LN, $G \approx 30$ dB
Cables	1 x $\approx 4$ m bedea#2261 HFX 50 BG 1.3L/3.6C 1 x $\approx 12$ m Ericsson Cables 5 M17/028-RG058 50Ohm 1 x 1.2 m GPS-Antenna DBBC Cable 1 x 0.2 m Semi-Rigid SMA to SMA 1 x 0.2 m Pig-Tail SMA to U.FL
Adapters	C-Type/BNC, N-Type/SMA, N-Type/BNC,

To test the LimeSDR concerning its GNSS-R capabilities a simple test setup was installed. This installation is shown in Figure 5.2.2 and consists of a GPS-Antenna (not visible), a RF-Splitter, a LNA and the SDR-PCB.

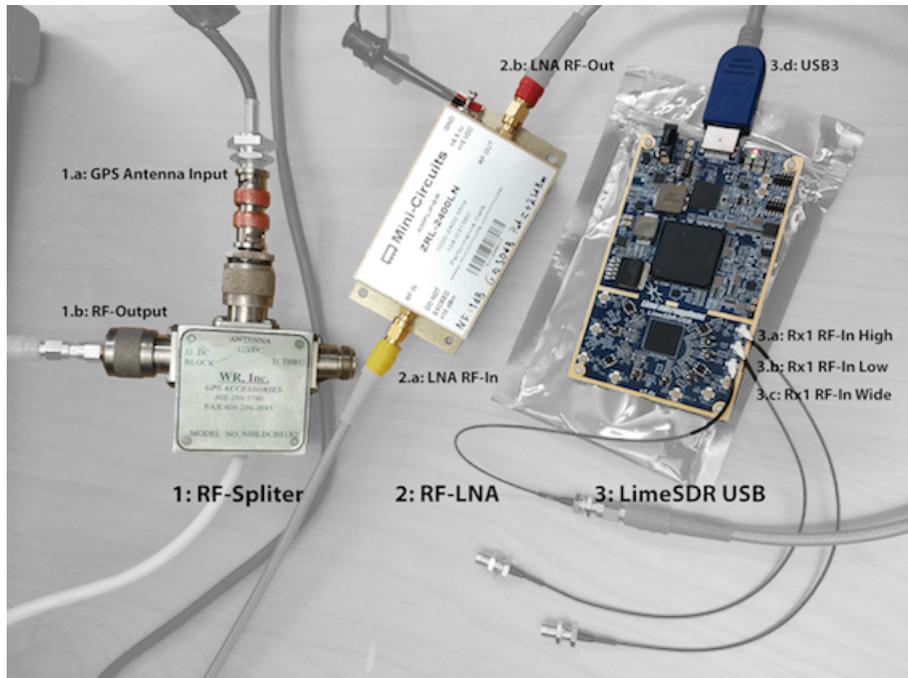


Figure 5.2.2: Radio-Frequency frontend including Antenna-Splitter (1), Low-Noise-Amplifier (2) and LimeSDR (3).

The RF-Splitter is required to supply power to the active GPS-Antenna while the RF path is kept clean from DC. Next in chain follows a LNA to amplify the input signal and finally the signal is fed to LimeSDR PCB containing the actual RF front-end. All the components used for the measurements are listed in Table 5.2.1.

### LimeSDR RF-Interfaces

The LimeSDR offers two Rx and two Tx channels where each Rx channel offers three different input ports and each of the Tx channel offers two different output ports. The channels are identical and can therefore be used for a Multiple Input - Multiple Output (MIMO) setups if required. To optimise the LimeSDR for mobile applications, the different ports are matched best to different frequency bands. Table 5.2.2 shows the optimised matching for each port as well as the supported frequency ranges by the LMS7002M.

<sup>2</sup><https://discourse.myriardrf.org/t/rf-connection-10-u-fl-connectors-6-rx-4-tx/305/9>, last visited 2018-04-30.

Table 5.2.2: RF port matching and noise figures of the LimeSDR input/output ports<sup>2</sup>.

Port	matched [MHz]	Chip Input [MHz]	Noise Figure [dB]
LNAL	800-900	0.1-2000	<2
LNAH	1900	1500-3800	<3
LNAW	0.1-2000	0.1-3800	5-7
TX1	2600	0.1-3800	
TX2	2000	0.1-3800	

## Receiver Parameters

The LimeSDR offers a set of parameters to tune the RF front-end to some extent. This includes several different amplifier gains which can be individually set or as a group, depending on the chosen Application Programming Interface (API). Furthermore, also the analog sampling rate and carrier frequency are configurable. The three relevant gains available are shown in Table 5.2.3.

Table 5.2.3: Available RF-amplifiers and gain levels for the LimeSDR<sup>3</sup>.

Amplifier	Gain Range	Steps
LNA	0 ... 30 dB	0-6 dB: +1 dB, 6-30 dB: +3 dB
TIA	0 / 9 / 12 dB	3 pre-set gains
PGA	-12 ... 19 dB	+1 dB

Figure 5.2.3 shows the internal architecture of the LMS7002's RF-input stage and the location of the three amplifiers. The LNA is placed at the RF-input before the mixer. For this amplification stage a low noise figure is important. The Transimpedance Amplifier (TIA) is located after the mixer and before the low-pass filter. To prevent intermodulation, this stage should be kept at a minimal gain level. To use the ADC stage optimally a last amplification stage with a large dynamic range is placed just in front of it. The Programmable Gain Amplifier (PGA) can be controlled in a range of 30 dB.

<sup>3</sup><https://github.com/myriadrf/LMS7002M-docs>, LMS7002M SPI Register Control, Page 39, last visited 2018-04-30

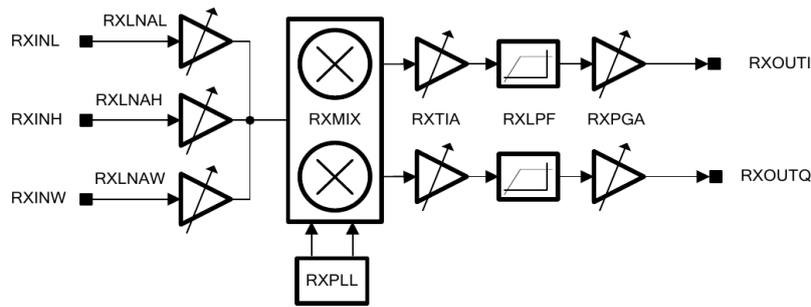


Figure 5.2.3: RF-channel diagram [11] of the LMS7002M radio-chip used by LimeSDR.

### LimeSDR Sensitivity

The LimeSDR offers three different amplifiers for the purpose of analog signal conditioning as shown in Figure 5.2.3. To estimate the sensitivity of the radio front-end and therefore the requirement of an external amplifier, a simple test was created including the following components:

- HP 8642B Signal Generator 0.1-2100 MHz
- 1.2 m GPS-Antenna DBBC Cable
- 0.2 m Pig-Tails SMA to U.fl Cable

Table 5.2.4 shows the signal levels read from Gqrx which was used to visualise the LimeSDR's output. The test was applied only to channel 0 and to all three available input matching networks (ports). The carrier frequency was set off the GPS centre frequency so that the DC in the spectra could be ignored. The calibrated generator was set to -84 dBm output power for the measurement.

Table 5.2.4: LimeSDR sensitivity of channel 0 and all three input ports (peak / noise floor). Precision of the reading  $\pm 0.5$  dBm.

SDR Port	@ 1.575 GHz [dBm]	@ 1.57562GHz [dBm]	@ 1.5758GHz [dBm]
High	-90 / -87	-89 / -86	-89 / -86
Low	-90 / -87	-89 / -86	-89 / -87
Wide	-90 / -87	-90 / -86	-91 / -87

The figures in Table 5.2.4 were evaluated with Gqrx's spectrum analyser. Therefore, a large FFT was used with averaging and hold active (Fig. 5.2.4). Bringing these figures into contrast with the LMS7002M RF characteristics from Table 5.2.2 it becomes clear that LNAL and/or LNAH are suitable for the purpose of a GNSS-R receiver based on the GPS L1 frequency.

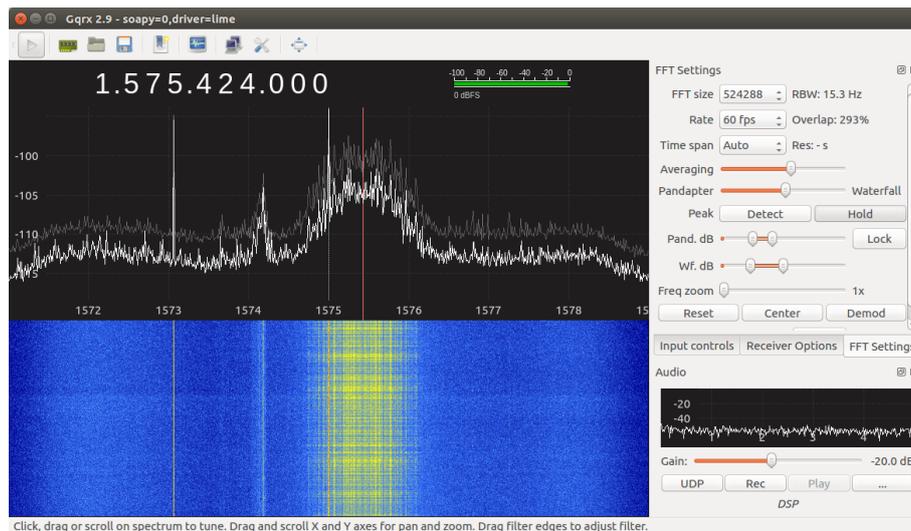


Figure 5.2.4: Gqrx SDR interface tuned to 1.575 GHz showing GPS carrier.

## 5.2.2 Digital Back-End

The digital backend of the LimeSDR is based on an FPGA processor from Intel (former Altera) of the type Cyclone 4 with 39'600 logic elements combined with 256 MB RAM and a USB3.0 controller (FX3). This FPGA could be used as a baseband processor if required. In this thesis the provided FPGA firmware from Myriadrf, the vendor of LimeSDR, is used. The standard functionality of the firmware covers the following operations<sup>4</sup>:

- "Glue logic between LMS7002M and FX3 USB MCU"
- "Signal waveform player (WFM) implementation"
- "Tx and Rx data stream synchronization if necessary"
- "NIOS MCU to control LMS7002M and on board devices like thermometer, Si5351C clock synthesizer etc."

<sup>4</sup><https://discourse.myriadrf.org/t/limesdr-fpga-secrets/903/2>, last visited 2018-05-01.

## 5.3 Host-Computer

The ultimate target platform for this implementation is a Linux based embedded system such as the Beaglebone-Black. Such COTS low cost solutions are typically based on an ARM core running on Linux. The DDM processor requires a considerable amount of memory as well as computational time to process the raw-data supplied by the SDR.

### 5.3.1 Requirements

**USB bandwidth:** To stream the data from the LimeSDR to the host system a USB connection is required. The GPS signal L1 requires a data stream of 1 Msps complex *int16*. With higher supported USB transmission speeds, the time for transmitting to the host system can be minimised. Therefore, a USB 3.0 is preferable.

**Power supply:** The LimeSDR PCB is host to the analog front-end as well as the digital back-end consisting of a FPGA, RAM and other peripheral chips. Only USB 3.0 supplies enough power, lower USB revisions require an additional power supply through the barrel connector on the board.

**Computational power:** Due to the complexity of a DDM receiver, the target platform should offer a powerful multicore CPU combined with sufficient memory (more about in Chapter 4). Additionally, for the purpose of recording raw samples, a storage bandwidth larger than the USB bandwidth for the selected sampling rate is required.

### 5.3.2 Beaglebone Black

The Beaglebone-Black is an ARM-based embedded system. The board hosts an ARM A8 single-core clocked at 1 GHz supported by 512 MB RAM and 4 GB eMMC non-volatile flash storage, one ethernet and one USB 2.0 port. The system comes with a Debian Linux pre-installed but Ubuntu images are also available<sup>5</sup>. The Beaglebone-Black was chosen since it was already available and could be reused for this purpose even though it was known that the offered computational power might not be sufficient.

---

<sup>5</sup><https://beagleboard.org/black>, last visited 2018-05-07

## **Bottlenecks**

**Single-Core:** The Beaglebone Black contains a single core ARM chip. For the multi threaded implementation and the computational time required, this might not be sufficient without the loss of raw-samples.

**USB Bandwidth:** A first test showed that the Beaglebone Black might be a bit short if it comes to USB transmission bandwidth. With a sample rate of 2 Msps and a sample size of 4 Bytes per sample, resulting in a bandwidth of 8 MBps, the board reached it limits.

**eMMC bandwidth:** Also the bandwidth to the onboard storage is limited which together with the single core may result in loss of computational time while data is transmitted to the storage unit.

# 6 Implementation Results

## 6.1 Dataset

As discussed in Chapter 3, the test data-set has to match the sample rate of the realtime implementation which is defined by the system design at 1 Msps. Both required streams, the raw samples as well as the meta-data, are either down-sampled or generated from a down-sampled reference data-set. All down-sampled data based on the data-set introduced in Chapter 3. The test data-set is characterised in Table 6.1.1.

Table 6.1.1: Test data-set characteristics.

<b>Parameter</b>	<b>Value</b>
Antennas	RHCP, LHCP
Sampling-Rate	1 Msps
Total Length	10 sec
Filename	40217 ... 40226
Filelength	10 x 1 sec, 1 x 10 sec
Available C/A	1, 3

## 6.2 Implementation Results

The results are either directly generated or post-processed outputs of the DDM processor. All DDMs were generated with the data-set described in Table 6.1.1. Figure 6.2.1 shows the up-looking DDM and power vs. delay plot which can be taken as a reference for the Doppler and delay spread. Since there is no delay and doppler at the SP, the power over delay plot shows peak right in the centre of the DDM.

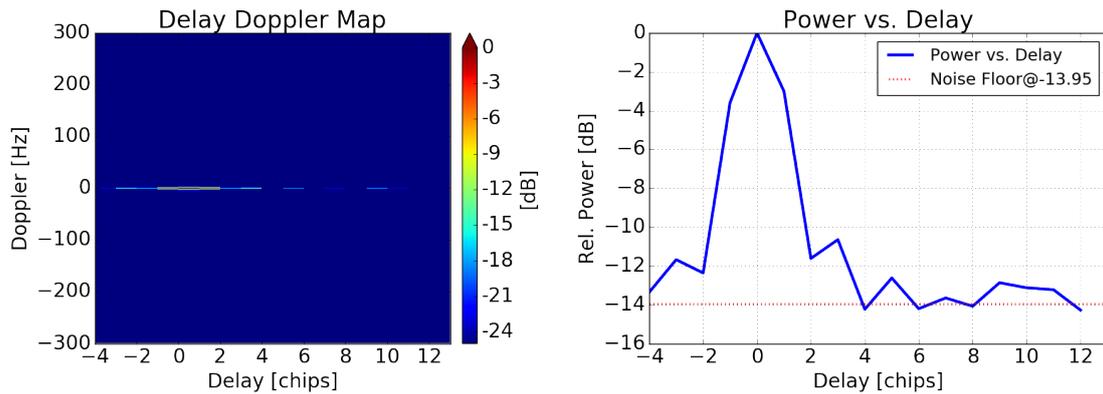


Figure 6.2.1: Coherently integrated DDM of second 40218 with PRN1 at 1 Msps from up-looking Antenna.

### 6.2.1 1 second Coherent DDM

Figure 6.2.2 and 6.2.3 show the DDMs generated by the implementation. The DDMs shown represent the first second of the data-set, sampled at 1 Msps and coherently integrated over one second.

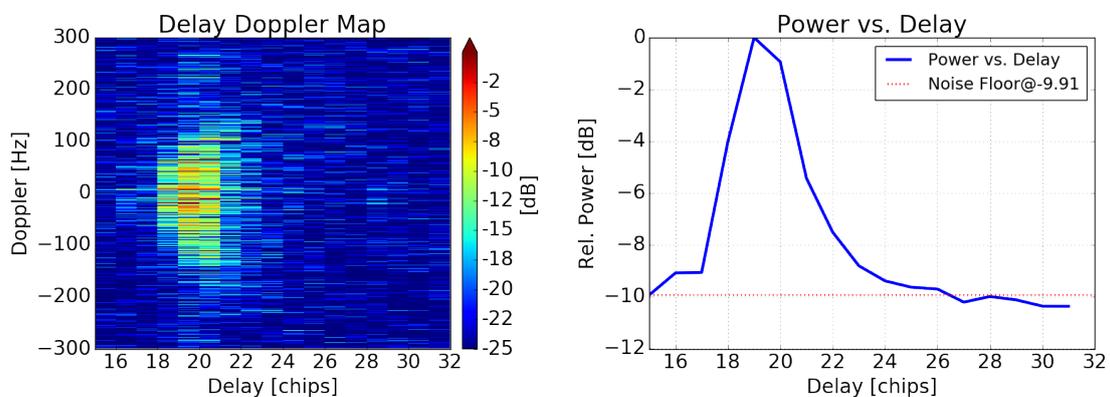


Figure 6.2.2: Coherently integrated DDM of second 40217 with PRN1 at 1 Msps.

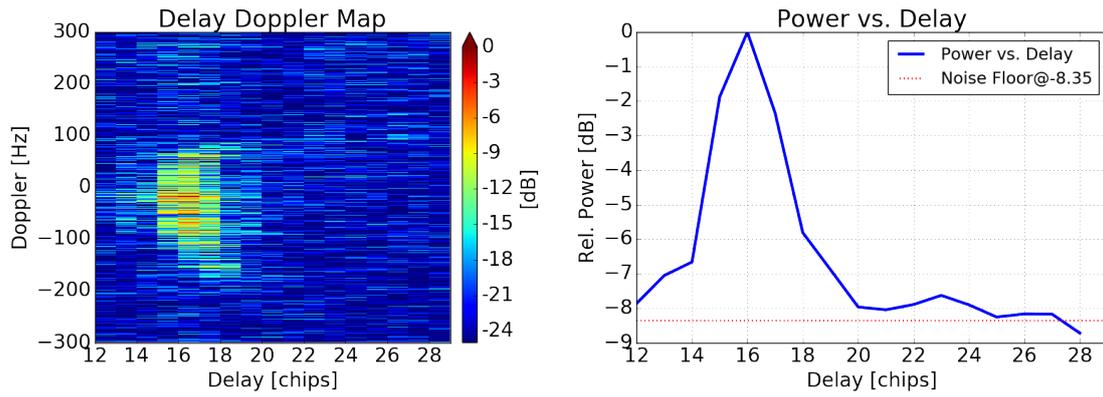


Figure 6.2.3: Coherently integrated DDM of second 40217 with PRN3 at 1 Msps.

## 6.2.2 10 second Incoherent Averaged DDM

Figure 6.2.4 and 6.2.5 show the DDMs generated by the implementation and post-processed on a workstation. The DDMs shown represent 10 seconds of data, sampled at 1 Msps, coherently integrated over one second and incoherently averaged over 10 seconds. Therefore, ten DDMs were accumulated and power normalised in a post-process to create this incoherent DDM.

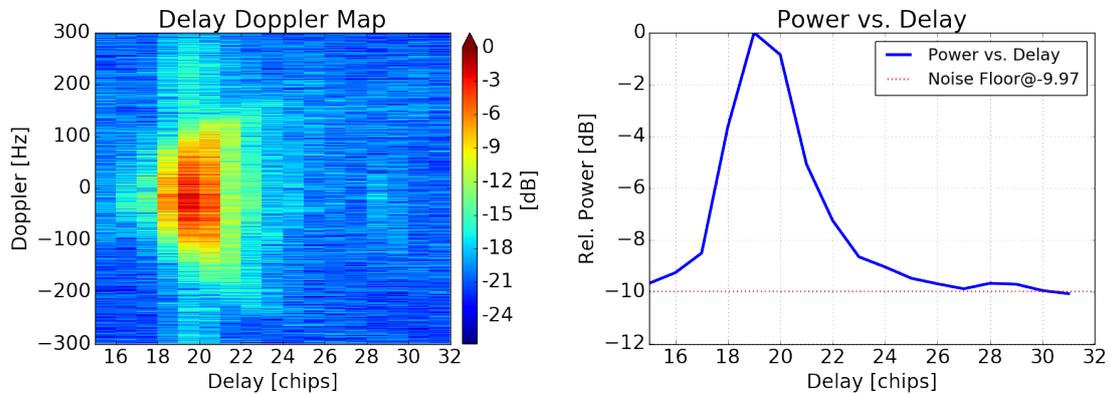


Figure 6.2.4: Coherently integrated and incoherently averaged DDM of the seconds 40217-40226 with PRN1 at 1 Msps.

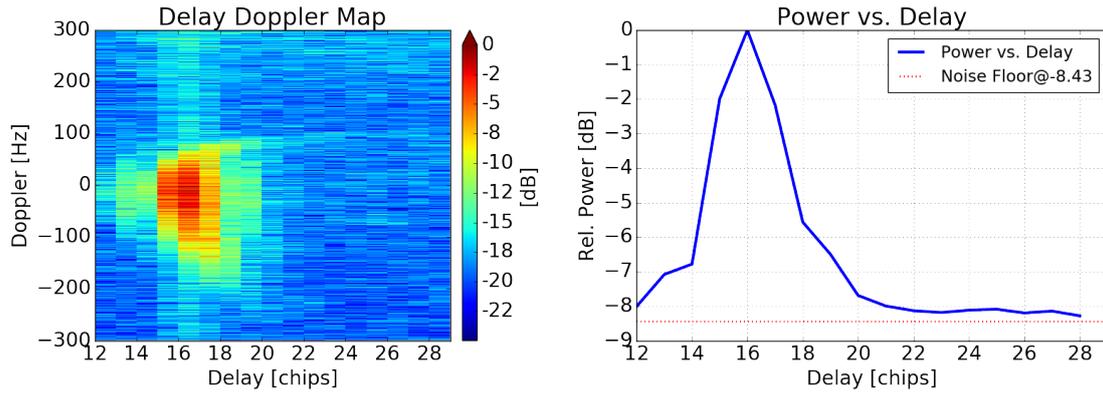


Figure 6.2.5: Coherently integrated and incoherently averaged DDM of the seconds 40217-40226 with PRN3 at 1 Msps.

## 6.3 Platform Benchmarking

The computational platforms used in this thesis are described in Table 6.3.1. The DDM-receiver is a pure software driven implementation capable on running on most Linux systems having preinstalled the required libraries (see 4.3.1). Therefore, the workstation and the available Beaglebone-Black were profiled.

Table 6.3.1: Reference systems on which solution was tested.

	Workstation	Beaglebone-Black
CPU	Intel i7 4770	Cortex-A8
Cores/Threads	4 / 8	1 / 1
Clock [GHz]	3.5 / 3.9	0.8
Cache	8 MB	32 KB / 256 KB
RAM [GB]	32	0.5
Storage [GB]	SSD > 128	eMMC, 4
USB	3.0	2.0

### 6.3.1 Benchmark Results

Table 6.3.2 shows the performance yield by the two platforms. As expected, the Beaglebona-Black does not perform nearly as good as the workstation. Also as expected did the embedded-system not achieve realtime performance. The workstation on the other hand performed as expected and was easily capable of processing the data in realtime. Real-time means processing continuously one DDM per second without losing any raw-data samples.

Table 6.3.2: Performance test results.

<b>Test</b>	<b>Workstation</b>	<b>Beaglebone-Black</b>
DDMs per minute	60	$\approx 6$
CPU load	2%	$\approx 100\%$

The memory consumption was about the expected 36 MB for both and stayed stable during operation. Valgrind, a memory leak detection tool, indicated that no growing memory leak occurred.

# 7 Summary

The baseline for the discussion is the 80 Msps offline processed dataset based on the dataset provided by CISC [9]. This data was recorded from an aircraft flying at an altitude around 3000 Meter above mean sea level (MAMSL). Therefore it can be said that due to the low altitude and velocity of the aircraft, the Doppler zones are relatively wide compared to the glistening zone as shown by Zavorotny and Voronovich [5] and as shown in Figure 3.2.6.

## 7.1 Discussion

Figure 7.1.1 shows the aligned 80 Msps reference in green, the 1 Msps reference in red and the 1 Msps implementation in blue, plotted over each other. While the 80 Msps and the 1 Msps references match quite well, the implementation shows a widen-up image. The low sampling-rate, resulting in a coarse delay resolution, together with the wide wave form from the implementation, may lead to wrong results while interpreting the plot as shown by Komjathy, Armatys, Masters et al. [12].

Figure 7.1.2 and 7.1.3 show the difference between the 1 Msps reference and the implementation for PRN1 and PRN3, respectively. In this comparison the two wave forms were not scaled. The implementation shows for both PRNs a slightly better relative noise floor than the reference. Also is the wide wave-form of the implementation in the linear differential plot (RHS) clearly visible.

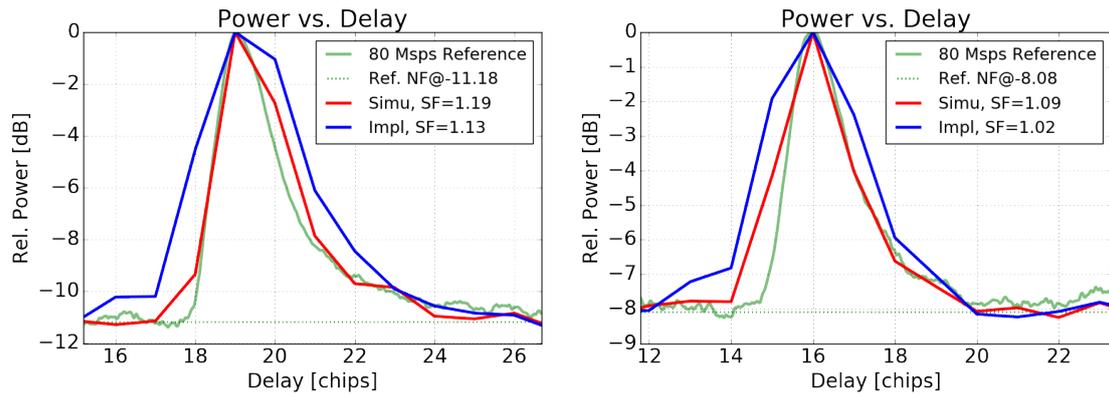


Figure 7.1.1: 1 Msps reference and implementation results compared to aligned 80 Msps reference DDM ( PRN1 on the left hand side, PRN3 on the right hand side). 1 Msps signals are scaled with scaling factor (SF) to fit reference signal.

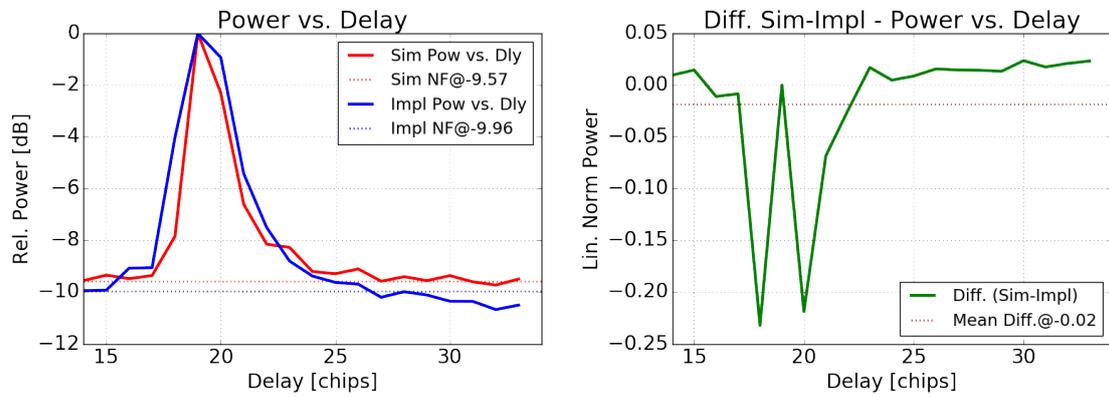


Figure 7.1.2: Power vs. delay of reference and implementation on the left and linear difference between reference and implementation on the right. Both for PRN1 and the first second at 1 Msps.

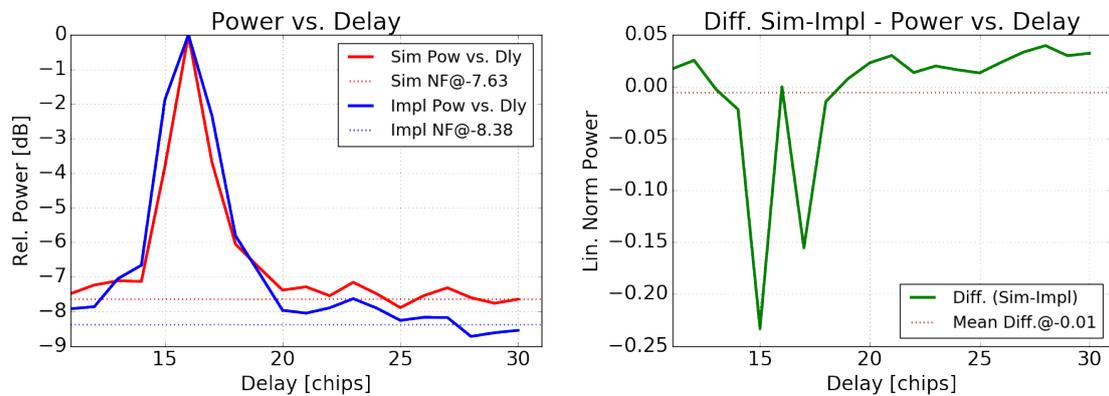


Figure 7.1.3: Power vs. delay of reference and implementation on the left and linear difference between reference and implementation on the right. Both for PRN3 and the first second at 1 Msps.

The last differential image, shown in Figure 7.1.4, represents the linear difference between the two 1 Msps DDMs for the implementation and the reference, respectively. Also this figure shows the difference mainly around the peak area where the implementation shows the wider wave-form for the power vs. delay plot.

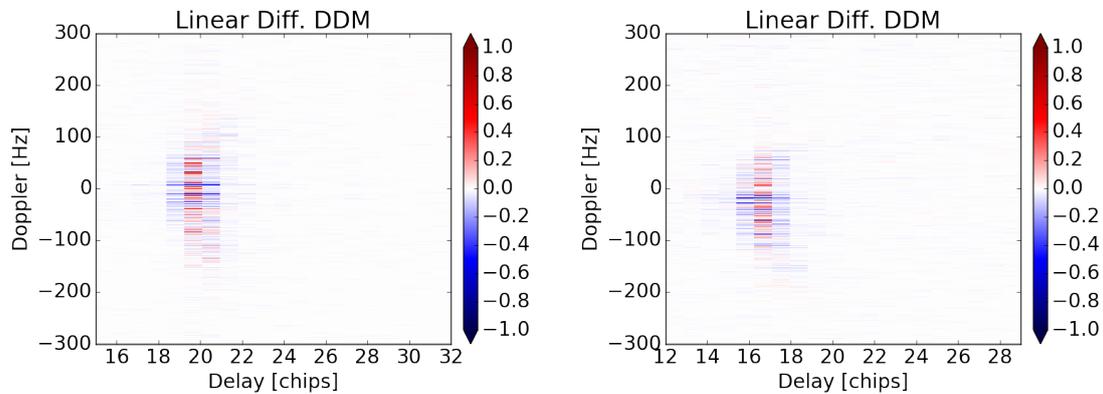


Figure 7.1.4: Linear differential DDM of PRN1 (on the left) and PRN3 (on the right) at 1 Msps.

### 7.1.1 Waveform Differences

As the presented results in Section 7.1 have shown, the waveform for the normalised power vs. delay plot for the implementation does show a wider image around the peak compared to the reference. At the moment of writing this thesis, the reason was not further investigated but two possible reasons should be mentioned.

**PRN-Sequence Generation:** Since the wave-forms for both investigated satellites (PRN1 and PRN3) show a similar behaviour, one could think that the PRN generator might be faulty.

**Meta-Data Generation:** Another reason could be the meta-data processing. The offline processor only considers those slices which have a Doppler-lock whereas the implementation assumes that the meta-data have to be right. For processing one second blocks, this reduces the number of usable slices for one DDM from 1000 to 800, in case of the offline processor and a sampling rate of 1 Msps.

## 7.1.2 Algorithm Complexity

The chosen algorithm has some benefits for the implementation but a lack in Doppler range. Also the computational time between the preprocessor thread and the DDM processor thread is unbalanced. Both issues could be corrected at a slightly computational costs as shown in Section 2.4. The cost at which this change would come is not clear but the implementation might require more memory for matrix operations.

## 7.1.3 Hardware Platform

As already pointed out in Chapter 5, there are some shortcomings with the performance of the selected embedded platform based on the Beaglebone-Black development board. Additionally, some improvements could be applied to the LimeSDR board and the FPGA firmware as well as to the LMS7002M front-end.

### Host System

For the embedded host system a more capable platform might be selected. A such one could be the upcoming Neutis N5<sup>1</sup>. A quad-core 64 bit ARM cortex-A53 based system coming along with 8 GB eMMC storage and 512 MB RAM. The additional development kit also offers 3 x USB 2.0 and on Ethernet interface.

### LimeSDR

On the LimeSDR side some minor changes could be applied to the antenna matching network as well as to the configuration of the FPGA to reduce bottlenecks and improve the overall performance of the system. Following suggestions are made for future projects:

**LMS7002M Antenna Matching:** The antenna matching network of one port of the LMS7002M receivers could be matched for the required GPS L1 frequency, at 1.575420 GHz, to get the maximum out of it.

---

<sup>1</sup><https://neutis.io>, last visited 2018-05-01

**Hardware Accelerator:** Using the the spare resources of the FPGA would allow one to off load computational tasks to LimeSDR instead of executing them on the host system. This would require a careful planing of the firmware as well as the data required to do such computational tasks.

**USB Datarate:** In case the USB bandwidth to the host system is not big enough, the 4 Bytes per sample could be reduced to 3 Bytes instead. This is possible since the ADC has 12 Bit resolution. This configuration change on the LimeSDR comes at the price of increased computational time on the host site.

## 7.2 Outlook

Remote sensing around the field of environmental science becomes more and more important due to the increase of environmental changes and severe weather phenomena in the past decade. To study and better understand the world we are living in, permanent and high density surveillance is required more than ever. Beside high-end solutions offering great resolution and refresh rate at a high cost, there is also a need for low-end implementations with basic functionality at low cost.

This master thesis focused on a low-end implementation with the aim of creating a software-based DDM-Receiver for COTS-components, guaranteeing a low price tag so that future nano-satellite projects could be equipped with such an inexpensive alternative to high-end satellite DDM-Receivers.

During this thesis the following contributions were made to achieve this goal.

- A DDM offline processor based on a high-level language was implemented to test the proposed algorithm and to evaluate the quality of the output. Furthermore, a high resolution 80 Msps reference as well as a 1 Msps reference DDMs were generated for later tests. From the algorithm used, a DDM resolving Doppler and delay range resulted.
- An initial implementation of a DDM-receiver was realised and verified against the created reference. The tests showed mixed results, some of them could be improved others show the limit of a low-end solution. With the used sampling rate of 1 Msps

and the resulting delay resolution of  $1 \mu\text{s}$ , parameters such as the SWH are hard to measure with high accuracy.

- The system was running on a generic embedded system based on an ARM processor and compared against a potentially stronger desktop computer. This showed that a single-core embedded CPU might not be sufficient to handle the workload. Nevertheless, the implementation is capable of running on a low cost embedded system with the potential of increase the efficiency.
- The resulting DDMs showed the spread over the Doppler and delay range for both, the offline processed reference and the results from the implementation. Furthermore, also geometrical differences in the transmitter and receiver alignment were visible.

While this implementation shows the capabilities of a software-based DDM-receiver, it also exposes the weakness of such an approach. The supported Doppler bandwidth of  $\pm 500 \text{ Hz}$  might be enough for low altitude flights but need to be increased for nano-satellite missions [5]. This could be overcome with a shorter correlation time than the PRN length of 1 ms.

For realtime testing, the implementation has to be married with a reference receiver supplying the required meta-data (mean Doppler, delay and navBit). Also further investigation requires the search of a suitable multi-core embedded platform to be capable of creating DDMs without the loss of data and therefore retaining the resolution.

Succeeding with those modifications this low cost software receiver could have the potential of being part of a sensor of an early warning system on-board of nano-satellite which itself is one of many covering the Earth.

All related datasets, Matlab/Octava code, source code of the implementation as well as scripts for plotting the graphs are available by request. The data can be requested from [r\\_gaehwiler@bluewin.ch](mailto:r_gaehwiler@bluewin.ch) or [yannick.lemoullec@ttu.ee](mailto:yannick.lemoullec@ttu.ee).

# References

- [1] V. U. Zavorotny, N. Rodriguez-alvarez, D. M. Akos, J. A. Smith, A. Camps and C. W. Fairall, 'Airborne GNSS-R wind retrievals using delay- Doppler maps Airborne GNSS-R Wind Retrievals Using Delay – Doppler Maps', vol. 51, no. January 2016, pp. 626–641, 2013. DOI: 10.1109/TGRS.2012.2196437.
- [2] J. NASA, *NASA - Jet Propulsion Laboratory - Scatterometry*. [Online]. Available: <https://winds.jpl.nasa.gov/aboutscatterometry/history/> (visited on 29/04/2018).
- [3] J. K. Harmon, R. E. Arvidson, E. A. Guinness, B. A. Campbell and M. A. Slade, 'Mars mapping with delay-Doppler radar', *Journal of Geophysical Research-Planets*, vol. 104, no. E6, pp. 14 065–14 089, 1999, ISSN: 0148-0227. DOI: Doi10.1029/1998je900042.
- [4] M. Martin-Neira, 'A Passive Reflectometry and Interferometry System(PARIS)- Application to ocean altimetry', *ESA journal*, vol. 17, no. 4, pp. 331–355, 1993, ISSN: 03792285.
- [5] V. U. Zavorotny and A. G. Voronovich, 'Scattering of GPS signals from the ocean with wind remote sensing application', *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 2 II, pp. 951–964, 2000, ISSN: 01962892. DOI: 10.1109/36.841977.
- [6] J. Wickert, E. Cardellach, M. Martin-Neira, J. Bandejas, L. Bertino, O. B. Andersen, A. Camps, N. Catarino, B. Chapron, F. Fabra, N. Floury, G. Foti, C. Gommenginger, J. Hatton, P. Hoeg, A. Jaggi, M. Kern, T. Lee, Z. Li, H. Park, N. Pierdicca, G. Ressler, A. Rius, J. Rosello, J. Saynisch, F. Soulat, C. K. Shum, M. Semmling, A. Sousa, J. Xie and C. Zuffada, 'GEROS-ISS: GNSS Reflectometry, Radio Occultation, and Scatterometry Onboard the International Space Station', *IEEE Journal of*

- Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 10, pp. 4552–4581, 2016, ISSN: 21511535. DOI: 10.1109/JSTARS.2016.2614428.
- [7] T. Hobiger, R. Haas and J. S. Löfgren, ‘GLONASS-R: GNSS reflectometry with an FDMA based satellite navigation system’, *Radio Science*, n/a–n/a, 2014, ISSN: 00486604. DOI: 10.1002/2013RS005359. [Online]. Available: <http://doi.wiley.com/10.1002/2013RS005359>.
- [8] C. Ruf, M. Clarizia, S. Gleason, Z. Jelenak, J. Murray, M. Morris, S. Musko, D. Posselt, D. Provost, D. Starkenburg and V. Zavorotny, *CYGNSS Handbook - Cyclone Global Navigation Satellite System*. 2016, ISBN: 978-1-60785-380-0.
- [9] CISC, *Gold RTR Mining*. [Online]. Available: [https://www.ice.csic.es/research/gold\\_rtr\\_mining/index.php](https://www.ice.csic.es/research/gold_rtr_mining/index.php) (visited on 28/04/2018).
- [10] Myriadrf, *LimeSDR Wiki*. [Online]. Available: <https://wiki.myriadrf.org/LimeSDR> (visited on 11/02/2018).
- [11] Lime Microsystems, ‘LMS7002M - FPRF MIMO Transceiver IC With Integrated Microcontroller’, 2015, [Online]. Available: <https://github.com/myriadrf/LMS7002M-docs>.
- [12] A. Komjathy, M. Armatys, D. Masters, P. Axelrad, V. Zavorotny and S. Katzberg, ‘Retrieval of Ocean Surface Wind Speed and Wind Direction Using Reflected GPS Signals’, *Journal of Atmospheric and Oceanic Technology*, vol. 21, pp. 515–526, 2004, ISSN: 0739-0572. DOI: 10.1175/1520-0426(2004)021<0515:R00SWS>2.0.CO;2.